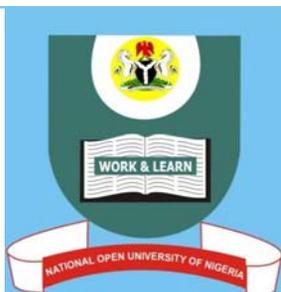


CIT 712

INFORMATION SYSTEM DESIGN AND PROGRAMMING



NATIONAL OPEN UNIVERSITY OF NIGERIA

COURSE GUIDE

**CIT 721
INFORMATION SYSTEMS DESIGN AND PROGRAMMING**

Course Developer	Professor Abdalla Uba Adamu Bayero University Kano.
Course Writer	Professor Abdalla Uba Adamu Bayero University Kano.
Course Editors	Mr. Olusanya Olufunso. O. Director, Gateway ICT Institute, Ogun State. & Mr. A. M. Balogun National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island
Programme Leader	Dr. Sunday Reju National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island
Course Co-ordinators	Mr A. M Balogun National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island & Mr Bankole Abiola National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island



NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Annex
245 Samuel Adesujo Ademulegun Street
Central Business District
Opposite Arewa Suites
Abuja

e-mail: centralinfo@nou.edu.ng

URL: www.nou.edu.ng

National Open University of Nigeria 2006

First Printed 2006

ISBN: 978-058-327-0

All Rights Reserved

Printed by

For

National Open University of Nigeria

TABLE OF CONTENTS	PAGE
Course Aim.....	1 - 2
Course Objectives.....	2 - 3
Course Materials.....	3

Course Aims

This course is designed as an introductory course to systems analysis and design. The course presents an overview of information systems and the systems development life cycle for the analyst. Course emphasis will then be on tools and techniques that the programmer and analyst can use to develop information systems. Classical and structured tools for describing data flow, data structures, process flow, file design, input and output designs and program specification will be applied to constructing and documenting systems. The course will survey other important skills for the analyst, such as interviewing, data communications, project management and cost-benefit analysis.

This course reflects the information explosion of recent years, the new technological advances in information systems, and the exponential growth in electronic business processes. The course provides extensive and comprehensive coverage of systems development techniques for the redesign of core information systems.

Thus the major goal of this course is to prepare future analysts, programmers, managers and consultants for the delivery and evaluation of IT-enabled process redesign projects. Additional goals include: An understanding of the systems development life cycle and alternative development methodologies, mastery of communication, project management, feasibility and cost analysis tools, mastery of analysis and design modeling techniques.

- An understanding of systems development tools available to analysts.
- The development of structured and object-oriented modeling skills.
- An understanding of system design issues.
- An understanding of application architectures and networks.
- An understanding of the application development process.
- A thorough understanding of system implementation issues.
- An understanding of how IT supports business requirements in changing environments.

- An understanding of IT development and trends.
- An understanding of legal and ethical considerations.
- Promotion of critical thinking.

Course Objectives

Students will be able to:

- Identify the organizational roles involved in systems development.
- Identify, compare and contrast several approaches to systems development.
- Identify, discuss and apply the SDLC phases.
- Identify the skills necessary to become a systems analyst.
- Discuss the process of project management.
- Describe CASE and its role within the SDLC.
- Discuss corporate strategic planning and information systems planning.
- Prepare, discuss and present a Statement of Work, Baseline Project Plan, and other SDLC phase deliverables.
- Identify various methods for assessing project feasibility.
- Select appropriate tools for determining systems requirements.
- Prepare selected process models, logic models, data models, and object models.
- Describe alternative design strategies.
- Discuss object oriented analysis and design.
- Discuss Rapid Application Development.
- Design inputs and outputs for a system.

- Discuss several approaches to implementation.
- Discuss the importance of systems maintenance.

Course Materials

Major components of the course are:

1. Course Guide
2. Study Units
3. Assignment File
6. Lack of flexibility in the MIS or DSS.
7. False assumptions made by data processing professionals in developing the DSS or MIS.

MAIN COURSE

Course Code	CIT 721
Course Title	Information Systems Design and Programming
Course Developer	Professor Abdalla Uba Adamu Bayero University Kano.
Course Writer	Professor Abdalla Uba Adamu Bayero University Kano.
Course Editors	Mr. Olusanya Olufunso. O. Director, Gateway ICT Institute, Ogun State. & Mr. A. M. Balogun National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island
Programme Leader	Dr. Sunday Reju National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island
Course Co-ordinator	Mr A. M Balogun National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island & Mr Bankole Abiola National Open University of Nigeria 14/16 Ahmadu Bello Way, Victoria Island



NATIONAL OPEN UNIVERSITY OF NIGERIA

National Open University of Nigeria
Headquarters
14/16 Ahmadu Bello Way
Victoria Island
Lagos

Abuja Annex
245 Samuel Adesujo Ademulegun Street
Central Business District
Opposite Arewa Suites
Abuja

e-mail: centralinfo@nou.edu.ng

URL: www.nou.edu.ng

National Open University of Nigeria 2006

First Printed 2006

ISBN: 978-058-327-0

All Rights Reserved

Printed by

For

National Open University of Nigeria

TABLE OF CONTENTS		PAGE
Module 1	System Concepts.....	1
Unit 1	Introduction to Systems.....	1 - 8
Unit 2	Information Concepts.....	9 -14
Unit 3	Information Systems.....	15-20
Unit 4	System Analysts and Designers.....	21-28
Unit 5	Systems Thinking.....	29-34
Module 2	Systems Approaches	35
Unit 1	Structured Information Systems Development....	35-39
Unit 2	System Analysis and Design in End User Developed Applications.....	40-44
Unit 3	System Development Life Cycle (SDLC).....	45-53
Unit 4	The System Development Matrix.....	54-57
Unit 5	Requirements Management.....	58-63
Module 3:	System Methodologies.....	64
Unit 1	Project Selection.....	64-68
Unit 2	System Investigation.....	69-76
Unit 3	System Design.....	77-85
Unit 4	Cost-Benefit Analysis (CBA) in System Designs.....	86-91
Unit 5	Tasks in System Design Phase.....	92-98
Module 4	System Design and Implementation.....	99
Unit 1	System Design Tools.....	99 -109
Unit 2	Development Phase.....	110-116
Unit 3	Implementation Phase.....	117-119
Unit 4	Operations and Maintenance Phase.....	120-127
Unit 5	The design process.....	128-132
Module 5	Information Systems Development toolkit.....	133

Unit 1	Key System Development Life Cycle	
	Supporting Processes.....	133-138
Unit 2	The process model.....	139-142
Unit 3	Examples of programming language.....	143-151
Unit 4	Risk Management.....	152-156
Unit 5	IT developments Case Study.....	157-169
Module 6	Management Information System.....	170
Unit 1	Theory of Management.....	170-176
Unit 2	Organizational Concepts.....	177-185
Unit 3	Management Information Systems (MIS).....	186-190
Unit 4	Risks Associated With MIS.....	191-195
Unit 5	Information Systems and Databases.....	196-202
References/Further Reading		203-205

MODULE 1 SYSTEM CONCEPTS

Unit 1	Introduction to Systems
Unit 2	Information Concepts
Unit 3	Information Systems
Unit 4	Systems Analyst and Designers
Unit 5	Systems Thinking

UNIT 1 INTRODUCTION TO SYSTEMS

CONTENTS

1.0	Introduction
2.0	Objectives
3.0	Main Content
3.1	Overview
3.2	Key Elements of a System
3.2.1	The Transformation Process
3.2.2	Outside and Inside the System
3.3	System Performance
3.4	Classification of Systems
3.4.1	Open vs. Closed Systems
3.4.2	Adaptive Systems
3.4.3	Deterministic vs. Probabilistic Systems
3.4.5	Shared and Overlapping Sub-Systems
3.5	The Systems Approach (or Systems Theory)
3.6	Analyzing a System
4.0	Conclusion
5.0	Summary
6.0	Tutor Marked Assignment

1.0 INTRODUCTION

This unit introduces you to the basic concepts of systems and theories of information management. This is meant as an overview to understanding the background to information system analysis and design.

2.0 OBJECTIVES

By the end of this unit, you should be able to

- Describe the key elements of a system
- Explain the measures of system performance
- Explain the structural patterns of analyzing systems
- Explain the difference between open and closed systems

3.0 MAIN CONTENT

3.1 Overview

It is informative to find out exactly what is meant by a system. A system can be defined as: an inter-related set of components that work together within an identifiable boundary to achieve some overall goals.

Put another way, a system is a collection of parts which are related to each other and which may depend on each other and which work together as a coherent whole. A system has input, process (or transform), output, feedback and control mechanisms. We are primarily concerned with organizational and information systems.

However, it is important to note that the term 'system' does not solely apply to computer systems. It can apply to things as diverse as the legal system, the transport system, the social services system, the solar system and even the digestive system! Any phenomenon which exhibits a relationship among interrelated components can be termed a system.

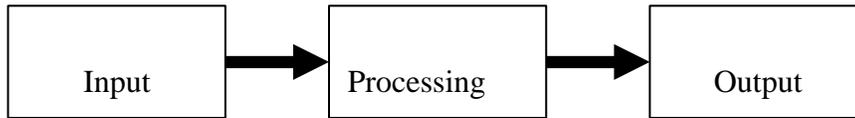
3.2 Key Elements of a System

3.2.1 The Transformation Process

All systems consist of input, process and output:

- **Input:** Capturing and assembling elements that enter the system to be processed e.g. raw materials, people, skills, money.
- **Processing:** The transformation processes which convert input into output, e.g. a production process, a mathematical calculation, a meeting of people.

- **Output:** Transferring elements that have been produced by a transformation process to their ultimate destination in a form which will be meaningful to its recipient, e.g. a product, a piece of information. In the case of information, output in the form of a report, a telephone call etc.



It is not necessary to consider all the inputs and outputs of a system, but consider only those inputs which are considered to affect the outputs and only those outputs which are relevant to the systems objectives.

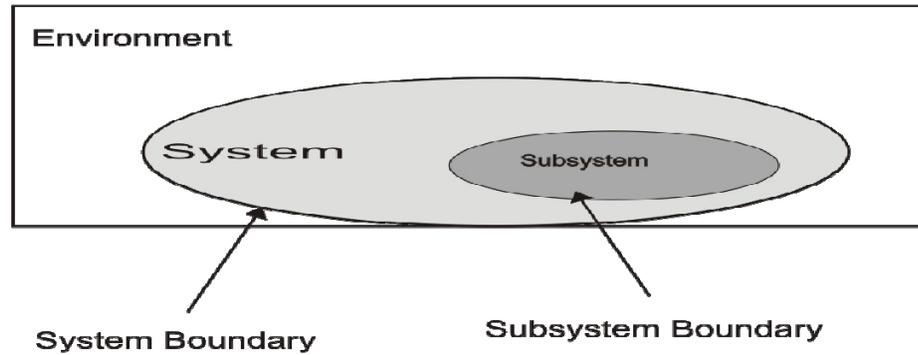
3.2.2 Outside and Inside the System

Environment: All those external elements, whose changes in behavior, attitude etc., effect the working of the system and all those external elements, which are effected by the system's environment.

Boundaries: The separation of a system from its environment i.e. the features or constraints which depict the scope of activities for a system and delineate areas of responsibility. The system is inside the boundary, whereas the environment lies outside. Boundaries can be imposed by the nature of the system itself or can be decided upon by management. They may be flexible, changing over time or as the wider organization changes.

Interfaces: It is the area of contact between one system boundary and another. Several systems may share the same environment and may be connected to one another by means of a shared boundary or interface.

Sub-system: A system representing a component of a larger system. Systems often consist of numerous subsystems. Each subsystem has elements, interactions with other subsystems, and objectives. Subsystems perform specialized task for the overall system. In business, functions such as marketing, finance, and manufacturing are subsystems.



3.3 System Performance

Two measures of system performance are:

Effectiveness: The degree to which set goals are achieved. It is therefore concerned with the results or the outputs of a system.

Efficiency: A measure of the use of inputs (or resources) to achieve results. How much money is used to generate a certain level of sales.

The efficiency and effectiveness of a system can be measured by having the following two elements built into the system:

Feedback: a flow of information from the output component to the decision-maker concerning the system's output or performance. Indicates if the system performance is meeting standards. Based on this, the decision-maker, who acts as a control, may decide to modify the inputs or the processes or both.

Control: A major system function which monitors and evaluates system feedback, to establish to what degree system goals are being achieved

A key element in the success of a system is **its ready adaptation to changes in its environment**. This again emphasizes the need for monitoring and control of systems based on relevant, timely feedback.

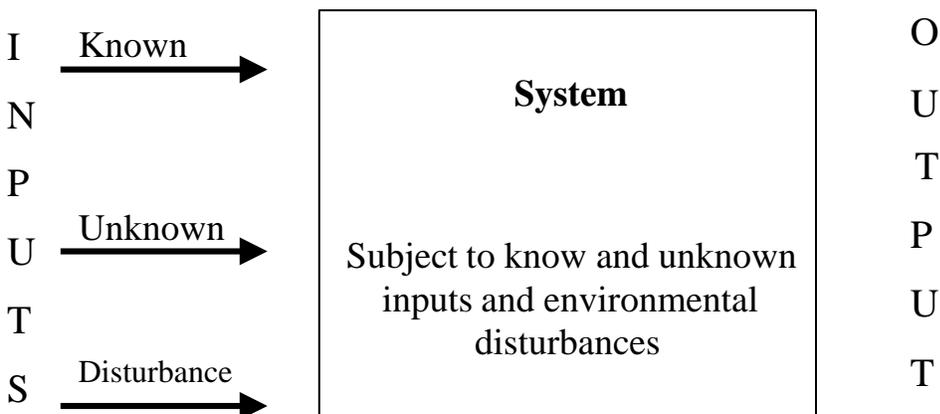
Thus a system is composed of subsystems or components that are inter-related, has boundaries, exists within an environment, has interfaces between subsystems and with the environment, faces constraints (limitations) and receives inputs from and delivers outputs to its environment in order to fulfill its purpose (goal oriented).

3.4 Classification of Systems

3.4.1 Open vs. Closed Systems

Closed Systems: A system that is cut off from its environment and does not interact with it. These systems have no exchange with the environment, i.e. all interaction goes on within the system's own boundaries. This term normally applies to machinery where, if inputs are known, then outputs can be accurately predicted. Systems within organisations cannot be described in this way - interaction with other systems is an inherent characteristic of organisational systems. A closed community would be an example of a (social) closed system.

Open Systems: A system that interacts freely with its environment. It receives inputs from its environment, processes or transforms these inputs and passes outputs of various types back into its environment. In this type of system, only some of the relevant inputs can be identified, others may occur unexpectedly, for example, a company's competitor may unexpectedly lower prices etc.



3.4.2 Adaptive Systems

A system that has the ability to change itself or its environment in order to survive

3.4.3 Deterministic vs. Probabilistic Systems

Deterministic/Mechanistic Systems: These are predictable systems where, if the inputs are known, as well as the present state of the system, then the

system's outputs can be accurately forecast. Machines and computer programs are examples.

Probabilistic/Stochastic Systems: The output of these systems can only be predicted as a probability rather than a certainty. This is true of almost all social or organizational systems as it is always impossible to account for all inputs.

3.4.4 Shared and Overlapping Sub-Systems

Systems consist of **sub-systems**, or in commercial terms, organizations are composed of departments and sections and these sections **interact** and are therefore **inter-dependent**. One sub-system can belong to one or more systems, therefore it can be inferred that a change to one sub-system may affect more than one system. The use of overlapping systems is often a sound economical arrangement, for example, a centralized computer facility can be used by a number of departments in an organization to reduce overall costs. Shared systems do however cause some problems. There is a need for a high level of co-ordination and in the event of change, approval will need to be sought from a number of sources, making shared systems less flexible where rapid change is needed.

3.5 The Systems Approach (or Systems Theory)

The Systems Approach or Systems Theory has a set of ideas with which we can view systems. This set of ideas can be summarized as follows:

All systems are composed of **inter-related** parts or **sub-systems** and the system itself can only be fully explained and understood when viewed as a whole. This is known as **holism** or **synergy**. The systems approach takes the view that the whole is greater than the sum of the parts and that by looking at separate parts of a system in isolation, **vital inter-relationships** will be ignored or misunderstood.

Systems are **hierarchical** in structure, i.e. a system is made up of sub-systems and each subsystem is made up of further sub-systems. For example, your course is a sub-system of the degree courses offered by the school, which is a subset of the Faculty, subset of the National Open University. It is important to decide where to impose a boundary when attempting to analyze a system, so that, only those sub-systems which relate to your area of interest are considered.

The components of a system form an indissoluble whole so that when one part is changed, this change will effect other parts of the overall system.

When change is effected, it is important to realise what systems it does effect and what needs to be done to ensure that the changes are properly implemented.

Sub-systems need to work towards the goals of the system to which they belong and not pursue their own goals independently. Where this latter situation does occur, a condition of sub-optimization occurs. This is to be avoided if possible, but it must be recognized that in most organizations, conflicting objectives across departments is inevitable. In these cases, some form of compromise need to be reached. For example, one department's goal may be to clear up a backlog of work. To do this they decide to work overtime. This in turn affects the Computer Services department who are quite content with leaving work at 5.00pm. How is this resolved?

3.6 Analyzing a System

Important questions to be answered when analyzing a system are:

- What are its boundaries?
- What are its inputs, processes and outputs?
- What feedback and control mechanisms are in place?
- How can its efficiency and effectiveness be measured?

The systems approach recommends the following steps:

- Decompose a system into smaller, more manageable and understandable subsystems, preferably of uniform size. This facilitates the focusing of attention on one area (subsystem) at a time without interference from other parts.
- Analyze each subsystem separately. This allows attention to concentrate on the part of the system pertinent to a particular audience, without confusing people with details irrelevant to their interest.
- Describe the subsystems/components and their relationships with each other and the external environment.

4.0 CONCLUSION

Thus, a system is the smallest conceptual unit of a holistic structure. It is made up of many components, and each component operates on a specific set of rules and regulations. System approach is always tackled from the

point of view of analyzing systems on a component basis, because understanding the components may lead to understanding the entire structure, and thus predicting and controlling it.

5.0 SUMMARY

In this unit you have learnt about the key elements of a system, and the factors that affect system performance. You have also learnt about the boundaries of systems, and the relationship between these boundaries and system performance. Finally, you have been able to classify systems into categories of control.

6.0 TUTOR MARKED ASSIGNMENT

- What are the key elements of a system?
- What are the measures of system performance?
- Describe the structural patterns of analyzing systems
- What are the differences between open and closed systems?

UNIT 2 INFORMATION CONCEPTS

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Information
 - 3.3 Data Vs Information
 - 3.4 Quality of Information
 - 3.5 Information Levels
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor marked Assignment

1.0 INTRODUCTION

The terms 'data' and 'information' are used interchangeably in everyday speech as meaning the same thing. However, the terms have distinct meanings. System analysis and design are about data and information management, and this unit sets out to introduce you to the basic concepts of information.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Define the concept of information
- Distinguish between **data** and **information**
- Explain the **qualities** of good information
- Explain the **levels** of information

3.0 MAIN CONTENT

3.1 Overview

Raw facts, figures, images or sounds collected from observations or recordings about events, objects or people, which can be stored on a manual or computer-based medium e.g. employee's name and number, number of

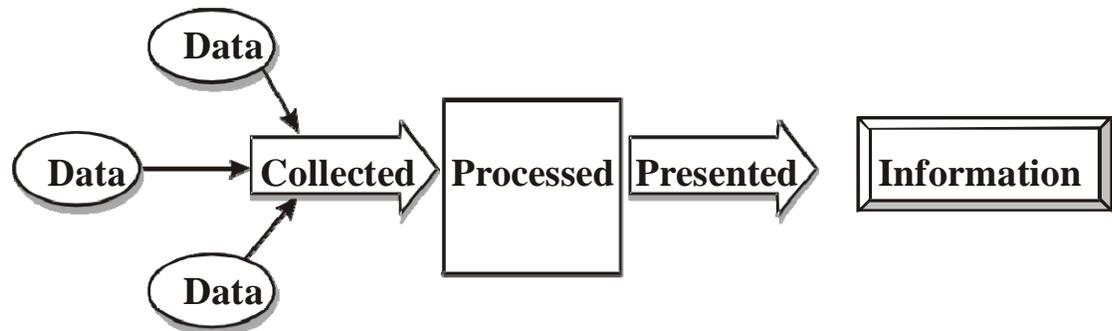
hours worked in a week, inventory part numbers, or sales orders. Data has little meaning/value in its own right, it only has meaning when it is processed and put into context as information. For example, if we are told that John scored 85 in a test, it does not tell us a lot. Is it 85 marks? If it is 85% it may appear a good result, but how did the other people taking the test perform? What is the pass mark in the test? If it is 85 marks, what was the number of marks available? To be of any value, it needs to be processed in some ways to produce information e.g. collected, sorted, organized, presented in acceptable formats. Data is the input raw materials from which information is produced.

3.2 Information

Information is a data that has been processed and presented in a useful format that will enable an individual to gain knowledge in order to be able to make a decision. The act of producing data does not itself produce information. Information is data that have been interpreted and understood by the recipient of the message.

Information has meaning within a specific context and is useful for making a decision, solving a problem, performing a task etc. It is important to note that different individuals may need the same data arranged in quite different ways to give them information. The information required varies with the type of decision, problem, or task, the type of person and his/her needs.

3.3 Data vs Information



Examples of Data and Information

	Payroll System	Sales Ledger System
Raw data	Rate of pay, hours worked, deduction etc.	Customer sales, customer remittances etc.
Information	Pay slip, report for management etc.	Balance owing statements, debtors balances etc.

3.4 Quality of Information

Good information is that which is used and which creates value. Research shows that good information has numerous qualities as follows:

- **Relevance/Appropriateness:** Information must be relevant to the problem being considered. Too often reports etc, contain irrelevant parts which make understanding more difficult and cause frustration to the user and lead to information overload.
- **Accuracy:** Information should be sufficiently accurate for the purpose for which it is intended, Need for accuracy varies according to information usage e.g. marketing director only interested in sales figures to +/- N15,000?
- **Completeness:** The information should be complete in respect of the key elements of the problem.
- **Timeliness:** Good information is that which is communicated in time to be used.
- **Reliable:** Users must have confidence in the source of the information for it to be used.
- **Communicated to the right person:** Information suppliers need to analyze the key decision points in an organization in order to direct information exactly where it is required.
- **That which contains the right level of detail:** Information should contain the least amount of detail consistent with effective decision making. The level of detail should vary with the level in

the organisation; the higher the level the greater the degree of compression and summarisation.

- **Communicated by an appropriate channel of communication.** To be **useable** by the manager, information must be transmitted by means of a communication process.
- Communication involves the interchange of facts, thoughts, value judgements and opinions and the communication process may take many forms: face-to-face conversations, telephone calls, informal and formal meetings, reports, tabulations, VDU transmissions etc. Whatever the process, good communication results where the sender and receiver are in accord over the meaning of a particular message.
- **That which is understandable by the user.** **Understandability** is what transforms data into information. If the information is not understood it cannot be used and thus cannot add value.

From above, we can see that many, many things need to be right before information can be considered as good. Note particularly, how many of the factors relate to social and behavioral characteristics. It is not sufficient merely to consider the technical aspects of data capture and processing; these are only one aspect of information system.

Information is crucial to business operation and survival interacting with environment -customers, suppliers etc. and for interaction/communication within the business. It is crucial that the business has information systems to manage this information.

3.5 Information Levels

Information within an organization can be analyzed into three levels. These levels of information are referred to as:

Strategic Information: used by senior managers to plan the objectives of their organisation and to assess whether the objectives are being met in practice e.g. population statistics, investment statistics, national resource availabilities. Much of this information must come from environmental sources, although internally generated information will also be used. It relates to long-term planning in an organisation, typically 3-5 years.

Tactical Information: used by middle management to ensure that the resources of the business are employed efficiently and effectively to achieve the strategic objectives of the organisation e.g. sales analyses and forecasts, production requirements, annual financial statements, predicted course/student numbers. A large proportion of this information will be generated from within the organisation. Tactical information is usually prepared regularly - perhaps weekly or monthly and it is for medium-term planning, typically 6-24 months.

Operational Information: used by frontline managers such as foremen or head clerks to ensure that specific tasks are planned and carried out properly within a factory or office. Operational information is prepared regularly - perhaps weekly or daily and it is used for short-term planning, days, weeks, possibly hourly e.g. stock levels, overdue purchase orders, production control, module enrolments.

The destination level of information is very important as the information must be material to the user, but without going into unnecessary and time consuming detail in order to achieve pointless accuracy. For example: Operational control may need information accurate to the nearest penny. Management control may be satisfied with costs to the nearest hundred or thousand pounds. Greater detail would serve no purpose. Strategic planning may be satisfied with figures to the nearest ten thousand, hundred thousand or even million pounds. In all cases, the information and reports must be relevant to the particular user.

4.0 CONCLUSION

Information is crucial to business operation and survival interacting with environment -customers, suppliers etc. and for interaction/communication within the business. It is crucial that the business has information systems to manage this information.

5.0 SUMMARY

In this unit, you have learnt the concept of information, and the differences between data and information. The unit has also explained the qualities of good information, as well as the various levels of information. All these are sub-sets of information needed for systems analysts before they begin the analysis and design of systems.

6.0 TUTOR MARKED ASSIGNMENT

- Explain the usefulness of information
- What is the difference between **data** and **information**”?
- What are the **qualities** of good information?
- Explain the differences between Strategic and Tactical **levels** of information.

UNIT 3 INFORMATION SYSTEMS

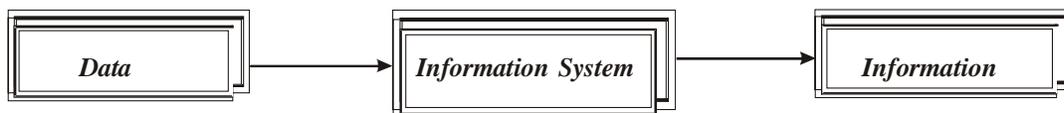
CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Contents
 - 3.1 Overview
 - 3.2 Components of CIS
 - 3.3 Information System (IS) vs. Information Technology (IT)
 - 3.4 Objectives of Information Systems in an Organization
 - 3.5 Major Categories of Information Systems
 - 3.5.1 Classification by functional areas
 - 3.5.2 Classification by Support provided
 - 3.5.3 Relationships among TPS, MIS, DSS, EIS and ES
 - 3.6 The Complexity of Information Systems
 - 3.7 The Benefits of Information Systems
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

Information systems are generally used in organizations to help provide for the information needs of the organization. An information system can be defined as: a set of interrelated components that collects, stores, and processes data from various sources to provide information necessary to support and improve the day-to-day operations in a business.

Thus graphically, the expression would look like this:



2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Describe the components of a computerized information system.

- Distinguish between information system and information technology.
- Explain the objectives of an information system in an organization.
- Classify major categories of information systems.

3.0 MAIN BODY

3.1 Overview

Information systems transform data into useful information and include both technology and people. An information system generally incorporates one or more computers and the software to control them. In this case, the information system is said to be a computerized information system.

3.2 Components of CIS

A computerized information system (CIS) can be understood as having the following parts:

- **Users** - the people who add information to the system and request information from the system
- **Hardware** - not only the computers in the system but also any networks linking the computers, input devices like keyboards and mice, output devices like monitor and printers.
- **Data** - what the system stores and processes.
- **Software** - the electronic coding which controls all aspects of the hardware and the data.
- **Documents** - manuals on how to use the system, sometimes even files of data which should not or could not be stored electronically.

3.3 Information System (IS) vs. Information Technology (IT)

Information System (IS): The flow of information in an organization and between organizations encompassing the information the business creates, uses and stores. Concerns with efficient use of resources for providing the

required level of information support for the management of business operations. Represents the applications perspective.

Information Technology (IT): The enabling mechanism which facilitates the processing and flow of this information, as well as the technologies used in the physical processing to produce a product or provide a service. Includes telecommunications, computers and automation technologies, represents the technical perspective. Information technology has significantly expanded the power and potential of most information systems. Information technology is a contemporary term that describes the combination of computer technology (hardware and software) with telecommunications technology (data, image, and voice networks). Technology has created a data and information explosion in virtually all businesses. The ability of businesses to harness and manage this data and information has become a critical success factor in most businesses.

3.4 Objectives of Information Systems in an Organization

Information systems in an organization are dedicated to improving the performance of knowledge workers in organizations through the application of information technology.

- Improving the performance is the ultimate objective of information systems - not the storage of data, production of reports, or even 'getting the right information to the right person at the right time.
- Knowledge workers (managers, professionals, staff analysts and clerical workers) are the clientele.
- Organizations are the context. Focus is on information handling in goal seeking organizations.
- The application of information technology is the challenge and opportunity facing the IS professional.

3.5 Major Categories of Information Systems

It is automated information systems that we are going to study. However, these still cover a huge variety of systems. It is useful to break them down into the following categories:

3.5.1 Classification by functional areas:

- Marketing Information System
- Manufacturing Information System
- Distribution
- Accounting and Finance
- Personnel
- Administration

Early stages saw ad hoc development with private ownership of data and programs. Current wisdom is to build “integrated systems” with common databases.

3.5.2 Classification by support provided:

- **Transaction Processing System (TPS):** A system that processes data resulting from business transactions, updates operational databases, and produces business documents.
- **Management Information System (MIS):** A system or group of systems which collects and presents management information relating to a business in order to facilitate its control. Management Information Systems are those that support all management levels in the conduct of their functions, such as operations, administration, or planning and programming.
- **Office Automation System (OAS):** A system that automates office procedures and enhances office communications and productivity.
- **Decision Support Systems (DSS):** A system that provides interactive ad-hoc support for the decision making process of managers.
- **Expert System (ES):** A knowledge-based system that provides expert advice and acts as expert consultants to users.
- **Executive Information Systems (EIS):** A system that provides critical information tailored to the information needs of top management.

3.5.3 Relationship among TPS, MIS, DSS, EIS and ES:

- The technologies can be viewed as being unique classes of information technology.
- They are interrelated.
- They each support some aspects of managerial decision making.
- The evolution and creation of the newer tools help expand the role of IT for the betterment of management in organizations.
- The interrelationship and co-ordination between these tools is still evolving.

3.6 The Complexity of Information Systems

Information systems can be very complex even in a small organization:

- Many components (lots of information).
- Much interaction between components.
- Systems within systems.
- The intangibility of information (it is hard to define).
- The subjective nature of information.

3.6 The Benefits of Information Systems

A good information system ensures that:

- the right information is provided.
- to the right degree of accuracy
- when it is required
- in the right format/layout
- to the right people
- in the most efficient way possible.

4.0 CONCLUSION

The major function of information in an organization is to improve organizational efficiency. This can best be done through the effective use of information technology. The variety and categories of classifying information clusters definitely call for a more sustained process of creating information sharing interfaces that enhance organizational efficiency.

5.0 SUMMARY

In this unit, you have learnt about the role of information in organizations and the various components of a computerized information system. You have also learnt about the major categories of information systems, as well as their complexity and benefits.

6.0 TUTOR MARKED ASSIGNMENT

- What are the components of a Computerized Information System (CIS)?
- Explain at least three objectives of an information system in an organization
- What are the functional categories of information systems?
- Explain the complexity of information systems

UNIT 4 SYSTEM ANALYSTS AND DESIGNERS

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Nature of the Work
 - 3.3 System Analyst
 - 3.4 The Changing Role of System Analysts
 - 3.5 Systems Interconnectivity
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

System analysis begins with a systems analyst, a highly trained analytical person with more skills in understanding human organizations and programming techniques. This unit will provide you with “behind-the-scenes” perspectives of system analyst and designers.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the functions of system analyst
- Explain the procedures adopted by system analysts in their jobs
- Explain the functions of programmer-analysts
- List the characteristics of a good systems analyst
- Interpret the changing role of systems analysts

3.0 MAIN CONTENT

3.1 Overview

The design of a new computer-based information system is an exciting and demanding undertaking. First, someone calls attention to a problem with existing information processing procedures. A design team assesses the benefits of using a computer to improve these procedures. Then an abstract model of present processing procedures is developed and designers create a new information processing system. The new procedures are converted into systems specifications and finally into computer programs. During the final stages of development, the system is tested and converted and becomes operational.

3.2 Nature of the Work

The rapid spread of computers and information technology has generated a need for highly trained workers to design and develop new hardware and software systems and to incorporate new technologies. These workers—computer systems analysts, computer scientists, and database administrators—include a wide range of computer specialists. Job tasks and occupational titles used to describe these workers evolve rapidly, reflecting new areas of specialization or changes in technology, as well as the preferences and practices of employers.

A systems analyst needs to know about **computers** and **programming**, but also should know and have a desire to use computers to solve problems. The solution to the “problem” is generally a new information system. Systems analysts solve problems for business organizations, such as these:

- Problems getting orders from customers
- Problems withholding amounts from pay slips for employees
- Problems getting needed information to management
- Problems getting information about orders to transport
- Problems about estimating what to produce in the factory

A systems analyst uses a generic problem solving approach. The analyst uses a series of steps in order to systematically understand and solve the problem. The steps are:

- Research and understand the problem
- Verify that the benefits of solving the problem outweigh the costs

- Develop a set of possible solutions (alternatives)
- Decide which solution is the best and make a recommendation
- Define the details of the chosen solution
- Implement the solution
- Monitor to make sure the desired results were obtained

When a new information system is a “solution” to a problem, it is important to understand the problem the system will solve. This is the essence of systems analysis - understanding and defining what it takes to solve the problem. A business case must be made for solving the problem - if the benefits don't outweigh the costs, then why should the problem be solved at all? There are always many alternative solutions that will solve the problem. They must be identified and evaluated. One of the solutions is chosen based on a variety of factors. The chosen solution is defined in detail and then it is implemented. While the new system is being used, it is important to monitor it to be sure it is doing what is needed to solve the problem. Over time it will need to be supported and perhaps modified.

Thus systems analysts solve computer problems and enable computer technology to meet individual needs of an organization. Some systems analysts also are referred to as **systems developers** or **systems architects**.

Analysts begin an assignment by discussing the systems problem with managers and users to determine its exact nature. They define the goals of the system and divide the solutions into individual steps and separate procedures. Analysts use techniques such as structured analysis, data modeling, information engineering, mathematical model building, sampling, and cost accounting to plan the system. They specify the inputs to be accessed by the system, design the processing steps, and format the output to meet the users' needs. They also may prepare cost-benefit and return-on-investment analyses to help management decide whether implementing the proposed system will be financially feasible.

When a system is accepted, analysts determine what computer hardware and software will be needed to set it up. They coordinate tests and observe initial use of the system to ensure it performs as planned. They prepare specifications, work diagrams, and structure charts for computer programmers to follow and then work with them to “debug,” or eliminate errors from the system. Analysts, who do more in-depth testing of products, may be referred to as software quality assurance analysts. In addition to running tests, these individuals diagnose problems, recommend solutions, and determine if program requirements have been met.

In some organizations, programmer-analysts design and update the software that runs a computer. Because they are responsible for both programming and systems analysis, these workers must be proficient in both areas. As this becomes more commonplace, these analysts increasingly work with object-oriented programming languages, as well as client/server applications development, and multimedia and Internet technology.

Thus the systems analyst, working with other IS personnel, defines the requirements that are used to modify an existing system, or to develop a new system. The systems analyst identifies and evaluates alternative solutions, makes formal presentations, and assists in directing the coding, testing, training, conversion, and maintenance of the proposed system.

3.3 System Analysts

The characteristics of a good systems analyst are:

Working Knowledge of Information Technology: The systems analyst is an agent of change. The systems analyst is responsible for showing end-users and management how new technologies can benefit their business and its operations. The systems analyst must be aware of both existing and emerging information technologies and techniques.

Programming Experience and Expertise: A systems analyst must know how to program, because, they are the principle link between business users and computer programmers. It is wrong to assume that a good programmer will become a good analyst or that a bad programmer could not become a good analyst. Most systems analysts need to be proficient in one or more high-level programming languages.

General Business Knowledge: The systems analysts are expected to immerse themselves in the business and be able to specify and defend technical solutions that address the bottom-line value returned to the business. Systems analysts should be able to communicate with business experts to gain knowledge of problems and needs. It is not uncommon for systems analysts to develop so much expertise, over time, they move out of information systems and into the user community.

Problem-Solving Skills: The systems analyst must have the ability to take a large business problem, break that problem down into its component parts, analyze the various aspects of the problem, and then assemble an improved system to solve the problem. The systems analyst must learn to analyze problems in terms of causes and effects rather than in terms of

simple remedies. The systems analyst must be well organized. System analysts must be able to creatively define alternative solutions to problems and needs.

Interpersonal Communications Skills: The systems analyst must be able to communicate effectively, both orally and in writing. The systems analyst should have a good command of the English language. Almost without exception, communications skills, not technical skills, prove to be the single biggest factor in career success or failure.

Interpersonal Relations Skills: Systems work is people-oriented and systems analysts must be extroverted or people-oriented. Interpersonal skills help systems analysts work effectively with people. Interpersonal skills are also important because of the political nature of the systems analyst's job. The systems analyst's first responsibility is to the business, its management, and its workers. The systems analyst must mediate problems between team problems and achieve benefits for the business as a whole. Systems analysts work in teams composed of IS professionals, end-users, and management. Being able to cooperate, to comprise, and to function as part of a team, is critical for success in most projects. Because development teams include people with dramatically different levels of education and experience, group dynamics is an important skill to develop.

Flexibility and Adaptability: No two systems development projects encountered by a systems analyst are identical. There is no single, magical approach or solution applicable to systems development. Successful systems analysts learn to be flexible and adapt to special challenges or situations presented by specific systems development projects. The systems analyst must be able to recognize when variations upon (or single-instance exceptions to) development standards are necessary and beneficial to a particular project. The systems analyst must be aware of the implications of not following the standards.

Character and Ethics: The nature of the systems analyst's job requires a strong character and sense of ethics. Ethics is a personal character trait in which an individual(s) understands the difference between 'right' and 'wrong' and acts accordingly.

Systems Analysis and Design Skills: All systems analysts need thorough and ongoing training in systems analysis and design. Systems analysis and design skills can be conveniently factored into three subsets: concepts and principles, tools and techniques.

3.4 The Changing Role of Systems Analysts

The role of systems analyst is changing into two distinct positions or roles: business analyst and application analyst.

A business analyst is a systems analyst that specializes in business problem analysis and technology-independent requirements analysis. Typically recruited from the user community, business analysts focus on business and non-technical aspects of systems problem solving. As experts in their business area, they help define system requirements for business problems, and coordinate interactions between business users and technical staff. Business analysts are usually appointed to a specific project, or for a fixed duration. Business analysts will typically be paired with one or more application analysts.

An application analyst is a systems analyst that specializes in application design and technology-dependent aspects of development. Application analysts usually come from computer or information systems backgrounds and education. While they are frequently capable of business problem analysis and requirements analysis, they are experts in the translation of business requirements into technical designs. It should be recognized that both business and application analysts share many skills; particularly, systems analysis skills.

3.5 Systems Interconnectivity

One obstacle associated with expanding computer use is the need for different computer systems to communicate with each other. Because of the importance of maintaining up-to-date information—accounting records, sales figures, or budget projections, for example—systems analysts work on making the computer systems within an organization compatible so that information can be shared. Many systems analysts are involved with “networking,” connecting all the computers internally—in an individual office, department, or establishment—or externally, because many organizations now rely on e-mail or the Internet. A primary goal of networking is to allow users to retrieve data and information from a mainframe computer or a server and use it on their machine. Analysts must design the hardware and software to allow free exchange of data, custom applications, and the computer power to process it all.

Networks come in many variations and network systems and data communications analysts analyze, design, test, and evaluate systems such as local area networks (LAN), wide area networks (WAN), Internet, Intranets, and other data communications systems. These analysts perform

network modeling, analysis and planning. They also may research related products and make necessary hardware and software recommendations. Telecommunications specialists focus on the interaction between computer and communications equipment.

The growth of the Internet and expansion of the World Wide Web, the graphical portion of the Internet, have generated a variety of occupations related to design, development, and maintenance of Web sites and their servers. For example, webmasters are responsible for all technical aspects of a website, including performance issues such as speed of access, and for approving site content. Internet developers or web developers, also called web designers, are responsible for day-to-day site design and creation.

Computer scientists work as theorists, researchers or inventors. Their jobs are distinguished by the higher level of theoretical expertise and innovation they apply to complex problems and the creation or application of new technology. Those employed by academic institutions work in areas ranging from complexity theory, hardware, to programming language design. Some work on multidisciplinary projects, such as developing and advancing uses of virtual reality in human-computer interaction or in robotics. Their counterparts in private industry work in areas such as applying theory, developing specialized languages or information technologies, or designing programming tools, knowledge-based systems, or even computer games.

With the Internet and electronic business creating tremendous volumes of data, there is growing need to be able to store, manage, and extract data effectively. Database administrators work with database management systems software and determine ways to organize and store data. They determine user requirements, set up computer databases, and test and coordinate changes. It is the responsibility of an organization's database administrator to ensure performance, understand the platform the database runs on, and add new users. Because they also may design and implement system security, database administrators often plan and coordinate security measures. With the volume of sensitive data generated every second growing rapidly, data integrity, backup, and keeping databases secure have become an increasingly important aspect of the job for database administrators.

4.0 CONCLUSION

There are many dimensions to system analysts and it is clear that their role as keepers of information security is critical to effective organizational development and progress.

5.0 SUMMARY

In this unit you have learnt about the variety of systems analysts and their changing roles in organizational maintenance.

6.0 TUTOR MARKED ASSIGNMENT

- As someone being trained as a computer engineer, what do you think distinguishes a systems analyst from a programmer?
- Explain at least five characteristics of a good systems analyst.
- Systems analysts are increasingly being called upon to acquire new skills. What are some of these skills?

UNIT 5 SYSTEMS THINKING

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 The Systems Thinking Approach
 - 3.3 Use of System Thinking
 - 3.4 A Better Way to Deal with Our Most Difficult Problem
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References and Further Readings

1.0 INTRODUCTION

While organizations provide a holistic picture of their activities, nevertheless they are a collage of individual, and often independent, constituent components. Systems thinking introduce the idea of breaking down organizations into their individual components for easier, and modular, analysis and improvement.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Distinguish between systems thinking and traditional forms of analysis.
- Explain the types of problems best suited to systems thinking approach.

3.0 MAIN CONTENT

3.1 Overview

Systems thinking have its foundation in the field of system dynamics founded in 1956 by MIT professor Jay Forrester. Professor Forrester recognized the need for a better way of testing new ideas about social systems, in the same way we can test ideas in engineering. Systems thinking allows people to make their understanding of social systems explicit and improve them in the same way that people can use engineering

principles to make explicit and improve their understanding of mechanical systems.

3.2 The Systems Thinking Approach

The approach of systems thinking is fundamentally different from that of traditional forms of analysis. Traditional analysis focuses on the separating the individual pieces of what is being studied; in fact, the word “analysis” actually comes from the root meaning “to break into constituent parts.” Systems thinking, in contrast, focuses on how the thing being studied interacts with the other constituents of the system—a set of elements that interact to produce behavior—of which it is a part. This means that instead of isolating smaller and smaller parts of the system being studied, systems thinking works by expanding its view to take into account larger and larger numbers of interactions as an issue is being studied. This results in sometimes strikingly different conclusion than those generated by traditional forms of analysis, especially when what is being studied is dynamically complex or has a great deal of feedback from other sources, internal or external.

The character of systems thinking makes it extremely effective on the most difficult types of problems to solve: those involving complex issues, those that depend a great deal dependence on the past or on the actions of others, and those stemming from ineffective coordination among those involved.

Examples of areas in which systems thinking has proven its value include:

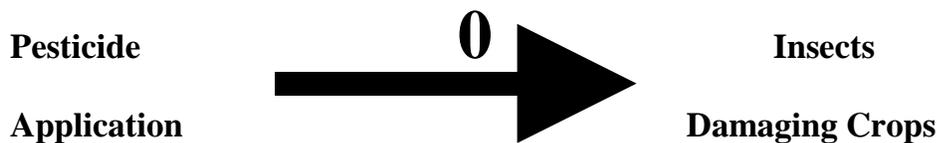
- Complex problems that involve helping many actors see the “big picture” and not just their part of it
- Recurring problems or those that have been made worse by past attempts to fix them
- Issues where an action affects (or is affected by) the environment surrounding the issue, either the natural environment or the competitive environment
- Problems whose solutions are not obvious

3.3 Use of Systems Thinking

An example that illustrates the difference between the systems thinking perspective and the perspective taken by traditional forms of analysis is the action taken to reduce crop damage by insects. When an insect is eating a crop, the conventional response is to spray the crop with a pesticide designed to kill that insect. Putting aside the limited effectiveness of some

pesticides and the water and soil pollution they can cause, imagine a perfect pesticide that kills all of the insects against which it is used and which has no side effects on air, water, or soil. Is using this pesticide likely to make the farmer or company whose crops are being eaten better off?

If we represent the thinking used by those applying the pesticides, it would look like this:



(Reading the Diagram: The arrow indicates the direction of causation - that is, a change in the amount of pesticide applied causes a change in the numbers of insects damaging crops. The letter indicates how the two variables are related: an “s” means they change in the same direction - if one goes up then the other goes up, and an “o” means they change in the opposite direction - if one goes up then the other goes down (or vice versa). This diagram is read “a change in the amount of pesticide applied causes the number of insects damaging crops to change in the opposite direction.” The belief being represented here is that “as the amount of pesticide applied increases, the number of insects damaging crops decreases.”)

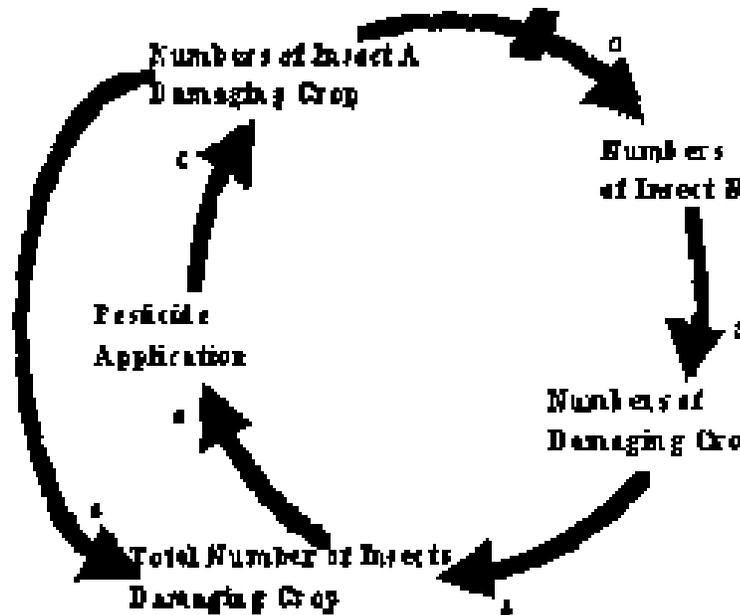
According to this way of thinking, the more pesticide is applied, the fewer insects there will be damaging crops, and the less total crop damage.

The temptation is to say that eliminating the insects eating the crops will solve the problem; however, that often turns out to not be the case. The problem of crop damage due to insects often does get better - in the short term. Unfortunately, the view diagrammed above represents only part of the picture. What frequently happens is that in following years the problem of crop damage gets worse and worse and the pesticide that formerly seemed so effective does not seem to help anymore.

This is because the insect that was eating the crops was controlling the population of another insect, either by preying on it or by competing with it. When the pesticide kills the insects that were eating the crops, it eliminates the control that those insects were applying on the population of the other insects. Then the population of the insects that were being controlled explodes and they cause more damage than the insects killed by the pesticide used to.

In other words, the action intended to solve the problem actually makes it worse because the way its unintended side effects change the system ends up exacerbating the problem.

In fact, some studies suggest that a majority of the 25 insects that cause the most crop damage each year became problems to begin with because of exactly this cycle. Graphically, the way this happens can be represented as:



According to this understanding, the greater the pesticide application, the smaller the numbers of Insect A (the original pest) that will eat the crop. This leads to an immediate decrease in the numbers of insects eating the crop (note that this is the effect those applying the pesticides are intending). However, the smaller numbers of insect A eventually lead to greater numbers of Insect B (the hash marks on the arrow indicate a delay), because insect A is no longer controlling the numbers of insect B to the same extent. This leads to a population explosion of insect B, to greater numbers of insect B damaging crops, and to greater numbers of insects damaging the crop, exactly the opposite of what was intended. Thus, although the short-term effects of applying the pesticide were exactly what was intended, the long-term effects were quite different.

With this picture of the system in mind, other actions with better long-term results have been developed, such as Integrated Pest Management, which includes controlling the insect eating the crops by introducing more of its

predators into the area. These methods have been proven effective in studies conducted by MIT, the National Academy of Sciences, and others, and they also avoid running the risk of soil and water pollution.

The way that the broader perspective of systems thinking creates the understanding necessary for better long-term solutions was also evident in the work I did with a company whose industry was being deregulated. They seemed to be doing everything right in working on a customer-relations problem they were experiencing: they had a team of capable people working on it, they were using a process that had been successful many times in the past, and they even had affected customers giving them feedback on proposals to rectify the situation.

However, they were having difficulty seeing the big picture of how the way they historically did things was contributing to the problem. Working together over two days, I was able to help them see how the problem was being exacerbated and the most powerful actions they could take to solve it. The session ended with the creation of a strategy for addressing the problem that was unanimously supported by the team and the customers.

By seeing the whole picture, the team was able to think of new possibilities that they had not come up with previously, in spite of their best efforts. Systems thinking have the power to help teams create insights like these, when applied well to a suitable problem.

3.4 A Better Way to Deal with Our Most Difficult Problems

So many important problems that plague us today are complex, involve multiple actors, and are at least partly the result of past actions that were taken to alleviate them. Dealing with such problems is notoriously difficult and the results of conventional solutions are often poor enough to create discouragement about the prospects of ever effectively addressing them.

One of the key benefits of systems thinking is its ability to deal effectively with just these types of problems and to raise our thinking to the level at which we create the results we want as individuals and organizations even in those difficult situations marked by complexity, great numbers of interactions, and the absence or ineffectiveness of immediately apparent solutions.

4.0 CONCLUSION

Systems thinking introduces to the idea of a complex and interconnected organizational structures, and the need to focus attention on the components of the system, so that the problems of the system as a whole are better diagnosed.

5.0 SUMMARY

In this unit, you have learnt that systems are made up of separate components, although, their facade is as a singular concept. Systems thinking attempt to force analyses along component lines.

6.0 TUTOR MARKED ASSIGNMENT

- Explain at least three areas in which systems thinking has proven its value
- In the unit an example of the use of systems thinking was given using pesticides in Agriculture. Using the same example, come up with another application of systems thinking in another discipline.

MODULE 2 SYSTEMS APPROACHES

Unit 1	Structured Information Systems Development
Unit 2	System Analysis and Design In End User Developed Applications
Unit 3	System Development Life Cycle (SDLC)
Unit 4	The System Development Matrix
Unit 5	Requirements Management

UNIT 1 STRUCTURED INFORMATION SYSTEMS DEVELOPMENT

CONTENTS

1.0	Introduction
2.0	Objectives
3.0	Main Content
3.1	Overview
3.2	The Software Crises
3.3	Why Use Structured Techniques
3.4	How do we achieve these Objectives?
4.0	Conclusion
5.0	Summary
6.0	Tutor marked Assignment

1.0 INTRODUCTION

When the application of computerized IS began to be applied to new areas for many different kinds of reasons, there were no recognized, formalized methods by which systems could and should be developed. Each programmer had his own informal, unique and rarely documented methods by which his software was developed. This diversity of methods made maintenance difficult, if not impossible, especially in those instances where the original programmer had left the organization. Structured techniques were introduced in an effort to overcome, minimize or avoid these types of problems.

2.0 OBJECTIVES

- Explain the rationale of introducing structured techniques of information systems development.
- List the primary objectives of structured techniques

3.0 MAIN CONTENT

3.1 Overview

So far, we have talked about systems, information systems, and organizations (viewed as systems). This module is largely concerned with how we go about creating (or developing) information systems, in particular large and complex information systems. One approach to developing information systems is the Structured Systems Development approach. There are many other approaches, but the structured approach is very common, and although not ideal, many of the others are variations of it.

3.2 The Software Crisis

Early computer systems' development was fraught with problems. The term, 'software crisis', was coined as far back as the 1960s to describe the state that the software industry found itself in. It was characterized by systems which:

- were delivered much later than scheduled.
- cost more than the budget allowed.

More seriously however, was that the delivered systems:

- rarely met user requirements.
- remained unused.
- were discarded at considerable cost to the organization.

Many of the problems came as a result of the decrease in hardware costs, which has occurred over the years. With hardware, which had previously been the more expensive component of an information system, now at such a low cost, information systems, which before had been too expensive to

implement, became feasible. This also meant that it was feasible to introduce computers to all areas of an organization. The result of these changes led to major software problems. Because the hardware in use was more powerful and more complex than before, bigger and more complicated software systems were needed to drive it. And, because hardware was less expensive, it became economically possible to apply software solutions to more and more widespread problems. Computerized IS began to be applied to new areas for many different kinds of reasons, for example:

- The current system might be unable to cope.
- They could help save money.
- They could provide better information for decision-making.
- They could provide better customer service.
- New technology could provide greater opportunities for organizations.

At that stage, the process for developing IS was, in effect, simple programming and most development work was done by programmers, who, although skilled in technical matters, were far from successful in social and communication skills; skills which are essential in finding out what the potential user wants from the new system. New systems were being developed in an ad hoc manner, whenever an organization saw the need for one. The programmer came up with a solution him or herself and started programming the new system. The user was generally not involved, which meant that systems were often inadequate or inappropriate for the users' needs.

At that stage also, there were no recognized, formalized methods by which systems could and should be developed. Each programmer had his own informal, unique and rarely documented methods by which his software was developed. This diversity of methods made maintenance difficult, if not impossible, especially in those instances where the original programmer had left the organization.

As time went on, systems grew in size and complexity, making them more difficult to be developed by a single individual programmer. In addition, ad hoc, one-off solutions to problems were becoming less attractive and managers wanted more integrated systems.

As this situation got steadily worse, with programmers overworked and spending much of their time on maintenance and enhancements instead of on the more profitable development, the realization dawned that things

would have to change. It was recognized that errors discovered early in the development life cycle were much less expensive to fix than those which came to light in testing or implementation. It was becoming apparent that there was more to development than just programming. Development had to concern some analysis of the problem and design. More importance, therefore, was attached to the analysis and design stages of the development process and the idea of approaching development, using some structured, formal technique was established.

3.2 Why Use Structured Techniques?

Structured techniques were introduced in an effort to overcome, minimize or avoid the types of problem mentioned previously. The primary objectives (there are others) of structured techniques are:

- achieve high-quality programs of predictable behavior.
- build easily maintainable programs.
- speed up system development.
- lower the cost of system development.

3.3 How do we achieve these Objectives?

To combat the problems inherent in the software crisis, a structured approach to development was felt to be the solution. Most large computer-based information systems are now developed in a sequence of steps called the **Systems Development Life Cycle** (or simply the systems life cycle). The key point is that it provides system builders with a **structured** approach to follow. It is used to:

- organize the large number of activities which are necessary in building a system.
- specify an orderly way to proceed through these activities.
- make it easier to solve problems as they occur in the development process.
- produce reports on project status, resource usage etc. as development progresses.

A number of different kinds of life cycle are used in practice, the most common systems development process takes an essentially **linear approach**, dividing the necessary tasks into a number of stages. No stage in the sequence can commence before the previous stage has been completed. There is little consensus on the terminology, the subdivision or the exact content of each stage of the life cycle. In the next unit we will look at what systems analysts and designers do, before going into their methodologies.

4.0 CONCLUSION

Structured techniques of information systems development bring an integrated order into the providing of information services through the application of many facets of systems development. It is no longer enough that programmers can come up with a programmed solution to information problems in organizations; they must do it in a way that is effective, logical and functional, in other words, structured.

5.0 SUMMARY

In this unit you have learnt about the structured techniques of system development and why such techniques are important to system developers.

6.0 TUTOR MARKED ASSIGNMENT

- What are the new areas to which computerized information systems are applied?
- What links the SDLC with structured approach to Information systems development?
- What are the primary objectives of structured techniques?

UNIT 2 SYSTEM ANALYSIS AND DESIGN IN END USER DEVELOPED APPLICATIONS

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Origins of End-user Computing
 - 3.3 Growth of End-User Computing
 - 3.4 Concerns with End-user Computing
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

Having looked at various components of organizational development — in effect applying our own systems thinking approach — we will now begin to focus attention on the software design processes, starting with the concept of End User Developed Applications (EUDA).

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the factors that contribute to the rapid expansion of end-user computing
- Explain the end-user discontentment with traditional MIS solutions

3.0 MAIN CONTENT

3.1 Overview

End-user computing has enjoyed rapid expansion during the last few years for many reasons, most of which have to do with the recent introduction of fourth generation languages, the continuing sophistication of the end-user, and the proliferation of inexpensive, but powerful personal computers; all coupled with the dissatisfaction of users with traditional data processing

capabilities. But this rapid growth has not taken place without its share of concerns; such as security of data, documentation of programs, and compatibility of hardware and/or software.

3.2 Origins of End-user Computing

End-user computing, the development of computer applications by non-data processing individuals (usually using fourth generation languages in the microcomputer environment), is growing at a tremendous rate. End-user computing has enjoyed such rapid expansion for many reasons, most of which have to do with the dissatisfaction of users with traditional data processing capabilities. Thus a systems development life cycle (SDLC), which uses structured tools and methods, was adopted by most firms as a framework for the orderly development of information systems in the 1960s.

Today similar solutions must be developed for the end-user computing environment. However, this is not a simple process. Although traditional and end user developed systems are similar in that they have the same goal of finding computer solutions to user problems, there are many differences that make many components of a traditional DSLC unnecessary in the end user environment. New processes must be developed to meet the specific needs of the management of end-user computing. The focus of this unit will be an examination of the problems in the user environment and how a user-oriented SDLC should be developed to solve those problems.

3.3. Growth of End-User Computing

Today there is growing discontent among users because they perceive MIS departments as unable to deliver a system to the user as MIS said it would, that is on time and within the predicted cost estimate. Other complaints deal with the inability of the MIS department to speak to the user in a language he can understand and the unwillingness of MIS to let the user play a significant role in project management. Many users feel that MIS has lost sight of the fact that they are a support function that was created to serve the customer (the user).

Perhaps of greatest concern for users is the time required to complete an application because of the backlog of requests that can delay a project for years. Even when these projects are completed, less than half of the systems developed by MIS are useful for decision-making purposes, which is what most mid and upper-level managers need. The systems that many users

want (modeling, simulation, and statistical analysis systems) are not requested because they are afraid of increasing the MIS backlog.

This invisible backlog for development of applications can be many times greater than the known backlog. For these reasons, end-user computing offers an attractive alternative to the user. However, it has only been recently that users could even consider an alternative to traditional application development through the MIS department. There are five factors that have been the main contributors to the rapid expansion of end-user computing.

- Technological Improvements
- Increased awareness on the part of users
- Economic conditions calling for productivity and cost-effectiveness
- Needs not satisfied by EDP departments
- Independence

Technological improvements have come about because of powerful personal computers that are available at continually lower cost, the increasing availability of sophisticated and easy-to-use software, and the development of fourth generation languages. Along with this superior technology, the knowledge of the user has also improved because of greater exposure to computers in schools and the awareness of the benefits of computer technology. Knowledge is power, and the user is putting his knowledge to work. Driven by the growing discontent with traditional solutions, which have failed to meet users' needs, users have used their knowledge and the technology available to them to develop viable solutions. As users rebel and are successful in their efforts, others have joined them in ever increasing numbers. Most have found increased satisfaction in their new independence. However, the many advantages that can be realized through end-user computing can also create their share of problems.

3.4 Concerns with End-user Computing

Many feel that the major concern with the growth of end-user computing today is that it is following the same path as the growth of computing in the 1960's: unmanaged expansion. Research found that there currently exists no general understanding of how this growth is taking place and no framework for the occurrence of responsible planning and management. It

is important to note that his lack of planning and management is prevalent even though end-user computing is consistently ranked as one of the top ten concerns by IS managers. This situation may be understandable given the limited resources of IS managers and their already over-burdened demand for traditional application development.

One of the main reasons that end-user computing is not being managed is that many upper level managers feel that too much interference will stifle its growth. They realize that their MIS departments are overtaxed and they see end-user computing as a way to shift the workload. These managers also feel that end-user applications can be developed in less time. But unfortunately, many of them also think that the use of a fourth generation language eliminates the need for proper systems analysis, design, testing, and documentation. Yet too much control over end-user computing may inhibit its growth, but too little control also has its share of concerns. These include:

- Poor documentation of programs.
- Lack of physical security over hardware and software.
- Lack of cost justification for hardware and software.
- Lack of adherence to management policies and standards.
- Incompatibility.
- Different people solving the same problem without communicating.
- Unreliability.

Of these problems, control over the security of the data is of major importance. In the end, user PC environment data is often stored on floppy disks in a plastic box next to the computer or on a hard disk that has no security method. For many user applications data is entered manually into the computer from a hard copy of a report from some other source. This data is not only unprotected from getting into the wrong hands, but the loss of data is also a very real threat. For a number of reasons, backup of data by end users is almost non-existence. Although users may not see the importance of data security and integrity, it is actually a mishandling of a major corporate resource.

Finding a solution to these many problems, while at the same time allowing for the growth of end-user computing, is a major concern for information system managers. Most IS managers believe that end users should follow the same standards and procedures that are required when data processing personnel develop computer applications, although the standards and procedures should be modified to account for differences of the creators of the systems. The next unit will present a traditional approach to systems analysis and design.

4.0 CONCLUSION

Traditional computer applications that were being developed in the 1960's, shared worries and concerns of security of data, documentation of programs and compatibility. Problems with those systems were largely solved by the development of structured analysis and design methodologies.

5.0 SUMMARY

This unit has examined the problems in the user environment and how a user-oriented SDLC should be developed to solve those problems

6.0 TUTOR MARKED ASSIGNMENT

- What is the most over-riding importance in the end-user computing?
- Explain at least three concerns in end-user computing.

UNIT 3 SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Goal of SDLCs
 - 3.3 Will the SDLC Work in the End-User Environment?
 - 3.3.1 The Monopolist
 - 3.3.2 The Laissez-Fair Approach
 - 3.3.3 The Information Centre Approach
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

The emergence, or shall we say the triumph, of end-user computing provides a vehicle for a more structured development of System Development Life Cycle (SDLC) as a core component concept of system analysis and design.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the goals of SDLC
- Graphically show the relationship of components requiring information linkage in a SDLC setup
- Discuss the three managerial approaches to SDLC

3.0 MAIN CONTENT

3.1 Overview

Use of a system development life cycle (SDLC) is regarded as a proper, disciplined approach to the analysis and design of traditional computer

applications. The number of SDLCs is almost as numerous as the companies that use them.

3.2 Goal of SDLCs

However, all SDLCs have the same goal and are fairly similar in the general way they achieve that goal. An eight-phase process explains how SDLC work. These eight steps are summarized below:

- Survey the situation.
- Study the current system.
- Define user requirements.
- Evaluate alternative solutions.
- Design the new system.
- Select new computer equipment and software.
- Construct the new system.
- Deliver the new system.

This whole process is started when a project request is submitted by someone within the firm because of a problem he/she has encountered that would appear to be able to be solved by a computer application. Once this has been done, the first step, survey of the situation is begun.

The purpose of this phase is to decide if resources of the firm should be committed to this project. A preliminary cost/benefit analysis is done to determine if the project is feasible and can therefore be approved for further development.

If the system is feasible and is important enough for the firm to commit its resources, then the current system is evaluated. This phase will help determine where problems in the current system exist and where opportunities for improvement can be found. A data flow diagram (see **Unit 21** for more on flow diagrams) is constructed that traces the flow of documents and reports in the current system. Depending upon the size and complexity of the project, phases may be combined or may overlap. See Fig.12.1

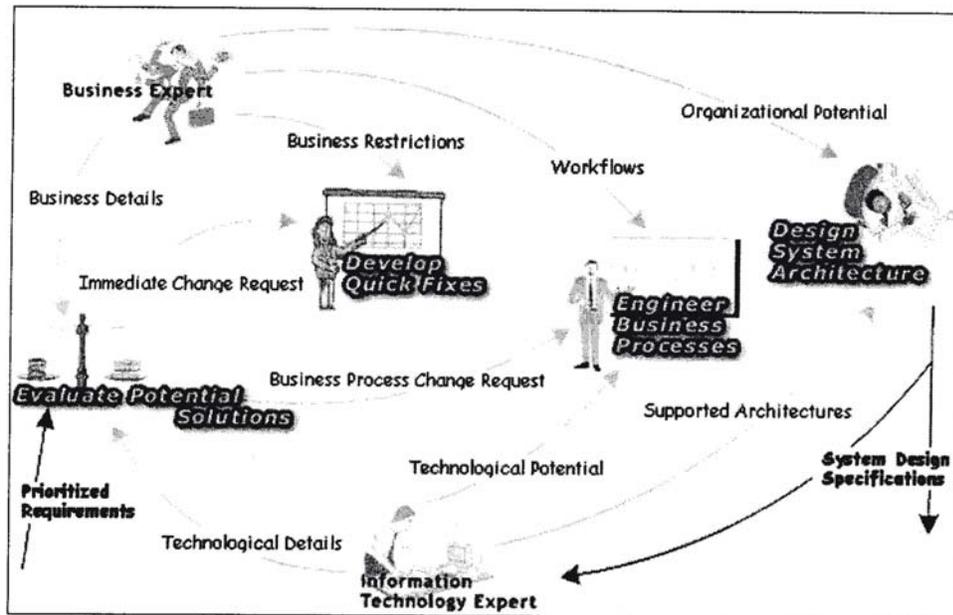


Fig 12.1: Services Supporting SDLCs

Next, user requirements are determined. Users of the current system are questioned to give the analyst (the person developing the system) an idea of what users will expect the new system to do. During this phase, inputs, files, processing, and outputs for the new system are defined along with policies and procedures that will need to be implemented. The requirements, which are written down in a formal document are then used to continue the SDLC. Alternative solutions to the problem must be evaluated to help specify how much of the new system will be computerized, how data will be accessed (online or batch), whether the system will be centralized or distributed, etc. Alternatives will also be evaluated for technical, operational, and economic feasibility. Once this general evaluation is under way, then the detailed design of the new system can begin along with the search for new computer equipment and software.

In the design phase, the specific inputs, outputs, files, methods, and procedures (i.e. backup of files, security) are determined. These components are all gathered in a form which will be understandable to the computer programmer when he begins to construct the new system.

During the construction phase, any software packages that have been purchased will be installed and modified along with programming any custom applications. As the programmers construct the system it is continually being tested to assure it is operating properly.

Finally the new system can be delivered to the user. However, before the new system is ready for use, files must be converted and the users must be trained.

Because this type of SDLC has been in existence for several years, many tools have been developed that can be employed to help make the development process proceed smoothly and to make sure that the new system is well documented. The use of an SDLC such as this has helped to make sure that the development of traditional systems are not plagued with the problems that exist in the current end-user environment.

3.3 Will the SDLC Work in the End-User Environment?

What is immediately apparent from the preceding discussion is that a complete development of an SDLC is a very involved and time consuming process. In many instances, the use of structured tools have had a lack of support even by data processing workers who feel they take too much time to develop. The problem with a lengthy SDLC is that the user has a full-time job that he must be concerned with and does not have time to devote to such a major endeavor. However, several aspects of the end-user environment that are different from the data processing approach may permit streamlining the traditional SDLC.

First of all, steps one through four of the SDLC presented here would appear to be already accomplished, because the user is the person that is developing the application. The user knows he has a problem and has a fairly good idea that a computer solution would work. The need to survey the situation or study the current system is not as great because the user, more than anyone else, is most familiar with the system. He also has a fairly good idea what his requirements are for the new system and has probably evaluated some alternatives to his problem.

A second major distinction is that the user is less interested in developing a solution that will work perfectly the first time. His approach is most likely to develop a solution quickly (a prototype) that approximates what he envisions the ultimate capabilities will be. As he continues to use the application, it will be updated to account for any deficiencies. Therefore, a very structured design phase is also undesirable.

Third, after the user has decided on his first application and bought the hardware to meet his needs, he will probably have no need for additional equipment. Most subsequent applications will be developed on the same hardware. This should also hold true for the software that he will use for his

application. The typical user will learn one or more sophisticated software packages have been purchased, there should be very little need for further software expenditures.

Although most companies have realized that this environment calls for a unique approach to an SDLC, there has been great difficulty in attempting to implement a plan that addresses these issues. The major problem in this endeavor is trying to decide who will make the decisions concerning the development and control of end-user computing. Usually both the MIS department and end users each feel they are most qualified to make these decisions. The MIS department argues that they are the ones qualified for the job. After all, they are the most knowledgeable about computer applications and how to solve the concerns involving end-user computing. The end users, on the other hand, struck out on their own in the first place because they were extremely dissatisfied with the solutions the MIS department had to offer. They now want control of their own environment. To resolve this conflict most firms have implemented one of three managerial approaches: the Monopolist, Laissez-Faire, and Information Center Approach.

3.3.1 The Monopolist Approach

The monopolist approach is used by firms who wish to keep a tight control over end-user computing. The Information System (IS) organization is given complete control of all computing resources, including both mainframe and micro computers. End-user computing is only allowed to grow at a very slow, controlled rate that is regulated by policies of the IS organization. For example, the acquisition of each personal computer is thoroughly researched to determine if it is really necessary and all applications must be approved and developed by the IS department. Firms that believe that the IS organization should have control of all information processing systems will adopt this approach.

This “go slow” arrangement solves many of the problem of end-user computing such as security, inadequate documentation, and incompatibility, but fails to satisfy the needs of the user. The huge backlog of applications that still persists and the inability of the user to get the exact system that he wants are problems that cannot be solved by using this approach.

In the past, this approach has failed or discontinued for several reasons. As was just mentioned, users are still faced with an enormous backlog of applications which causes some of them to circumvent the system and buy their own computers rather than go through the formidable IS department.

Another reason is that due to the constantly declining cost of microcomputers, any intensive research to justify their acquisition may cost more than the hardware itself. A third reason is that users are becoming more knowledgeable and capable of developing their own systems. They see no reason to wait months or years for an application they know they can develop themselves in only a few days. Finally, tight controls and extensive documentation are unnecessary for some applications the user could build quickly and use only one time.

This approach is basically trying to impose a traditional SDLC approach on the end-user environment. This assures that concerns about end-user computing will be dealt with. However, little has changed from the traditional approach and the user is probably no better off than when the IS was developing all applications.

3.3.2 The Laissez-Faire Approach

The laissez-faire approach is almost the exact opposite of the monopolist approach. In this system, departments within the organization are each given their own budgets for end-user computing needs. It is felt that each department will make better use of this money in acquiring systems tailored to their needs than by letting a central organization do it for them. IS managers have no say in the growth of end-user computing. While this approach allows for extreme creativity and perhaps greater short term user satisfaction it also has some major drawbacks.

The greatest problem with this approach is that there are many benefits that will be missed because no central organization is established. For example, the firm may miss out on quantity discounts offered by hardware and software vendors. Also, there may be extensive duplication of effort by many users developing the same systems because they are unaware of other similar applications. Not only might duplication exist, but an application that could be of major benefit to the whole firm may go unnoticed. Compatibility could also become a major concern if computer links between departments are to be established in the future. In the end the laissez-faire approach may fail because it proves too costly.

What happens in this environment is that an SDLC ceases to exist in any form. Users with very little training have not been taught and do not see a need for any type of an SDLC. There is usually insufficient/little backing up of files to protect information, no security measures for data, as well as other problems that have been outlined earlier. Although initial satisfaction of end users will probably be very high, it will likely diminish over time as

they become frustrated because applications that need to be changed have not been well documented or because they are continually losing important information.

3.3.3 The Information Center Approach

The Information Center (IC), which has been implemented by many firms and is known by many names in different organizations, is in the best position to deal with the problems concerning end-user computing. Those firms who have used the IC to its potential have achieved good results. The main duty for information centers has been to purchase hardware and software that meets the needs of the end user while maintaining standards that ensure compatibility and a good use of resources. They determine what products will be used, where they will be bought, and how they will be maintained.

Many IC managers in large corporations are also initiating policies and procedures to guide user development. For instance, companies are establishing guidelines to assure that systems are compatible which will help to guarantee that microcomputers and main-frame-based data resources and network facilities can be linked (e.g. such as the banks in Nigeria are currently doing).

The IC has also been a catalyst in ensuring that data which is accessed by users from corporate databases is secure. This has been accomplished through policies that limit the types of applications that users are allowed to develop and what data they are allowed to have access to.

While it is true that ICs have shown much promise in dealing with problems concerning end-user computing, many corporations are seeing the demise of the Information Center. One reason is that in these organizations, strong initial support has begun to wane over time. The level of support is withdrawn by various means such as smaller budgets, less adequate facilities (or being forced to share facilities with others), and smaller staffing. Usually support is diminished when the belt must be tightened throughout the corporation. Since the IC is one of the newest departments in most organizations, it is also one of the first to be cut.

Another reason for the decay of the IC is their status as a support function. In many cases they are able to establish policies, but are unable to enforce them. Many times users are unaware that these policies even exist. This support role also makes it difficult for them to cost-justify their resources to corporate management. Because the IC is usually created out of the MIS

department, it is often unsure of exactly what its role should be. This is a difficult time for MIS people who must switch from the role of owner and operator of all corporate computers to the role of facilitator of the use of computers by users. This link with the MIS department has created other problems as well because of negative attitudes that persist toward users.

The information center has been the most recent structure set up by firms to address the end-user dilemma in an attempt to exploit its benefits. Of the three major management approaches, it has met with the most success. In this environment there is a realization that the uniqueness of end-user computing calls for a different approach than the traditional SDLC. Users are trained on the importance of backing up files, security of information, and documenting their programs. Once they have realized this importance, both the IC and the users can work together on developing an SDLC that meets the unique needs of the end user environment. Thus SDLC is very different from a traditional approach. It is much less structured and consists of policies that must be followed by users who develop applications. What should make this system work where others have failed is that users will be involved in these decisions and have a voice in the development of the policies that govern them.

4.0 CONCLUSION

The various managerial approaches serve to highlight the involvement of users in the development of software to solve information system problems. While there is no emphasis in this unit on the superiority of any particular type of approach over others, nevertheless the mere concept of SDLC means evaluating an approach and weighing its effectiveness as a problem-solving strategy.

5.0 SUMMARY

In this unit you have learnt about the Software Development Life Cycle (SDLC), a more integrated approach to system development than the traditional approach used by programmers. The strongest component of SDLC is the involvement of users in the final design of the system.

6.0 TUTOR MARKED ASSIGNMENT

- What are the goals of SDLC
- Discuss the three managerial approaches to SDLC

- What are the advantages and disadvantages of each of the three managerial approaches?

UNIT 4 THE SYSTEM DEVELOPMENT MATRIX

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 The System Framework
 - 3.3 The Use of Model
- 4.0 Conclusion
- 50 Tutor Marked Assignment

1.0 INTRODUCTION

An information or decision support system follows a definite life cycle. This life cycle, which has been mentioned in the previous unit as a rationale to SDLC, includes planning, analysis, design, and implementation. The components of the life cycle are similar to the steps of systems analysis and design.

2.0 OBJECTIVES

By the end of this unit, you should be able to.

- Explain at least three stages of data processing.
- Describe the types of models used to explain data processing.
- Justify the most appropriate schematic model for data processing.

3.0 MAIN CONTENT

3.1 Over view

The planning phase of the system life cycle includes setting organizational objectives, setting the overall mission of the information and decision support system, analyzing environmental factors, and forming an overall strategy for developing the information and decision support system. Analysis includes investigating current and future needs and the strengths and weaknesses of the existing system. Design involves setting the

objectives of the new system and developing both a general and detailed systems design. Implementation includes preparing, installing, reviewing, and evaluating the system in terms of organizational objectives. In many cases, the review and evaluation performed during implementation lead to new planning and a new beginning for the system life cycle.

3.2 The Systems Framework

The six stages of data processing system (DPS) growth include: (1) initiation, (2) contagion, (3) control, (4) integration, (5) data administration, and (6) maturity. All organizations start at stage 1 and progress to later stages of data processing growth. Depending on the data processing growth and experience, different organizations have different approaches to implementing the system life cycle.

The development and use of a DPS (Data Processing System), an MIS (Management Information System), or a DSS (Decision Support System) is based on a systems framework or foundation. The systems approach, the use of models, the model of the organization, and the theory of management will be discussed in this section and related to DPS, MIS, and DSS.

3.3 The Use of Models

The real world is complex and dynamic. Due to this fact, the systems approach normally uses models of the system under investigation instead of the system itself. A **model** is an abstraction or an approximation of reality. There are a number of different types of models, and some are listed below:

1. Verbal.
2. Physical.
3. Schematic.
4. Mathematical.

A **verbal** model, as the name implies, is based on words. It is a verbal or narrative description of reality. Both spoken and written descriptions are considered verbal models. Reports, documents, and conversations concerning a system are all verbal models. Examples include the following: a salesperson describing the competition to a sales manager, a report describing the functioning of a new piece of manufacturing equipment, and a statement about the economy or future sales.

A physical model is a physical representation of reality, An engineer may develop a physical model of a chemical reactor, a builder may develop a scale model or mock-up of a new shopping center, the marketing research department may develop a prototype of a new product, and a doctor may build a plastic skeleton. These are all examples of physical models. Of course, such childhood toys as dolls and model cars are also examples of physical models.

A schematic model is a graphic representation of reality. Graphs, charts, figures, diagrams, illustrations, and pictures are all types of schematic models. Actually, you have already been studying and using schematic models throughout this book. Flowcharts, HIPO diagrams, grid charts, decision tables, structure charts, and organizational charts are all schematic models. Most data processing related models are schematic.

A mathematical model is a mathematical representation of reality. They are used in data processing, accounting, finance, management science, and production management. For example, the following mathematical model might be developed to determine the total cost of a project:

$$TC = (V)(X) + FC,$$

where:

TC = Total cost

V = Variable cost per unit

X = Number of units produced

FC = Fixed cost

The above model assumes that the total cost of a project can be divided into fixed and variable costs. In other words, this model assumes that there are no semi variable costs. In developing any model, it is important to make it as accurate as possible. An inaccurate model will usually lead to an inaccurate solution to the problem.

Since a data processing system, a management information system, and a decision support system all attempt to benefit a particular organization, it is useful to develop a model of the organization. The most appropriate type to use in this setting is a schematic model.

4.0 CONCLUSION

The actual procedures used during the system life cycle depend on the organization. Organizations that have had previous experience with information and decision support systems usually have refined system life-cycle procedures. Furthermore, managers and decision makers in these firms do not need to be trained or educated about the potential and hazards of a computerized information and decision support system.

5.0 TUTOR MARKED ASSIGNMENT

- Interlink the relationships between DPS MIS and DSS concepts
- Explain the different types of models as iconic symbolizations in systems design.

UNIT 5 REQUIREMENTS MANAGEMENT

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Identifying the Requirement Owner
 - 3.3 Identifying the Requirement
 - 3.3.1 Identifying the Need
 - 3.3.2 Defining System Requirements
 - 3.3.3 Requirements Development by System Element
 - 3.4 Controlling the Requirement
 - 3.5 Requirements Baseline Management
 - 3.5.1 Changing the Requirement
 - 3.5.2 Reactive System Change Management
 - 3.5.3 Proactive System Change Management
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor marked Assignment

1.0 INTRODUCTION

A system requirement is a description of an essential capability or constraint that pertains to the product performance, functional characteristics, and/or physical characteristics. Management of these requirements is essential to successful satisfaction of the user's need.

2.0 OBJECTIVES

By the end of this unit, you should be to:

- Describe the mechanism of user requirements identification
- Explain the components of system functions
- Outline the detailed structure of baseline documents

3.0 MAIN CONTENT

3.1 Overview

Requirements management begins with the identification and documentation of the user need. This need should be expressed in user defined terms detailing both their real needs and their expectations. The essence of system requirements analysis is the translation of these needs and expectations (through a defined, repeatable, and reversible process) into unambiguous, measurable, technical requirements (e.g., specifications).

Additionally, non-technical requirements should also be developed and managed. These requirements include such things as contractual terms and conditions, license and royalty arrangements, and financial reporting requirements. For all but the very smallest projects, a formal requirements identification, analysis, tracking, and adjudication process should be used.

3.2 Identifying the Requirements Owner

As part of the project planning process, it is essential to identify the user or user surrogate with the authority and responsibility to state the user needs and expectations. This individual or organization should be able to state explicit needs, or at least describe the desired outcome if the need is met.

In cases where the identified requirements owner is unable to articulate specific needs, the project office may assist in performing the equivalent of a market or technology analysis to define the possible and feasible outcomes of applying technology to the problem stated by the user. However, the responsible user should then specifically state the need in user—not specification—terms.

The need should be captured in a formal statement of need that includes the initial concept of how the system will be used or operated. This initial concept of operations should evolve into detailed operational processes and procedures as the system is implemented.

3.3 Identifying the Requirements

Using the systems engineering management process, requirements are identified through a repeated, recursive process over the acquisition phases of a project. First, the user need is defined and evolved into a top-level requirements hierarchy (requirements architecture). As development proceeds, these requirements are assigned to functional groupings

(functional architecture), which are later allocated to system components (physical architecture). For software intensive information systems, these architectures are typically contained in the system specification, the software development specifications, and the code listings/version description documents. Data development/synchronization requirements should also be specifically addressed (data architecture).

3.3.1 Identifying the Need

A statement of need should be prepared by the user and summarized in the System Boundary Document (SBD) — which provides a basis for internal project management. Additionally, non-technical needs or unusual externally imposed constraints should also be listed in the initial SBD. In each case, the stated needs should have an owner (originator), and, if possible, a hierarchy of priority in relationship to each other. It should be stated if the needs are inflexible.

For poorly defined or unknown needs on potentially large information systems, the initial acquisition strategy associated with the system development life cycle model should be modified to include interactive or experimental needs assessments (to include extensive subsystem prototyping) prior to moving into a more formal acquisition program. Initiating formal projects with unstable user needs typically results in significant cost and schedule problems for the project.

3.3.2 Defining System Requirements

The user need is first translated into high-level technical needs that are not system specific. During the initial project phases, these technical needs are analyzed and evolved into system-level requirements. Even at this early stage of development, a formal, robust requirements traceability process is essential to allow tracing system requirements back to the stated user needs. At each phase of development, all system requirements should trace to a need and all needs should trace to system requirements.

3.3.3 Requirements Development by System Element

System functions are allocated to one or more system element: Products (software, hardware, facilities, data, materials, communications), Processes (services, techniques), and People (personnel). At each phase of development, requirements are decomposed into more detailed functions and further allocated to system sub-elements. They should also be written

as testable statements. For example, in the end, a very specific, detailed requirement is allocated to a single software module.

3.4 Controlling the Requirements

The beginning and most essential step in controlling the requirements of an information technology project is to determine what they are. Having a definitive set of technical and nontechnical requirements is critical in controlling project cost and schedule. “Requirements creep” is a common, and sometimes fatal, problem for a project. The only cure is a clear understanding of user needs and expectations.

3.5 Requirements Baseline Management

Everyone associated with the project manages requirements. However, only a selected set of requirements needs to be baseline and then formally tracked and controlled. These controlled requirements are developed incrementally over the acquisition phases and are captured in formal documentation and tool sets. These baseline documents may include:

- User Statement of Need
- Acquisition Program Baseline
- Functional Baseline (e.g., system specification)
- Allocated Baseline (e.g., software development specifications)
- Product Baseline (e.g., source code listings)

Depending on the complexity of the project, requirements baselines may be tracked, using software tools specifically designed for that purpose. These tools allow tracing individual requirements from the stated user need, through the system and subsystem specifications, down to the individual software unit or module.

Additionally, changes to key documentation that describe the basic processes used on the project should be controlled, but not necessarily through a formal process.

3.5.1 Changing the Requirements

Needs change; opportunities are discovered; errors are found; technical efforts fail; schedules slip; cost estimates are inaccurate; budgets are cut; priorities are adjusted - many things cause requirements to change. As a result, the project team should continuously assess the impact of changes on the project cost, implementation schedule, and system performance. The

key to performing these assessments is to be able to trace any requirements change back to highest requirement (user need) and forward to the lowest requirement (software module). This allows accurate identification of changes to development activities, schedules, and costs.

3.5.2 Reactive System Change Management

Requirements changes are a normal part of the development process. Since the project team must react to these issues and still keep the project within the parameters set in the Acquisition Program Baseline, most projects should establish technical, schedule, and budget management reserves. In the technical area, this is sometimes called design margin — that margin above the minimum acceptable need that is justified by design uncertainty. Similarly, schedules are designed with slack and budgets have reserve (or at least have lower priority items which can be eliminated, if needed). The key to successful reactive change management is actually successful cost, schedule, and performance reserve management.

3.5.3 Proactive System Change Management

During project development, opportunities may arise that lead to system changes. For example, a new commercial off-the-shelf product may eliminate the need for a development task or advanced state-of-the-art hardware can be applied to the project to either increase performance or reduce life-cycle costs. Proactively, the project office should perform cost-benefit analyses to assess these opportunities. However, incorporating these changes should not unacceptably increase the risk of meeting the stated user need.

4.0 CONCLUSION

Requirements management takes the concept of end-user computing a step further in fashioning out specific channels and pathways through which requirements are identified through a repeated, recursive process over the acquisition phases of a project. These revolve around the defining user needs, and being dynamic enough to reconfigure these needs as system development begins.

5.0 SUMMARY

In this unit, you have learnt the mechanism of identifying user requirements, a process that enhances our earlier discussions on end-user computing and systems thinking.

6.0 TUTOR MARKED ASSIGNMENT

- What are the mechanisms of user requirements identification
- What are the components of system functions?
- Outline the detailed structure of baseline documents
- What is a design margin?

MODULE 3 SYSTEM METHODOLOGIES

Unit 1	Project Selection
Unit 2	System Investigation
Unit 3	Systems Design
Unit 4	Cost-Benefit Analysis (CBA) in System Designs
Unit 5	Tasks in System Design Phase

UNIT 1 PROJECT SELECTION

CONTENTS

1.0	Introduction
2.0	Objectives
3.0	Main Content
3.1	Why System Projects?
3.2	Source of Requests
3.2.1	Requests from Department Managers
3.2.2	Request from Senior Executives
3.2.3	Requests from System Analysts
3.2.4	Requests from Outside Group
4.0	Conclusion
5.0	Summary
6.0	Tutor Marked Assignment
7.0	References and Further Readings

1.0 INTRODUCTION

Systems analysts do not start working on any projects they desire. They receive a lot of requests from the management for starting different types of projects. When projects are formalized, the systems analysts, under the management's direction, conduct a preliminary investigation to analyze the reasons for the request and collect various facts to respond to the request in a systematic way. Some projects are feasible, while others may not be feasible for various reasons.

2.0 OBJECTIVES

After going through this unit you will be able to:

- describe different reasons for developing new systems projects.

- state the different sources of project requests.
- discuss how to select a project out of a number of project requests.

3.0 MAIN CONTENT

3.1 Why System Projects?

Systems projects are initiated for different reasons. The most important reasons are:

(a) Capability

Business activities are influenced by an organization's ability to process transactions quickly and efficiently. Information systems add capability in three ways:

- **Improved processing speed:** The inherent speed with which computers process data is one reason why organizations seek the development of systems projects.
- **Increased volume:** Provide capacity to process a greater amount of activity, perhaps to take advantage of new business opportunities.
- **Faster retrieval of information:** Locating and retrieving information from storage. The ability in conducting complex searches.

(b) Control

- **Greater accuracy and consistency:** Carrying out computing steps, including arithmetic, correctly and consistently.
- **Better security:** Safeguarding sensitive and important data in a form that is accessible only to authorized personnel.

(c) Communication

- **Enhanced communication.** Speeding the flow of information and messages between well as within offices. This includes the transmission of documents within offices.

- Integration of business ideas: Coordinating business activities taking place in separate areas of an organization, through capture and distribution of information.

(d) Cost

- Monitor costs: Tracking the costs of labor, goods and overhead is essential to determine whether a firm is performing in line with expectations — within budget.
- Reduce costs: Using computing capability to process data at a lower cost than possible with other methods, while maintaining accuracy and performance levels.

(e) Competitiveness:

- Lock in customers: changing the relationship with the services provided to customers in such a way that they will not think of changing suppliers.
- Lockout competitors: Reducing the chances of entering the competitors in the same market because of good information systems being used in the organization.
- Improve arrangements with suppliers: Changing the pricing, service or delivery arrangements, or relationship between suppliers and the organization to benefit the firm.
- New product development: Introducing new products with characteristics that use or are influenced by information technology.

3.2 Sources of Project Requests

There are mainly four primary sources of project requests. The requesters inside the organization are: Department Managers, Senior Executives and Systems Analysts. In addition, government agencies outside the organization may also ask for information systems projects.

3.2.1 Requests from Department Managers

Frequently, department managers who deal with day-to-day business activities are looking for assistance within their departments. They are often

not satisfied with the amount of time that the staff takes to complete the job. Sometimes, they feel that the staff members are involved in duplication of work also. In this case, the manager will discuss this problem with other administrators regarding their clerical as well as processing work and persuade higher authority to approve the development of a computer based system for office administration.

3.2.2 Requests from Senior Executives

Senior executives like presidents, vice-presidents usually have more information about the organization as compared to departmental managers. Since these executives manage the entire organization, so naturally they have broader responsibilities. Obviously, systems project requests submitted by them carry more weight and are generally broader in scope also.

3.2.3 Requests from System Analysts

Sometimes systems analysts find areas where it is possible to develop projects. In such cases, they may prefer either writing systems proposal themselves or encouraging a manager to allow the writing of a proposal on their behalf. For instance, in an organization, an analyst who sees that the library information system takes more time in processing and is inefficient, may prepare a project proposal for a new library information system. By the direction of the analyst who is fully aware about the new technology that improves the existing library information-system, the librarian may initiate the development of information system to the higher authority for approval.

3.3.4 Requests from Outside Group

Developments outside the organization also lead to project requests. For example, government contractors are required to use special cost accounting systems with government stipulated features. Generally, it has been observed that new demands from external groups bring about project requests, either for new systems or changes in current ones. Project requests originated from this source are also quite important.

4.0 CONCLUSION

It is often easy to overlook the initial routing of project initiation. But it is generally accepted that someone will start the request. Often there are different requests from different sources, but all aimed at ensuring the informational efficiency of the organization. It is vital to the systems

analyst to ensure that these different requests are all catered for since they reflect the various segments of the organization.

5.0 SUMMARY

In this unit you have learnt the origins of project requests for system analysis and design in organizations. You have also learnt the rationale behind initiating such system projects.

6.0 TUTOR MARKED ASSIGNMENT

- Name some important reasons for system projects.
- Name some primary sources of project requests.
- Discuss some causes due to which a Department Manager request for development of computer based system for his office.

UNIT 2 SYSTEM INVESTIGATION

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Systems Investigation
 - 3.3 System Analysts
 - 3.3.1 General Evaluation Considerations
 - 3.3.2 Assembling the study
 - 3.3.3 Data collection
 - 3.4 Data analysis
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References and Further Readings

1.0 INTRODUCTION

During the life of any organization, there will be new problems, new needs, and a changing environment. These problems and needs call for the organization to investigate current procedures and systems. Are the existing systems adequate? Should a new system be developed, or should the old system be modified? System analysis and design attempt to answer these questions.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Identify the four basic steps of system analysis and design cycle.
- Explain the rationale of systems investigation.
- Outline the steps in a formalized evaluation procedure.
- Describe major internal sources of data.
- Explain the sequences in data collection.
- Enumerate the techniques of data analysis.

3.0 MAIN CONTENT

3.1 Overview

Whether a company is using a computer system now or is planning to use one in the future, it will undoubtedly have to make some type of computer-related purchase. This purchase could be as small as a few a couple of hundred naira for a new ribbon for a printer or as large as a few million naira for a very large and complex computer system. In general, more planning and detailed analysis is needed for larger and more-expensive computer systems. If you know what is involved and what has to be done in acquiring a large and expensive system, you will also know how to acquire less-expensive systems, equipment, software, service, and maintenance. Thus, it will be our approach to investigate the fundamental steps in acquiring a large system. Everything else will be simpler and require fewer steps.

The complete analysis and design cycle requires four basic steps:

- Systems investigation.
- Systems analysis.
- Systems design.
- Systems implementation.

3.2 Systems Investigation

The analysis and design cycle begins with a desire to find a better data and information processing system. The particular systems that will undergo analysis and design are determined during systems investigation.

The overall purpose of systems investigation is to determine whether the existing system is satisfying the goals and objectives of the organization.

In the past, analysis and design were initiated by someone in the data processing department, such as the data processing manager or a systems analyst. As managers and other non-data processing personnel learn more about the capabilities and limitations of computer systems, the initiation of analysis and design is being called for by people outside the data processing department. Companies can be bought, sold, merged, and reorganized. These types of major changes will usually result in a need for systems analysis and design.

The first step in performing systems investigation is to form an investigation team, made up of data processing personnel and top- or middle-level managers. The exact makeup of the team depends on the size of the company and nature of the investigation. This team is then charged with the responsibility of gathering and analyzing data, preparing a report on the justification of systems analysis and design, and presenting the results to top-level managers, who will make the final decision concerning the proposed analysis and design. In larger organizations, several teams will be making a systems investigation at the same time. Due to limited resources, these teams usually compete with each other. After all of the presentations have been made by the teams, top-level managers will select problems or needs that require the most attention or that will result in the greatest financial gain.

The actual data collection and analysis performed during systems investigation is similar to the type of data collection and analysis made for those projects and problem areas selected. Interviews, questionnaires, cost/benefit analysis and so forth can be used at this stage. These topics are covered in the next unit.

Once the data has been collected and analyzed by the teams, a report is made on the feasibility and cost justification of performing a complete systems analysis and design study. The study teams will try to convince top management to continue the study. Therefore, this report is important and should be carefully drafted. Even if systems analysis and design are not undertaken for a particular system, systems investigation may reveal areas that could substantially improve the system. When detailed systems analysis and design is to be undertaken for a system, the president or chairman of the board may issue a directive to all employees to support and cooperate with the analysis and design effort.

3.3 Systems Analysis

After a project has been given approval for further study, the next step is to perform a detailed analysis of the existing system. The overall emphasis is to uncover some of the inherent problems and limitations of the existing system and to determine the extent to which the existing system is achieving the organization's goals and objectives.

3.3.1 General evaluation considerations

Systems analysis starts by clarifying the overall objectives of the organization and by determining which documents and information are needed to obtain these objectives. A manufacturing company, for example, might have an objective to reduce the number of stock outs. This objective can then be translated into one or more information needs. One information need might be to have an accurate list of inventory levels for all items. If the number of items in inventory is large, other information needs can be established. For example, a manager might want a list of only those inventory items that have an on-hand inventory level of 50 units or less.

In generating the information needs, it is important not to limit the study to the capabilities of the existing data processing system. In a sense, you need to forget about how you are currently processing data and generate information needs based on what you want. These information needs can include graphic presentations, along with more traditional reports.

Analysis of a small company's computer system can be straightforward. On the other hand, evaluating an existing data processing system for a large company can be long and tedious. As a result, large organizations that are evaluating a major data processing system normally go through a formalized evaluation procedure involving the following major steps:

- Assembling of the study team.
- Collection of appropriate data.
- Analysis of the data and the existing system.
- Preparation of a report on the existing system.

3.3.2 Assembling the study

The first step in systems analysis is to formulate a team of individuals team to study the existing system, The study team is not only given the responsibility of investigating the condition of the existing system but is usually given the responsibility of performing systems design and aiding in implementation of the new system as well.

The study team usually includes the members of the investigation team that completed the initial investigation report. Depending upon the size of the problem or need to be investigated, a number of additional people are placed on the study team. The selection of these additional members is

important. Without the support and involvement of middle- and top-level managers, analysis and design may never get implemented or may never be fully utilized if implemented. This occurs primarily because managers and other employees are often reluctant to change. Today, many study teams have representatives from management, marketing, production, accounting, and finance. These individuals can help in getting acceptance of the results, in implementing the results, in negotiating for better contracts if additional hardware and software are required, and so on.

Once the study team is assembled, it develops a list of specific objectives and activities. A schedule for obtaining the objectives and completing the specific activities is also developed, along with a statement of the resources required at each stage, such as clerical personnel, supplies, and so forth. Major milestones are normally established to help the study team monitor progress and to determine if there are any problems or delays in performing systems analysis.

3.3.3 Data collection

In the systems investigation report, various problems or needs were outlined, it is the purpose of data collection to seek additional information about the problems or needs under investigation. During this process, emphasis is given to the strengths and weaknesses of the existing data and information processing system.

Data collection requires that two steps be performed in sequence. The first step is to identify and locate the various sources of data. In general, there are both internal sources and external sources. The second step is to actually collect the data. This may require a number of tools, such as interviews, direct observation, and the development of questionnaires.

The major internal sources are: (1) organizational charts, (2) forms and documents, (3) procedure manuals and written policies, (4) financial reports, (5) data processing documentation manuals, (6) top-, middle-, and low-level managers, (7) other employees of the organization, (8) staff personnel, such as lawyers, accountants, or internal consultants, (9) the data processing manager, (10) systems analysts and computer programmers, and (11) other data processing personnel. Some of the external sources include: (1) computer manufacturers' and vendors, (2) customers, (3) suppliers, (4) stockholders, (5) government documents, (6) local, state, and federal government agencies, (7) competitors, (8) outside groups, such as environmental groups or special interest groups, (9) newspapers, (10) trade journals related to the organization, (11) data processing journals, (12)

textbooks, and (13) external consultants and other professional groups. Once the data sources have been identified, the next step is to collect data that relates to the existing system.

One of the most popular and effective data collection techniques is the interview. In a structured interview, the questions are written in advance. Some structured interviews only allow for a certain type of response, such as yes or no, or a response that ranges from strongly agree to strongly disagree or from not important to very important. Other structured interviews allow open-ended responses, where the person being interviewed responds with a few sentences for each question. In an unstructured interview, the questions are not written in advance, and the person doing the interview relies on experience in asking the best questions to uncover some of the inherent problems and weaknesses of the existing system. Before the main interviews, a few sample interviews should be conducted to make sure the right questions are being asked. This is called pilot testing. When the members of the study team are satisfied with the interview process, the interviews are scheduled and conducted. In conducting the interviews, it is important to select a location that is convenient, comfortable, and private. During the interview, the person conducting it should be straightforward and should tell the respondent that all individual responses are kept confidential. Good interviewing is an art requiring experience and excellent personal relations skills.

Another data collection technique is direct observation. With this approach, one or more of the members of the study team directly observe the existing system in action. They see how data flows between the production department and the accounting department, how sales orders are processed, how stock outs are handled, the time it takes to transfer information from one department to another, and more. By direct observation, members of the study team can determine which forms and procedures are adequate and which ones are inadequate and need improvement. Direct observation, however, does require great skill. The observer must be able to see what is really happening and not be influenced by his or her own attitudes or feelings. In addition, the observer needs good writing skills to record observations for other study team members.

When the data sources are many and spread over a wide geographic area, questionnaires may be the best data collection approach. Like interviews, questionnaires can be either structured or unstructured. In most cases, a pilot study is conducted to fine tune the questionnaire. A follow-up questionnaire can also be used to capture the opinions of those who do not respond to the original questionnaire.

A number of other-data collection techniques can be employed. Again, the overall emphasis is to collect data on the existing system. In some cases, telephone calls are an excellent method. In other cases, activities may be simulated to see how the existing system reacts. Thus, fake sales orders, stock outs, customer complaints, and data flow bottlenecks may be created to see how the existing system responds to these situations. Statistical sampling and testing techniques can also be used.

3.4 Data analysis

The data collected in its raw form from the data collection stage is usually not adequate to make a determination of the effectiveness and efficiency of the existing data processing system. This data needs to be manipulated into a form that is usable by the members of the study team. The manipulation is called data analysis. Some of the most commonly used techniques are:

- Grid charts.
- System flowcharts.
- Program flowcharts.
- Decision tables.
- Structure charts.
- HIPO diagrams.
- Layout charts.

The cost of the existing system may also be investigated. Direct, indirect, fixed, and variable costs are identified and determined for the existing system. Direct costs are costs incurred as a direct result of the existing system, while indirect costs occur with or without the existing system in operation. Indirect costs include lighting, air conditioning, staff personnel, insurance costs, and so on. Fixed and variable costs can also be associated with the existing system. A fixed cost is one incurred whether the system is operating or not, while the variable costs are costs incurred when the system is in operation. Fixed costs would include the cost of equipment, hardware, software, and so on.

Report on the existing Systems analysis concludes with a formal report on the status of the system existing system. This report contains both the strengths and weaknesses of the system.

Particular attention is placed on those areas that could use improvement. This leads into the systems design stage.

4.0 CONCLUSION

System analysis is essentially a sequence of procedures, techniques or strategies aimed at creating a rapport between the systems developer and the systems user. It is essentially a diagnostic stage in systems development and gives the user the opportunity to determine, in the final analysis, the outcome of the design.

5.0 SUMMARY

In this unit you have learnt beginning steps towards system analysis. The various methods outlined all emerged from the conviction of making end-user computing more tuned to the user's needs and requirements, and the unit has taken you through the series of steps needed to fine-tune these needs and requirements.

6.0 TUTOR MARKED ASSIGNMENT

- What are the four basic steps of system analysis and design cycle?
- How would you start system analysis?
- Explain internal and external sources as sequences in data collection
- What is the role of observation in data collection?
- Explain the various data treatment techniques

UNIT 3 SYSTEMS DESIGN

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Interactive design
 - 3.3 General design
 - 3.4 Generating system design alternatives
 - 3.5 System evaluation and selection
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

While in systems analysis the focus is on identifying the needs of the organization, the emphasis of systems design is to develop a new system that helps to achieve the goals and objectives of the organization and overcomes some of the shortcomings and limitations of the existing system. If the problems are minor, only small modifications are required. On the other hand, major changes may be suggested by systems analysis. In these cases, major investments in additional hardware, software, and additional personnel may be necessary. Regardless of the complexity and scope of the new system, it is the purpose of systems design to develop the best possible system.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the process of interactive design
- Outline the various design techniques in systems
- Outline the various design considerations
- Describe procedures design
- Discuss system design alternatives

- Describe the evaluation and selection procedures for system designs

3.0 MAIN CONTENT

3.1 Overview

For smaller design projects, systems design can be done in one stage. All of the steps and procedures described in this unit can be accomplished by the same design team over a given time span. For larger design projects, on the other hand, design is broken down into two phases. During the first phase, preliminary design, the overall system is investigated from a general point of view. During the second phase, detailed design, the steps of design are done in such a way that the new system can be acquired and implemented. This means that all reports and documents have to be specified in detail, that the inputs have to be specified and located, that the exact layout of all data on storage devices has to be developed, and so on.

It is unlikely that the entire system will be redesigned. In most cases, a number of applications will remain unchanged. Because of the high interaction between various applications, integrating the new system with the old system is essential. Making sure that the newly designed system is compatible with the existing applications and systems is called designing system interfaces. This is critically important to the success of the new system.

3.2 Interactive design

Today most computer systems allow the use of interactive process considerations. With this type of processing, people directly interact with the computer system through computer terminals. The computer and the user respond to each other in a real-time mode, which means within a matter of seconds or minutes. This type of processing brings new opportunities for good and effective design. The new design approach can include: (1) menu-driven system, (2) help commands, (3) table-lookup facilities, and (4) restart procedures.

Many new computer systems are menu driven. This is similar to a menu in a restaurant, which offers you the choice of one or more food items. With a computer system, a number of processing alternatives (a menu) are provided. These alternatives could be performing inventory control, accounts receivable, accounts payable, customer lookup, general ledger, and much more. You simply pick what you want to do, and then the computer provides you with another menu. Most people without extensive

computer training can easily operate these types of systems. They simply respond to the questions the computer asks, and the computer does the rest.

Another commonly used design technique with interactive computer systems is the help command facility. Even with a menu-driven system, many people may have questions about exactly what they are doing or what response is desired from a particular question displayed on a terminal. Thus, many designers incorporate the help command into the computer program. Whenever a person is having difficulty in understanding what is happening or what type of response is expected, the command HELP or a similar command can be typed into the terminal. The computer will then respond with exactly which stage of processing the computer program is performing, what possible commands can be given by the person using the computer, and what is expected in terms of data entry.

The use of tables within an application is another very useful design technique. Most computer applications require a substantial amount of data entry. For example, a simple sales order might require a customer's number, address, telephone number, and credit rating. Typically, however, the person entering the data does not have all of this information at his or her fingertips. To simplify things, elaborate tables can be developed and used by computer programs. A customer table, for example, might contain a customer's number, name, address, telephone number, credit rating, the supplies and inventory ordered within the last year, the total naira amount of sales for the month, the total naira amount of sales for the year, and more. This table can also contain abbreviations for the customer's number, name, and address.

Then if you want to enter data about this customer, you can simply type in one or more of these items. For example, if you are entering a sales order for a company, you simply type in its abbreviation, such as ABCO. The computer will then go to the customer table and look up all the information required to make that sales order. This information is then displayed on a terminal screen for confirmation. The use of these tables can prevent wasting a tremendous amount of time entering the same data over and over into the system.

In the middle of an application, things can go wrong. There may be a temporary interruption of power, a line printer may run out of paper, a terminal may have a defective ribbon, and so on. When these problems occur, the application that is currently being run is typically shut down. With interactive processing, there needs to be a way to quickly restart the application where it left off. As a result, easy-to-use restart procedures are

developed and incorporated into the design phase. With a restart procedure, it is very simple for an individual running an application to quickly restart it from where it left off. Restart procedures save both user time and computer resources.

For small companies or minor systems, design can be almost trivial. If you are considering the acquisition of a new piece of equipment, such as a terminal, there may be little design work involved. You already know exactly, what you want. It is simply a matter of going through the acquisition steps to get the best prices and features. On the other hand, if you are designing a completely new data processing system to handle such activities as payroll, accounts receivable, accounts payable, inventory control, distribution and warehousing, and numerous reports and documents for management, the design stage can be rather long and tedious. For these larger and longer projects, a systematic approach is needed. This approach should include general design considerations, evaluation and selection procedures, and development of written contracts when equipment and services are to be obtained from outside sources.

3.3 General design

A large design effort requires a number of general design considerations, including the following:

- Organizational constraints.
- Functional design.
- Output design.
- Input design.
- Processing design.
- File and data base design.
- Procedures design.
- Personnel and job design.

The first step in design is to lay out various restrictions and constraints placed upon you by the organization. Companies have limited time to develop new systems, limited financial resources, limited manpower,

limited equipment, and limited procedures. Being too conservative can also limit the overall effectiveness of the new system. You need to be as realistic and accurate as possible in specifying true organizational restrictions and constraints.

Functional design is a layout of the major functions of the organization. For example, a wood products company may perform two overall functions—one to manufacture paper products and the other to manufacture timber products. Each of these two major functions can be further subdivided into what the company produces. A functional chart can be developed to reveal the major functions and their sub functions. This functional chart is very similar (and in some cases almost identical) to the company's organizational chart.

Once the functional chart has been developed, the desired information or output for each function and sub-function can be determined. When a large number of managers and decision makers is to be involved, small groups can be formed that analyze needs for their particular function. Each output from the computer for each functional area should be described with a great amount of detail and precision. The detailed layout of every report and output, the number of times the report is to be produced, how the report is to be distributed, and who should receive the various reports are described in detail. A short paragraph can be written describing each report, and in some cases, a layout chart or graphic picture can be used to reveal how each output should be constructed.

After managers have described in detail what they want from the new computerized system, it will be up to the data processing personnel to determine the best way to generate these reports and outputs. Managers should still be involved, however, in monitoring and controlling these activities, which include input, processing, file and data base design, and personnel selection and job design. Once data processing personnel know what managers want, they can work backwards to determine the input data needed to generate various reports and outputs. Layout charts can be used to describe the format of the input data. Next, processing design is done. This describes in detail how the input data is to be manipulated to produce the desired reports and documents. Processing design can be done with the use of such tools as flowcharts and decision tables. Once processing design is complete, file and data base design can be undertaken. File and data base design describes exactly how all the data is to be stored on the system.

Procedures design is another important and often overlooked area of designing a new computerized system. This involves developing rules,

regulations, and guidelines that are to be used while operating the new system. Computer security, crime and fraud prevention, and privacy policies are developed during procedures design as well.

Another key design consideration is personnel. For large and complex designs, new data processing personnel are usually needed to make the new system effective and operational. Personnel and job design, which is typically done by the data processing manager, describes what new people are required or how existing job titles and responsibilities are to be changed to accommodate the new computerized system.

Another important part of design is to determine which backup systems are required. In general, everything should have a backup, including all hardware, software, data, personnel, and supplies and facilities. What to do in case of a computer-related disaster should also be considered in the design phase.

The final design consideration is documentation. The documentation of the new system is very similar to the documentation of the existing system. Tools such as flowcharts, decision tables, grid charts, and so on can be employed.

3.4 Generating system design alternatives

When additional hardware and software are not required, alternative designs are generated by the members of the study team. The study team may also wish to get other personnel involved in generating alternative designs if the new system is more complex and involved than the existing system. If new hardware and software is to be acquired from an outside vendor, a formal request for proposals (REP) should be made.

There are a number of advantages to developing a formal RFP. To begin with, the RFP can make interacting with computer vendors easier. It can stipulate various ground rules. For example, most companies only have one or two people who interact with computer vendors directly. This prevents the computer vendors from trying to contact numerous managers and executives within the organization. Furthermore, one RFP can answer questions from many vendors, Without the RFP, the same questions could be asked dozens of times by different vendors. Even though developing the REP takes time, it can save time in the long run. Another advantage-is better price/performance ratios.

Vendors who know several companies will be bidding on the proposed new system often do a better-job estimating actual costs. As a result, the company may be able to obtain superior equipment for less money. The RFP can also prevent disputes and problems down the line. Typically, the RFP is included with the contract signed by the user and the vendor, protecting the company purchasing the equipment, software, and/or services.

The contents of the RFP should include some background information, the procedures that are to be followed during the selection process, a description of the existing company and data processing system, a description of the proposed system and its requirements, and a list of exactly what will be expected from the data processing vendor or manufacturer. The major part of the RFP is the description of the existing data processing system and the proposed system.

The request for proposals should be carefully written. The procedures to be followed are especially important. The computer vendors should be told which person on the study team is to be contacted, how their proposals will be evaluated, and the deadlines for proposals. In addition, the necessary support from the vendor, such as training, maintenance, installation restrictions, and so on, should be clearly stated.

3.5 System evaluation and selection

The final step in systems design is to evaluate the various systems design alternatives and to select that design which meets the needs of the new system at the least cost. Normally, a preliminary evaluation and a final evaluation are conducted before one design is selected.

When outside equipment, software, support, and maintenance are required, a contract between the organization and outside vendor is needed. Some computer manufacturers publish statistics on how fast their computer systems can perform certain operations, such as multiplying or dividing eight-digit numbers. These are called benchmark tests. Usually, computer manufacturers pick the data and the process to be done (i.e., multiplication) to make their system appear to be as fast as possible. As a result, it is very difficult to compare different computer systems using their tests and statistics. If you need to make speed comparisons, you can use the results of tests made by an independent group or organization. The best way to compare speed is to have the manufacturers perform a real application, such as payroll using actual payroll data. The same approach can be used if you are comparing storage capacity or any other system characteristics; actual

data run on real applications is the best way to compare different computer systems.

Preliminary evaluation begins after the deadline for the submission of proposals. The purpose of this preliminary evaluation is to eliminate some of the proposals. Several vendors can usually be eliminated by investigating their proposals in detail. The remaining vendors are asked to make a formal presentation to the study team. Small demonstrations may also be conducted. Furthermore, the vendors should be asked to supply a list of organizations using their equipment for a similar purpose. These organizations are then contacted and asked to evaluate their hardware, software, and the vendor. It is also a good idea to ask these companies for further references that can be contacted. After evaluating the presentations, demonstrations, and the evaluations from other companies using similar equipment, the list of vendors is usually reduced to a few.

The final evaluation begins with a detailed investigation of the hardware, software, and support offered by each of the remaining vendors. The vendors should be asked to make a final presentation and to arrange for an extensive demonstration. The demonstration should be as close to real operating conditions as possible. Such applications as payroll, inventory control, and billing should be conducted with a large amount of test data. The same data and applications should be used by each vendor. Sometimes, it takes day or longer for each demonstration. But this is one of the best ways to learn more about the capabilities and limitations of a proposed data processing system.

After the final presentations and demonstrations, it is necessary to make the final evaluation and selection. During this stage, a detailed analysis of the hardware, software, and vendor is made. Cost comparisons, hardware performance, delivery dates, price, modularity, backup facilities, available software training, and maintenance factors are considered. Although it is good to compare computer speeds, storage capacities, and other similar characteristics, this is not enough. It is necessary to carefully analyze how the characteristics of the proposed systems will help the organization or company solve problems and obtain goals and objectives.

Although the exact procedure used to make the final evaluation and selection varies from one organization to the next, there are three commonly used approaches. These are: (1) group consensus, (2) cost-benefit analysis, and (3) point evaluation. In group consensus, a decision-making group is appointed and given the responsibility of making the final

evaluation and selection. Usually, this group includes the members of the study team.

4.0 CONCLUSION

The objective of the design phase is to transform the detailed, defined requirements into complete, detailed specifications for the system to guide the work of the Development Phase. The decisions made in this phase address, in detail, how the system will meet the defined functional, physical, interface, and data requirements.

5.0 SUMMARY

In this unit you have learnt about the steps, procedures and techniques of system design. Starting from the interactive design considerations to evaluation of the system, through the application of benchmark tests, the unit has taught you about the necessary skills needed to design effective information systems.

6.0 TUTOR MARKED ASSIGNMENT

- Explain RFP and its advantages
- Why would you create backups of your system designs?
- Why should personnel be a key consideration in system design?
- What is the first step in system design?
- Explain the role of benchmarks in system design

UNIT 4 COST-BENEFITS ANALYSIS IN SYSTEM

DESIGNS

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Time Period
 - 3.3 Alternatives
 - 3.4 Types of Analysis
 - 3.5 Identifying and Measuring Benefits and Costs
 - 3.6 Decision Criteria
 - 3.7 The Mechanics of the CBA Process
 - 3.7.1 When Is A CBA Required?
 - 3.7.2 When is the CBA performed
 - 3.7.3 Who Should do the CBA
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

In a way, Cost Benefit Analysis (CBA for short) is the starting point of any system analysis. It is the procedure by which the worthwhile of a system approach is determined, without which there would be no effective idea of the costs and consequences of the new information system being contemplated.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the role of CBA in system analysis
- Outline the steps of CBA
- Explain the different types of CBA
- Explain the mechanics of CBA

3.0 MAIN CONTENT

3.1 Overview

The purpose of a CBA is to support better decision-making to ensure that resources are effectively allocated to support the missions of the organization. The CBA should demonstrate that at least three alternatives were considered, and the chosen alternative is the most cost-effective within the context of budgetary and political considerations.

3.2 Time Period

The CBA time period should match the system life cycle. The system life cycle includes the following stages/phases:

- feasibility study
- design
- development
- implementation
- operation
- maintenance

A system life cycle ends when the system is terminated or is replaced by a system that has significant differences in processing, operational capabilities, resource requirements, or system outputs. Significant differences is a very subject term, and some organizations may feel that a 10% change is significant, while others may feel that the change must be over 30% to be significant.

3.3 Alternatives

Analyses must consider at least three alternative means of achieving program objectives, one of which is to continue with no change. This provides a comparative baseline. Other alternatives could include:

- in-house development versus contractor development
- in-house operation versus contractor operation

- leasing equipment versus purchasing equipment
- current operational procedures versus new operational procedures
- one technical approach versus another technical approach

3.4 Types of Analysis

Benefit-Cost Analysis (BCA) is a systematic, quantitative method of assessing the life cycle costs and benefits of competing alternative approaches. This includes determining which one of the alternatives is best.

A Cost-Effectiveness Analysis (CEA) is a simplified BCA which can be done when either the benefits or the costs are the same for all alternatives. The analysis is greatly simplified because the best alternative is either the one with the most benefits (when the costs are the same for all alternatives) or the one with the lowest cost (when the benefits are the same for all alternatives).

3.5 Identifying and Measuring Benefits and Costs

CBAs must include comprehensive estimates of the projected benefits and costs for all alternatives. Benefits to which naira value cannot be assigned (intangible benefits) should be included along with tangible benefits and costs. Intangible benefits should be evaluated and assigned relative numeric values for comparison purposes. For example, maximum benefit could be assigned a value of 5, average benefits a value of 3, and minimum benefits a value of 1. Evaluating and comparing benefits that have both naira values and relative numeric values requires extra effort, but it allows subjective judgment to be a factor in the analysis.

CBAs should be explicit about the underlying assumptions used to arrive at estimates of future benefits and costs. For example, the number of users of an IT system might be assumed to increase at a rate of 10% each of the 6 years of the system life cycle.

Costs incurred in the past (Sunk Costs) and savings or efficiencies already achieved (Realized Benefits) should not be considered in a CBA. When a CBA is done on a project that is already underway, there may be pressure to compare all costs and benefits from the beginning of the project. In that situation, the question to be answered is whether or not the benefits of proceeding justify the costs associated with continuing the project. The classic example of this is a situation where large amounts of money have

been spent designing a system that has not been successfully implemented, and the project is being re-evaluated. The fact that a lot of money has been spent is no reason to continue spending. CBAs focus on the future and decisions have to be based on the expected costs and benefits of the proposed alternatives. Past experience is relevant only in helping estimate the value of future benefits and costs.

3.6 Decision Criteria

Projects should be initiated or continued only if the projected benefits exceed the projected costs. The only exception is if benefits are mandated by law.

Benefit-Cost Analysis - The standard criterion for justifying an IT project is that the benefits exceed the costs over the life cycle of the project. The competing alternative with the greatest net benefit (benefits minus costs) should be selected. When all benefits and costs cannot be assigned monetary values, relative values for costs and benefits can be used, and the alternative with the greatest net benefit (benefit values minus cost values) should still be selected.

Cost-Effectiveness Analysis - When comparing alternatives with identical costs and different benefits, the alternative with the largest benefits should be selected. When comparing alternatives with identical benefits and different costs, the alternative with the lowest costs should be selected.

3.7 The Mechanics of the CBA Process

3.7.1 When Is A CBA Required?

A CBA is always required before a decision is made to initiate or continue an IT project; the only issue is the level of detail required for the analysis. The process described here is appropriate for a very large, complex, and costly IT project. Scaled down versions of the CBA would be appropriate for smaller, less costly projects; and your organization should provide guidelines to determine the amount of scaling that would be appropriate for IT projects based on their size, cost, and complexity.

3.7.2 When is the CBA Performed?

A cost-benefit analysis should occur prior to initiating or modifying an IT system. Most of the activities described below may be completed before the CBA is initiated, concurrently with the CBA, or as part of the CBA. The CBA is a key input for the investment review that should take place before a new project proceeds to the acquisition or development phase.

- **Define The Problem** - Clearly define and document the problem. If possible, it should be described from a management perspective.
- **Review the Current Work Process Documentation** - If no Documentation exists, it must be developed. If it is not clear and up-to-date, it should be updated to clearly describe the current work process. The information processing requirements must be part of the documentation for the current work process or the current IT system.
- **Evaluate the Work Process** - There are two questions to address in the work process evaluation: Should We Be Doing This? and Can the Process Be Improved?
- **Define the New Processing Requirements** - Define the information processing requirements for the proposed work process at a general level. The security requirements should be addressed in terms of data integrity, reliable processing, privacy and confidentiality.
- **Determine Its Performance Measures** - Identify indicators for measuring and assessing performance of the process and the IT system in relation to the mission of the organization. Also determine the means of collecting and storing the performance data.

The Cost-Benefit Analysis may have to be updated several times during the life cycle of a system. The first cut at a CBA may be quite brief, and can be used to get concept approval to proceed with a detailed CBA. After the detailed CBA has been completed, the development and implementation plans may call for a prototype system or a pilot phase to test the costs and benefits on a limited scale before the full system is implemented for all users. If that occurs, a third version of the CBA would reflect revised costs and benefits, and would be used to decide whether or not to proceed with full implementation of the system. The post-implementation review of a system may also require an updated CBA to determine if the expected benefits are being achieved, and to decide if the operation of the system should continue as implemented, or if the system should be modified to achieve benefits to justify continued operation.

3.7.3 Who Should Do The CBA?

One person should be responsible for ensuring that a CBA is done. However, that person will need to assemble a team with expertise in IT systems development and operation, budget, finance, statistics, procurement, IT architecture and the work process being analyzed. A team brings different perspectives to the analysis and the process of estimating costs and benefits, and should ensure more realistic estimates than those of just one person. Additionally, one person rarely has expertise in all of the areas required for a CBA and the knowledge of the work process that is being automated.

4.0 CONCLUSION

CBA, as we noted at the beginning of the unit, is the bedrock of the system analysis, for it gives the analyst an accurate idea of the costs involved in implementing systems. The outcome of the CBA would therefore inform on the range of alternatives available.

5.0 SUMMARY

In this unit you have learnt the specific steps and procedures adopted to conduct a cost benefit analysis in system designs.

6.0 TUTOR MARKED ASSIGNMENT

- Why is CBA important?
- Who should conduct CBA?
- What are the steps in CBA procedures?
- What should be done with the results of CBA?

UNIT 5 TASKS IN SYSTEM DESIGN PHASE

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Task and Activities
 - 3.2.1 Establish the Application Environment
 - 3.2.2 Design the Application
 - 3.2.3 Develop System Design Document
 - 3.2.4 Develop Maintenance Manual
 - 3.2.5 Development Operations Manual
 - 3.2.6 Conduct Preliminary Design Review
 - 3.2.7 Design Business Processes
 - 3.2.8 Design Human Performance Support (Training)
 - 3.2.9 Design Conversation/Migration/Transition Strategies
 - 3.2.10 Develop Security operating Procedures
 - 3.2.11 Conduct Final Design Review
 - 3.3 Roles and Responsibilities
 - 3.4 Deliverables, responsibilities, and actions
 - 3.4.1 Design Document
 - 3.4.2 Maintenance Manual
 - 3.3.3 User manual
 - 3.3.4 Training Plan
 - 3.4.5 Conversion Plan
 - 3.4.6 Implementation Plan
 - 3.4.7 Operations Manual or System Administration Manual
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

Since the system design is a group activity, one of the surest ways of ensuring the success is to identify and assign specific roles to each member of the development team.

2.0 OBJECTIVES

By the end of this unit, you should be able to

- List the specific tasks in the system design phase.
- Outline the roles and responsibilities of each member of the design team

3.0 MAIN CONTENT

3.1 Overview

Design Phase activities may be conducted in an interactive fashion, producing first a general system design that emphasizes the functional features of the system, then a more detailed system design that expands the general design by providing all the technical detail.

3.2 Tasks and Activities

The following tasks are performed during the Design Phase. The tasks and activities actually performed depend on the nature of the project.

3.2.1 Establish the Application Environment

Identify/specify the target environment, the development environment and the design environment.

3.2.2 Design the Application

In the system design, first the general system characteristics are defined. The data storage and access for the database layer need to be designed. The user interface at the desktop layer needs to be designed. The business rules layer or the application logic needs to be designed. The interfaces from application to application and application to database also need to be designed and documented.

3.2.3 Develop System Design Document

The system design document will be developed by the project manager and project team, identifying the steps used in the design of the application/system. The System Design Document is a deliverable in the Design Phase.

3.2.4 Develop Maintenance Manual

Develop the maintenance manual to ensure continued operation of the system once it is completed. This is a deliverable in the Design Phase.

3.2.5 Develop Operations Manual

Develop the Operations Manual for mainframe systems/applications and the System Administrators Manual for client/server systems/applications. This is a deliverable in the Design Phase.

3.2.6 Conduct Preliminary Design Review

This is an ongoing interim review of the system design as it evolves through the Design Phase. Detailed objective system functions, performance requirements, security requirements, and system platform characteristics will be reviewed.

3.2.7 Design Business Processes

The business organization, roles and procedures for designing this system/application need to be articulated.

3.2.8 Design Human Performance Support (Training)

The Training Plan and the User Manual are created during the Design Phase. These will be deliverables in the Design Phase.

3.2.9 Design Conversion/Migration/Transition Strategies

If current information needs to be converted/migrated/transitioned to the new system, plans need to be designed for those purposes, especially if converting means re-engineering existing processes. The Conversion Plan, Implementation Plan and Contingency Plan need to be designed in this phase and are deliverables.

3.2.10 Develop Security Operating Procedures

Develop procedures on how security will be handled throughout the Design Phase and who will be responsible for carrying out these procedures.

3.2.11 Conduct Final Design Review

The Project Manager and System Proponent conduct the final design review and approve/disapprove the project into the Development Phase. This review is conducted at the end of the Design Phase and confirms that modifications prompted by earlier reviews are incorporated.

3.3 Roles and responsibilities

- **Project Manager.** The project leader is responsible and accountable for the successful execution of the Design Phase. The project leader is responsible for leading the team that accomplishes the tasks shown above.
- **Project Team.** The project team members (regardless of the organization of permanent assignment) are responsible for accomplishing assigned tasks as directed by the project managers.
- **Contracting Officer.** The contracting officer is responsible and accountable for preparing solicitation documents under the guidance of the program manager.
- **Oversight Activities.** Agency oversight activities, including the IRM office, provide advice and counsel to the project manager on the conduct and requirements of the Design Phase. Additionally, oversight activities provide information, judgments, and recommendations to the agency decision makers during project reviews and in support of project decision milestones.

3.4 Deliverables, responsibilities, and actions

The content of these **deliverables** may be expanded or abbreviated depending on the size, scope, and complexity of the corresponding systems development effort.

3.4.1 Design Document

It describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interface, detailed design, processing logic, and external interfaces. It is used in conjunction with the Functional Requirements Document (FRD), which is finalized in this phase, to provide a complete system specification of all user requirements for the system and reflects the user's perspective of the system design. Includes all information required for the review and approval of the project development. The sections and subsections of the design document may be organized, rearranged, or repeated as necessary to reflect the best organization for a particular project.

3.4.2 Maintenance Manual

The Maintenance Manual provides maintenance personnel with the information necessary to maintain the system effectively. The manual provides the definition of the software support environment, the roles and responsibilities of maintenance personnel, and the regular activities essential to the support and maintenance of program modules, job streams, and database structures. In addition to the items identified for inclusion in the Maintenance Manual, additional information may be provided to facilitate the maintenance and modification of the system.

3.4.3 User Manual

The User Manual contains all essential information for the user to make full use of the information system. This manual includes a description of the system functions and capabilities, contingencies and alternate modes of operation, and step-by-step procedures for system access and use.

3.4.4 Training Plan

The Training Plan outlines the objectives, needs, strategy, and curriculum to be addressed when training users on the new or enhanced information system. The plan presents the activities needed to support the development of training materials, coordination of training schedules, reservation of personnel and facilities, planning for training needs, and other training-related tasks. Training activities are developed to teach user personnel the use of the system as specified in the training criteria. Includes the target audience and topics on which training must be conducted on the list of training needs. It includes, in the training strategy, how the topics will be addressed and the format of the training program, the list of topics to be covered, materials, time, space requirements, and proposed schedules.

3.4.5 Conversion Plan

The Conversion Plan describes the strategies involved in converting data from an existing system to another hardware or software environment. It is appropriate to re-examine the original system's functional requirements for the condition of the system before conversion to determine if the original requirements are still valid.

3.4.6 Implementation Plan

The Implementation Plan describes how the information system will be deployed and installed into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort (such as hardware, software, facilities, materials, and personnel), and any site-specific implementation requirements. This plan is updated during the Development Phase; the final version is provided in the Integration and Test Phase and used for guidance during the Implementation Phase.

3.4.7 Operations Manual or Systems Administration Manual

For mainframe' systems, the Operations Manual provides computer control personnel and computer operators with a detailed operational description of the information system and its associated environments, such as machine room operations and procedures. The Systems Administration Manual serves the purpose of an Operations Manual in distributed (client/server) applications.

4.0 CONCLUSION

Thus the system design process is approached with a studied methodological skill typical of an analyst. It is a process involving many key actors, and each member must be made to account for his role. In other words, the design team must work as a **system**, themselves.

5.0 SUMMARY

In this unit you have learnt the various tasks each member of the design team must perform in order to be part of the design team system.

6.0 TUTOR MARKED ASSIGNMENT

- Explain the roles of Contacting Officer and Project Manager. What is the difference between the two?
- What types of manuals needed to be developed in the design of systems?
- What should the Design Document look like?
- What is a Conversion Plan. How does it differ from Implementation Plan?

MODULE 4 SYSTEM DESIGN AND IMPLEMENTATION

Unit 1	System Design Tools
Unit 2	Development Phase
Unit 3	Implementation Phase
Unit 4	Operations and Maintenance Phase
Unit 5	The Design process

UNIT 1 SYSTEM DESIGN TOOLS

CONTENTS

1.0	Introduction
2.0	Objectives
3.0	Main Content
3.1	Overview
3.2	Screen Layout Charts
3.2.1	Flowcharts
3.2.2	Program flowchart
3.2.3	Problems with Flowcharts
3.2.4	Decision Tables
3.2.5	Grid Charts
4.0	Conclusion
5.0	Summary
6.0	Tutor Marked Assignment

1.0 INTRODUCTION

This unit introduces you to the traditional (and some would say, ancient!) tools of system analysis and programming. Remember, these were the methodologies used before the advent of fast compilers and affordable computing.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the rules and guidelines of flowcharting standards
- Explain the purpose of a screen layout chart
- Design a program flowchart

- Explain system flowchart symbols used in programming

3.0 MAIN CONTENT

3.1 Overview

A number of tools and techniques are available to computer programmers and systems analysts to assist in developing computer programs and applications, in this unit, some of the more traditional tools and techniques are covered. They include print and layout charts, screen layout charts, flowcharts, decision tables, and grid charts.

3.2 Screen Layout Charts

A number of forms have been developed to make it easier to specify in detail the input and output from a computer application. A print chart is used to describe the format of an output report from a printer. A card layout chart is used to describe how input data is placed on one or more computer cards. Different companies and data processing departments might use slightly different forms. In addition, there are other charts and forms that describe the exact layout or positioning of data on magnetic tape, disk, and the display terminal or CRT.

During the last decade, more and more companies have started using terminals or CRTs to perform data input instead of computer cards and card readers. Because of this, there is a greater emphasis on the use of screen layout charts rather than the traditional card layout charts. The purpose of a screen layout chart is to reveal what information and data is to be displayed on the screen during processing. The screen layout chart reveals both data to be entered into the computer as well as the information and data to be displayed on the terminal device. For some applications, there may be 100 or more different screens that can appear on the display terminal. To perform an adequate job during design and system documentation, screen layout charts should be developed for every possible screen that can appear on the display terminals.

Screen layout charts can take on a number of different forms, depending upon the detail that is to be displayed on the chart. General screen layout charts reveal an overview of what is to be displayed on the screen. A detailed screen layout chart reveals in detailed fashion what information is to be displayed on the display terminal. Screen layout charts can be done by hand or by using preprinted forms.

3.2.1 Flowcharts

Like a road map, flowcharts are used to reveal how to go from a starting point to the final destination and can be used to display any amount of detail. In developing application software, a general chart to reveal the overall purpose and structure of the application is needed.

This is usually called the system flowchart or application flowchart. More detailed flowcharts are also needed that reveal how each program is to be developed. This type of flowchart is called a **program flowchart**.

Both system and program flowcharts are drawn using a special flowcharting template, which contains a set of symbols needed in developing flowcharts. System flowcharts are some symbols are used for both system and program flowcharts, while other symbols are fairly specialized. Some of the commonly used system flowcharting symbols are presented in Fig. 21.1.

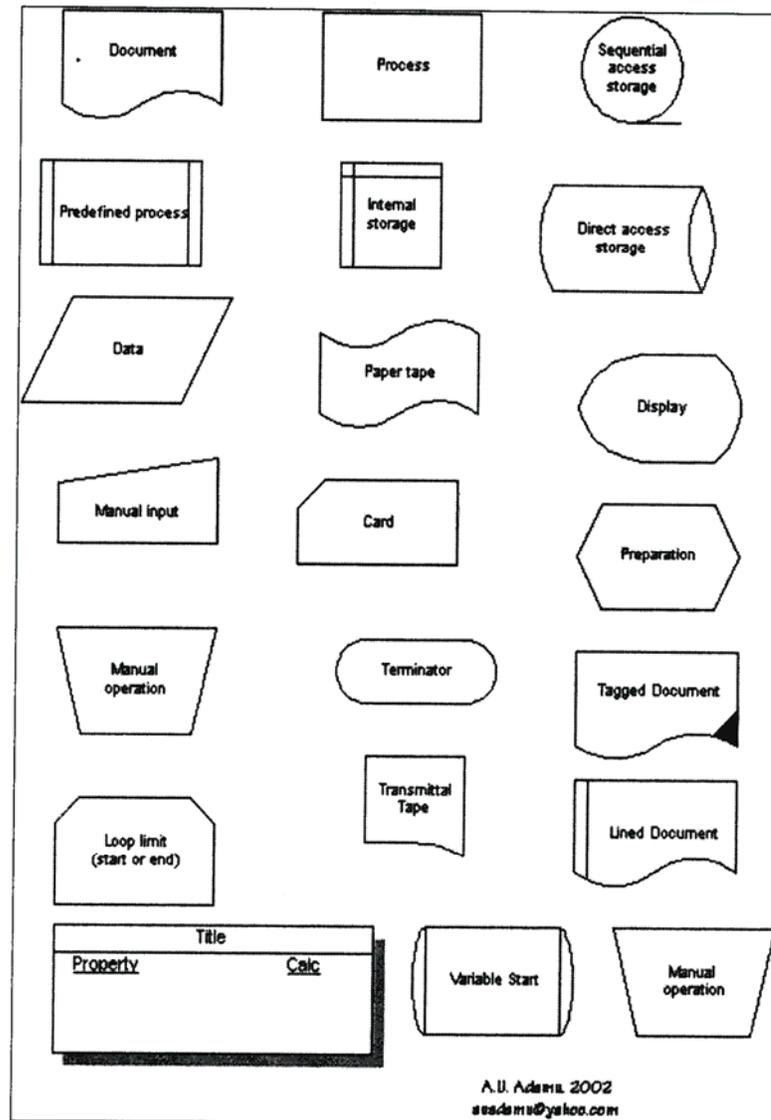


Fig 21.1 Flowcharting Templates

These system flowchart symbols are used to reveal, in a general fashion, the relationships among the input data, the programs, and the desired output. For example, most realistic payroll applications require multiple computer programs interacting with multiple data files. A simplified system flowchart for a payroll application is displayed in Fig. 21.2.

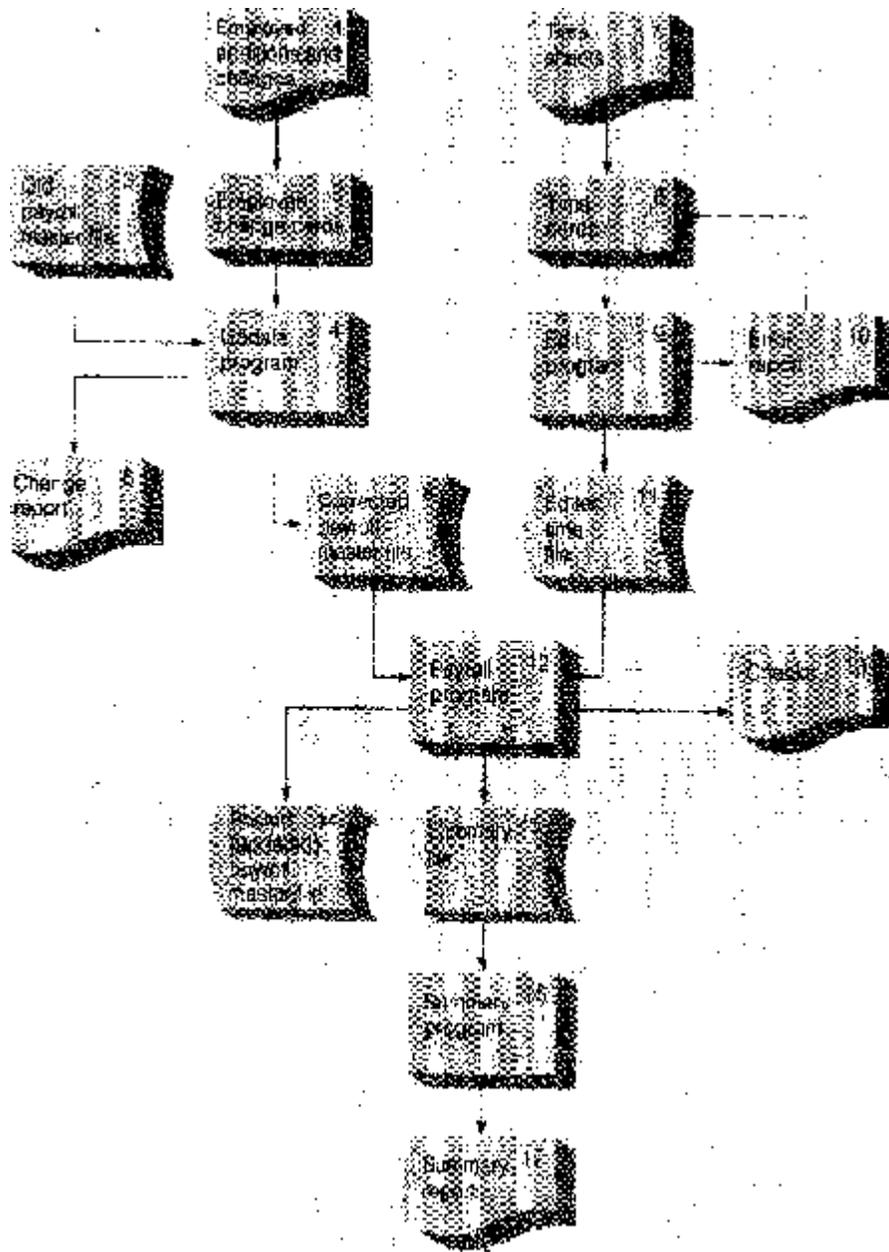


Fig. 21.2 System flowcharts for a simplified payroll application

The payroll application shown in Fig. 2 includes an update routine (symbols 1-6), an audit routine (symbols 7-11), the payroll program (symbol 12), the checks (symbol 13), the posted master file (symbol 14), and a summary routine (symbols 15-17). Most payroll applications are more complex and involved. For example, federal and state income tax withholding statements must be prepared for each employee. These forms

must then be sent to all employees to enable them to fill out their state and federal income tax forms each year.

Furthermore, most payroll applications generate additional reports. One manager might want a report containing all employees working overtime this week, and another manager might want a report listing all employees qualified to perform certain jobs. These and similar reports and outputs become a part of the payroll application.

3.2.2 Program flowcharts

Once the system flowcharts have been developed, the next step is to design and structure the computer programs. Like blueprints of a building, program flowcharts reveal in detail how the program is to be built. Program flowchart symbols are used to show activities to be performed. The flowcharting template shown in Fig. 1 is used.

A flowchart for a payroll program (refer to symbol 12 in Fig. 2) can be developed. This program should read from the updated master file and the edited transactions file. It should write checks, update the master files, and prepare a summary report. In both program and system flowcharts, there can be various levels of detail. A flowchart that shows less detail is a macro flowchart, while a flowchart with more detail is a micro flowchart, or detailed flowchart.

While programmers and system analysts have a good deal of freedom in creating flowcharts, there are a number of general rules and guidelines recommended by the American National Standards Institute (ANSI) to help standardize the flowcharting process. Various computer manufacturers and data processing departments usually have similar flowcharting standards. Some of these rules and guidelines are:

- Go from left to right and top to bottom in constructing flowcharts.
- When a reference for a symbol is needed, it should be placed above the symbol.
- If a new flowcharting page is needed, it is recommended that the flowchart be broken at an input or output file.
- Use an annotation symbol when a more detailed explanation is needed.

- Words in the flowchart symbols should be common English words and easy to understand.
- Maintain a consistent level of detail for a given flowchart.
- Keep the flowchart as simple as possible and try to avoid flowcharting lines that cross or loop.
- Be consistent in using names and variables in the flowchart.

Flowcharting books and manuals have been developed by various organizations. While these specific rules are beyond the scope of this course, good flowcharting procedures require they be followed.

3.2.3 Problems with Flowcharts

Flowcharts were originally developed to help programmers and analysts design and document computer programs. As programs became larger, the use of flowcharts became more difficult to implement. New software has emerged over the last decade to facilitate programming. This includes Visio, a software program exclusively designed for flowcharting, as shown in Fig. 21.33.

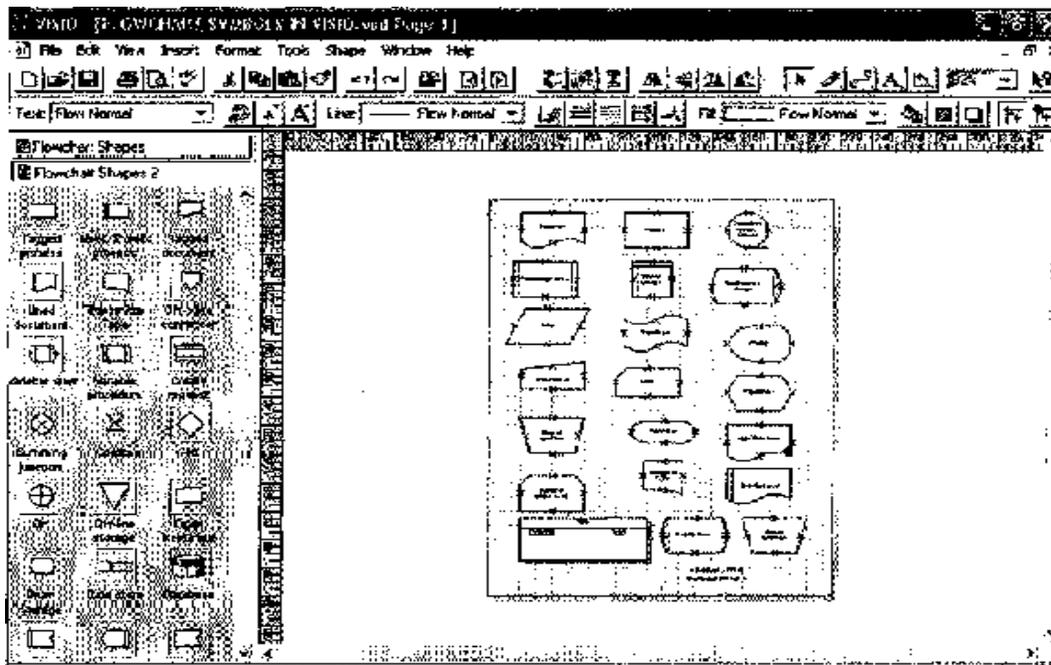


Fig 21.3: Flowcharting software (Visio)

As a result, many organization are now reducing the amount of flowcharting used. In its place, internal documentation, decision tables, grid charts, and structured design methodologies are used. With internal documentation, the purpose and functioning of the program is actually contained within documentation statements in the particular programming language. Depending on the language, these are called remark statements, comment statements, or documentation statements. Decision tables and grid charts are discussed later in this unit.

3.2.4 Decision Tables

A decision table can be used as an alternative or as an addition to flowcharts. The objective of both decision tables and flowcharts is the same. They reveal how the input data is to be manipulated to obtain the desired output. As the name implies, decision tables reveal what decisions or actions the computer is to take as a result of the input data. When the computer has to make a large number of decisions or if there are a large number of different branches within a program, decision tables are particularly useful. In these cases, decision tables are preferred to flowcharts.

In general, a decision table displays the different conditions that could exist and the different actions that the computer should take as a result of these conditions. Most decision tables have six parts. A complete decision table for the airline reservation application is given in Fig. 21.4.

Program Inventory data	Invoice	Inventory reorder	Transport	Sales analysis	Stock adjustment	Master file update
Item name	X	X	X	X	X	X
Item code number	X	X	X	X	X	X
Item price	X			X		X
Item amount		X		X	X	X
Reorder point		X				X
Warehouse location			X			X
Sales-to-date				X		X

Fig. 21.4: Decision table for the airline reservation application

The first part of the decision table contains the name and/or number of the table. For some of the larger applications, two or more decision tables are used in the analysis part of program development. For the computer program developed to make airline reservations, the decision table is called the airline reservation application. This appears at the upper left of the table.

The second part of the table contains the condition statements, which describe the conditions that could exist with the input data. If the computer program is to make tourist and first-class reservations, there may be four possible states or conditions: (1) a first-class reservation is requested, (2) a tourist-class reservation is requested, (3) a first-class seat is available, and (4) a tourist-class seat is available. These conditions are then listed in the second area of the decision table.

The third part of the decision table contains the action statements. While the condition statements reveal the possible states of the input data, the action statements describe the possible actions of the computer system. Given the conditions discussed above, what types of actions might the computer take? Most airline reservation systems will either issue a first-class ticket, wait list a passenger for a first-class ticket, issue a tourist-class ticket, or wait list a passenger for a tourist class ticket.

The right-hand side of the decision table contains the rule numbers (part 4), the actual conditions (part 5), and the actions taken by the computer (part 6). Each column indicates what the computer should do as result of certain conditions. Four columns correspond to the four rule numbers. Under each rule number is a given condition followed by a specific action to be taken by the computer. The four rule numbers, the four actual conditions, and the associated actions taken by the computer system are discussed below.

- Rule 1: A first-class ticket is requested, and a first-class seat is available. The computer should issue a first-class ticket.
- Rule 2 : A first-class ticket is requested, but a first-class seat is not available. The computer should wait list the passenger.
- Rule 3: A tourist-class ticket is requested, and a tourist-class seat is available. The computer should issue a tourist-class ticket.
- Rule 4: A tourist-class ticket is requested, but a tourist-class seat is not available. The computer should wait list the passenger.

In the decision table, “Y” means yes, “N” means no, and “X” means the computer should take this action.

3.2.5 Grid Charts

Application programs are usually complex and interrelated in business. The same data may be used for several programs, and the output from one program may be the input to another. A grid chart is an excellent method to reveal the relationships between application programs. What data is used for several application programs? What data is used to produce different reports and documents? For which applications is the output from one application used as input for another application? These and similar questions can be answered using grid charts. For example, the same inventory data may be used in several application programs. See the example in Fig. 21.5

Name of decision table: 1	Airline reservation application	Rule Number				Rule numbers: 4
		1	2	3	4	
Condition statements						
	First class requested	Y	Y	N	N	
Condition statement: 2	First class available	Y	N	N	N	Actual Conditions: 5
	Tourist class requested	N	N	Y	Y	
	Tourist class available	N	N	Y	N	
Action taken						
	First class ticked issued	X				
Action statement: 3	Tourist class ticked issued			X		Actions taken: 6
	First class wait listed		X			
	Tourist class wait listed				X	

Fig. 21.5: The use of grid charts

The Xs in the grid chart in Fig. 21.5 reveal what data is used for each application program. Look at the first column. The invoice application uses the item name, code number, and price. Likewise, the data used in any of the application programs listed in the grid chart can be found by reading down the appropriate column. In what applications is the reorder point data used? Going across the reorder point row, it can be seen that this data is used in the inventory reorder program and the master file program

4.0 CONCLUSION

Many traditional system design schemas, typical of the waterfall design described in SDLCs in earlier units are still employed to put thoughts to paper. Print charts, screen charts, and layout charts are used to reveal the exact layout and format of input data and output reports and documents. Both flowcharts and decision tables are used to display how the input data is to be manipulated to obtain the desired output. System flowcharts give the overall structure of the application while program flowcharts reveal how each individual program is to be written or constructed. In addition, micro- or detailed flowcharts can be used to reveal in detail how a computation or process is to be performed.

5.0 SUMMARY

In this unit you have learnt the various traditional “waterfall SDL” system design tools, especially flowcharts and decision tables. In general, where there are a number of interrelated decisions or branches, a decision table is preferred over a flowchart. Of course, flowcharts and decision tables can be used together. Finally, this unit concluded with grid charts which relate one or more applications.

6.0 TUTOR MARKED ASSIGNMENT

- What is the purpose of a print chart? What is the purpose of a layout chart? Give examples of both.
- Describe the difference between a system flowchart and a program flowchart. Why are both types of flowcharts useful when developing application software?
- What is a macro flowchart? What is a micro or detailed flowchart?
- What are the basic parts of any decision table?
- Under what circumstances is a decision table preferred over a flowchart? When would a decision table not be appropriate?
- What are some of the uses of a grid chart?

UNIT 2 DEVELOPMENT PHASE

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Tasks And Activity
 - 3.3 System Requirement Analysis
 - 3.4 System Architectural Design
 - 3.5 Software Requirements Analysis
 - 3.6 Software Architectural Design
 - 3.7 Software Details Design
 - 3.8 Software Coding and Testing
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

The objective of the Development Phase will be to convert the deliverables of the Design Phase into a complete information system. Although much of the activities in the Development Phase address the computer programs that make up the system, this phase also puts in place the hardware, software, and communications environment for the system and other important elements of the overall system.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Acquire enough skills to perform software requirements analysis
- List the various stages of system architectural design
- Enumerate the stages of systems requirement analysis

3.0 MAIN CONTENT

3.1 Overview

The activities of this phase translate the system design produced in the Design Phase into a working information system capable of addressing the information system requirements. The development phase contains activities for requirements analysis, design, coding, integration, testing, and installation and acceptance related to software products. At the end of this phase, the system will be ready for the activities of the Integration and Test Phase.

3.2 Tasks and Activities

These activities consist of several tasks that are the responsibility of the developer. The developer shall place the outputs under the Configuration Management Process and perform change control in accordance with it. The developer shall also document and resolve problems and non conformance found in the software products and tasks.

The developer shall select, tailor, and use those standards, methods, tools, and computer programming languages that are documented, appropriate, and established by the organization for performing the activities of the Development Process.

Plans for conducting the activities of the development phase should be developed, documented and executed. The plans should include specific standards, methods, tools, actions, and responsibility associated with the development and qualification of all requirements including safety and security. Separate plans may be developed. The detailed project Work Breakdown Structure (WBS) developed during the Planning Phase should be expanded to incorporate the WBS structure into each module or software configuration item to be developed.

3.3 System Requirements Analysis

- Analyze the intended use of the system to be developed to specify the system requirements. The system requirements specification shall describe the functions and capabilities of the system; the business, organizational and user requirements, the safety, security, human-factors engineering (ergonomics), interface, operations, and maintenance requirements; the design constraints and qualification

requirements. The system requirements specification shall be documented.

- Evaluate the system requirements using the criteria listed below. The results of evaluations shall be documented.
 - traceability to acquisition needs
 - consistency with acquisition needs
 - testability
 - feasibility of system architectural design
 - feasibility of operation and maintenance

3.4 System Architectural Design

- Establish a top-level architecture of the system and document it. The architecture shall identify items of hardware, software, and manual-operations. All the system requirements should be allocated among the hardware configuration items, software configuration items, and manual operations.
- Evaluate the system architecture and the requirements for the above items using the criteria listed below. Document the results.
 - traceability to the system requirements
 - consistency with the system requirements
 - appropriateness of design standards and methods used
 - feasibility of the software items fulfilling their allocated requirements
 - feasibility of operation and maintenance

3.5 Software Requirements Analysis

Establish and document software requirements, including the quality characteristics specifications described below.

- functional and capability specifications, including performance, physical characteristics, and environmental conditions under which the software item is to perform;
- interfaces external to the software item;
- qualification requirements

- safety specifications, including those related to methods of operation and maintenance, environmental influences, and personnel injury;
- security specifications, including those related to compromise of sensitive information;
- human-factors engineering (ergonomics), including those related to manual operations, human-equipment interactions, constraints on personnel, and areas needed concentrated human attention, that are sensitive to human errors and training;
- data definition and database requirements;
- installation and acceptance requirements of the delivered software product at the operation and maintenance site(s);
- user documentation
- user operation and execution requirements
- user maintenance requirements
- Evaluate the software requirements using the criteria listed below and document them.
 - traceability to system requirements and system design
 - external consistence with system requirements
 - internal consistency
 - testability
 - feasibility of software design
 - feasibility of operation and maintenance
- Conduct joint reviews. Joint reviews are at both project management and technical levels and are held throughout the life of the contract. This process may be employed by any two parties, where one party (reviewing party) reviews another party (reviewed party).

3.6 Software Architectural Design

- Transforms the requirements for the software item into an architecture that describes its top-level structure and identifies the software components.

- Ensures that all the requirements for the software item are allocated to its software components and further refined to facilitate detailed design.
- Develop and document a top-level design for the interfaces external to the software item and between the software components of the software item.
- Develop and document a top-level design for the database.
- Develop and document preliminary versions of user documentation.
- Define and document preliminary test requirements and the schedule for Software Integration.
- Evaluate the architecture of the software item and the interface and database designs using the criteria listed below.
 - traceability to the requirements of the software item
 - external consistency with the requirements of the software item
 - internal consistency between the software components;
 - appropriateness of design methods and standards used
 - feasibility of detailed design
 - feasibility of operation and maintenance
 - Conduct joint reviews. Joint reviews are at both project management and technical levels and are held throughout the life of the contract. This process may be employed by any two parties, where one party (reviewing party) reviews another party (reviewed party).

3.7 Software Detailed Design.

- Develop a detailed design for each software component of the software item. The software components shall be refined into lower levels containing software units that can be coded, compiled, and tested. Ensure that all the software requirements are allocated from the software components to software units.
- Develop and document a detailed design for the interfaces external to the software item, between the software components, and between the software units. The detailed design of the interfaces shall permit coding without the need for further information.

- Develop and document a detailed design for the database.
- Update user documentation as necessary.
- Define and document test requirements and schedule for testing software units. The test requirements should include stressing the software unit at the limits of its requirements.
- Update the test requirements and the schedule for Software Integration.
- Evaluate the software detailed design and test requirements considering the criteria listed below:
 - traceability to the requirements of the software item
 - external consistence with architectural design
 - internal consistency between software components and software units
 - appropriateness of design methods and standards used
 - feasibility of testing
 - feasibility of operation and maintenance
 - Conduct joint reviews. Joint reviews are at both project management and technical levels and are held throughout the life of the contract. This process may be employed by any two parties, where one party (reviewing party) reviews another party (reviewed party).

3.8 Software Coding and Testing.

- Develop and document each software unit and database as well as test procedures and data for testing each software unit and database.
- Test each software unit and database ensuring that it satisfies its requirements. Document the results.
- Update the user documentation as necessary.
- Update the test requirements and the schedule for Software Integration.
- Evaluate software code and test results considering the criteria listed below:

- traceability to the requirements and design of the software item
- external consistency with the requirements and design of the software item
- internal consistency between unit requirements
- test coverage of units
- appropriateness of coding methods and standards used
- feasibility of software integration and testing
- feasibility of operation and maintenance

4.0 CONCLUSION

The systems development phase is made of up a series of stages. These must be accomplished sequentially, if a successful development of an information system is to be conducted for any organization.

5.0 SUMMARY

This unit has shown you the various stages of the systems development phase and the specific steps to successfully accomplish each task assignment at each stage of systems development.

6.0 TUTOR MARKED ASSIGNMENT

- List the various stages of systems development phase
- Pick at least 3 of these stages and apply them to an imaginary organization
- List the contact persons in each stage that would provide you with the necessary data in the organization

UNIT 3 IMPLEMENTATION PHASE

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Notification
 - 3.3 Executive
 - 3.4 Data Entry or Conversion
 - 3.5 Install System
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

In this phase, the system or system modifications are installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. Activities in this phase include notification of implementation to end users, execution of the previously defined training plan, data entry or conversion, completion of security certification and accreditation and post implementation evaluation. This phase continues until the system is operating in production in accordance with the defined user requirements.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- List the details of an implementation notice
- Explain system conversion strategies

3.0 MAIN CONTENT

3.1 Overview

The new system can fall into three categories, replacement of a manual process, replacement of a legacy system, or upgrade to an existing system. Regardless of the type of system, all aspects of the implementation phase

should be followed. This will ensure the smoothest possible transition to the organization's desired goal.

3.2 Notification of implementation

The implementation notice should be sent to all users and organizations affected by the implementation. Additionally, it is good policy to make internal organizations not directly affected by the implementation aware of the schedule so that allowances can be made for a disruption in the normal activities of that section. Some notification methods are email, internal memo to heads of departments, and voice tree messages. The notice should include:

- the schedule of the implementation;
- a brief synopsis of the benefits of the new system;
- the difference between the old and new system;
- responsibilities of end user affected by the implementation during this phase; and
- the process to obtain system support, including contact names and phone numbers.

3.3 Execution of Training Plan

It is always a good business practice to provide training before the end user uses the new system. Because there has been a previously designed training plan established, complete with the system user manual, the execution of the plan should be relatively simple. Typically, what prevents a plan from being implemented is lack of funding. Good budgeting should prevent this from happening.

3.4 Data Entry or Conversion

With the implementation of any system, typically, there is old data, which is to be included in the new system. This data can be in a manual or an automated form. Regardless of the format of the data, the tasks in this section are two fold, data input and data verification. When replacing a manual system, hard copy data will need to be entered into the automated system. Some sort of verification that the data is being entered correctly should be conducted throughout this process. This is also the case in data transfer, where data fields in the old system may have been entered inconsistently and therefore affect the integrity of the new database. Verification of the old data becomes imperative to a useful computer system.

One of the ways verification of both system operation and data integrity can be accomplished is through parallel operations. Parallel operations consist of running the old process or system and the new system simultaneously until the new system is certified. In this way if the new system fails in any way, the operation can proceed on the old system while the bugs are worked out.

3.5 Install System

To ensure that the system is fully operational, install the system in a production environment. After the system has been fielded, a post-implementation evaluation is conducted to determine the success of the project through its implementation phase. The purpose of this evaluation is to document implementation experience to recommend system enhancements and provide guidance for future projects.

In addition, change implementation notices will be utilized to document user requests for fixes to problems that may have been recognized during this phase. It is important to document any user request for a change to a system to limit misunderstandings between the end user and the system programmers.

4.0 CONCLUSION

Once the system has been developed, it must be installed and the necessary conversions and trainings undertaken. To the systems analyst, this will mean a virtual life-long relationship with the system environment because he will be almost always on the call to ensure the sustainability of the system

5.0 SUMMARY

In this unit you have learnt details of the system conversion strategies and the determination of the implementation notice

6.0 TUTOR MARKED ASSIGNMENT

- What should implementation notice entail?
- Why is conversion crucial to system implementation?
- What follows after the system has been installed?

UNIT 4 OPERATIONS AND MAINTENANCE PHASE

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 System Operation
 - 3.3 Data/Software Administration
 - 3.4 Problem and Modification Process
 - 3.5 System Software maintenance
 - 3.6 Roles and Responsibilities
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

The emphasis of this phase will be to ensure that the users needs are met and the system continues to perform as specified in the operational environment. Additionally, as operations and maintenance personnel monitor the current system, they may become aware of better ways to improve the system and therefore make recommendations. Changes will be required to fix problems, possibly add features and make improvements to the system. This phase will continue as long as the system is in use.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Enumerate data administration tasks and activities
- Identify systems operations key tasks and activities
- Design a User Satisfaction Review
- Create a Daily Maintenance Activities for the system
- List the roles and responsibilities of maintenance personnel

3.0 MAIN CONTENT

3.1 Overview

More than half of the life cycle costs are attributed to the operations and maintenance of systems. In this phase, it is essential that all facets of operations and maintenance are performed. The system is being used and scrutinized to ensure that it meets the needs initially stated in the planning phase. Problems are detected and new needs arise. This may require modification to existing code, new code to be developed and/or hardware configuration changes. Providing user support is an ongoing activity. New users will require training and others will require training as well.

3.2 Systems Operations

Operations support is an integral part of the day to day operations of a system. In small systems, all or part of each task may be done by the same person. But in large systems, each function may be done by separate individuals or even separate areas. The Operations Manual is developed in previous SDLC phases. This document defines tasks, activities and responsible parties and will need to be updated as changes occur. Systems operations activities and tasks need to be scheduled, on a recurring basis, to ensure that the production environment is fully functional and is performing as specified. The following are checklist of systems operations key tasks and activities:

- Ensure that systems and networks are running and available during the defined hours of Operations;’
- Implement non-emergency requests during scheduled Outages, as prescribed in the Operations Manual;
- Ensure all processes, manual and automated, are documented in the operating procedures. These processes should comply with the system documentation;
- Acquisition and storage of supplies (i.e. paper, toner, tapes, removable disk);
- Perform backups (day-to-day protection, contingency);

- Perform the physical security functions including ensuring adequate UPS, Personnel have proper security clearances and proper access privileges etc.;
- Ensure contingency planning for disaster recovery is current and tested ;
- Ensure users are trained on current processes and new processes;
- Ensure that service level objectives are kept accurate and are monitored;
- Maintain performance measurements, statistics, and system logs. Examples of performance measures include volume and frequency of data to be processed in each mode, order and type of operations;
- Monitor the performance statistics, report the results and escalate problems when they occur.

3.3 Data/Software Administration

Data/Software Administration is needed to ensure that input data and output data and data bases are correct and continually checked for accuracy and completeness. This includes insuring that any regularly scheduled jobs are submitted and completed correctly. Software and data bases should be maintained at (or near) the current maintenance level. The backup and recovery processes for data bases are normally different than the day-to-day DASD volume backups. The backup and recovery process of the data bases should be done as a Data/Software Administration task. A checklist of Data / Software Administration tasks and activities are:

- Performing a periodic Verification / Validation of data, correct data related problems;
- Performing production control and quality control functions (Job submission, checking and corrections);
- Interfacing with other functional areas for Day-to-day checking / corrections;
- Installing, configuring, upgrading and maintaining data base(s). This includes updating processes, data flows, and objects (usually shown in diagrams);

- Developing and performing data / data base backup and recovery routines for data integrity and recoverability. Ensure documented properly in the Operations Manual;
- Developing and maintaining a performance and tuning plan for online process and data bases;
- Performing configuration/design audits to ensure software, system, parameter configuration are correct.

3.4 Problem and Modification Process

One fact of life with any system is that change is inevitable. Users need an avenue to suggest change and identified problems. A **User Satisfaction Review** which can include a Customer Satisfaction Survey, can be designed and distributed to obtain feedback on operational systems to help determine if the systems are accurate and reliable. Systems administrators and operators need to be able to make recommendations for upgrade of hardware, architecture and streamlining processes. For small in-house systems, modification requests can be handled by an in-house process. For large integrated systems, modification requests may be addressed in the Requirements document and may take the form of a change package or a formal **Change Implementation Notice** and may require justification and cost benefits analysis for approval by a review board. The Requirements document for the project may call for a modification cut-off and rollout of the system as a first version and all subsequent changes addressed as a new or enhanced version of the system. A request for modifications to a system may also generate a new project and require a new project initiation plan.

3.5 System/Software Maintenance

Daily operations of the system /software may necessitate that maintenance personnel identify potential modifications needed to ensure that the system continues to operate as intended and produces quality data. Daily maintenance activities for the system takes place to ensure that any previously undetected errors are fixed. Maintenance personnel may determine that modifications to the system and databases are needed to resolve errors or performance problems. Also modifications may be needed to provide new capabilities or to take advantage of hardware upgrades or new releases of system software and application software used to operate the system. New capabilities may take the form of routine maintenance or may constitute enhancements to the system or database as a response to

user requests for new/improved capabilities. New capabilities needs may begin a new problem modification process described above.

At this phase of the SDLC, all security activities have been at least initiated or completed. An update must be made to the Sensitive System Security plan; an update and test of the contingency plan should be completed. Continuous vigilance should be given to virus and intruder detection. The Project Manager must be sure that security operating procedures are kept updated accordingly.

3.6 Review Previous Documentation

Review and update documentation from the previous phases. In particular, the Operations Manual, SBD and Contingency Plan need to be updated and finalized during the Operations and Maintenance Phase.

3.7 Roles And Responsibilities

This list briefly outlines some of the roles and responsibilities for key maintenance personnel. Some roles may be combined or eliminated depending upon the size of the system to be maintained. Each system will dictate the necessity for the roles listed below.

- **Systems Manager.** The Systems Manager develops documents and executes plans and procedures for conducting activities and tasks of the Maintenance Process. To provide for an avenue of problem reporting and customer satisfaction, the Systems Manager should create and discuss communications instructions with the systems customers. Systems Managers should keep the Help Desk Personnel informed of all changes to the system especially those requiring new instructions to users.
- **Technical Support.** Personnel with proven technical support to the program. This support may involve granting access rights to the program, setup of workstations or terminals to access the system, maintaining the operating system for both server and workstation. Technical support personnel may be involved with issuing user ids or login names and passwords. In a Client server environment technical support may perform systems scheduled backups and operating system maintenance during downtime.

- **Vendor Support.** The technical support and maintenance on some programs are provided through vendor support. A contract is established outlining the contracted systems administration, operators, and maintenance personnel duties and responsibilities. One responsibility which should be included in the contract is that all changes to the system will be thoroughly documented.
- **Help Desk .** Help Desk personnel provide the day to day users help for the system. Help desk personnel should be kept informed of all changes or modifications to the system. Help Desk Personnel are contacted by the user when questions or problems occur with the daily operations of the system. Help Desk Personnel need to maintain a level of proficiency with the system.
- **Operations or Operators (turn on/off systems, start tasks, backup etc).** For many mainframe systems, technical support for a program is provided by an operator. The operator performs scheduled backup, performs maintenance during downtime and is responsible to ensure the system is online and available for users. Operators may be involved with issuing user ids or login names and passwords for the system.
- **Customers.** The customer needs to be able to share with the systems manager, the need for improvements or the existence of problems. Some users live with a situation or problem because they feel they must. Customers may feel that change will be slow or disruptive. Some feel the need to create work-around. A customer has the responsibility to report problems or make recommendations for changes to a system.
- **Program Analysts or Programmer.** Interprets user requirements, designs and writes the code for specialized programs. User changes, improvements, enhancements may be discussed in Joint Application Design sessions, analyze programs for errors, debugs the program and tests program design.
- **Process Improvement Review Board.** A board of individuals may be convened to approve recommendations for changes and improvements to the system. This group may be chartered. The charter should outline what should be brought before the group for consideration and approval. The board may issue a Change Directive.

- **Users Group or Team.** A group of computer users who share knowledge they have gained concerning a program or system. They usually meet to exchange information, share programs and can provide expert knowledge for a system under consideration for change.
- **Contracting Officer Technical Representative (COTR).** The COTR has many responsibilities when a contract has been awarded for maintenance of a program. The COTR should have a certificate of training for completion of a COTR course. The COTR'S main role is to make sure that the interests of the Contracting Office are protected and that no modifications are made to the contract without permission from the Contracting office.
- **Data Administrator.** Performs ensures that accurate and valid data are entered into the system. Sometimes this person creates the information systems database, maintains the databases security and develops plans for disaster recovery. The data administrator may be called upon to create queries and reports for a variety of user requests. The data administrator responsibilities include maintaining the databases data dictionary. The data dictionary provides a description of each field in the database, the field characteristics and what data is maintained with the field.
- **Telecommunications Analyst and Network System Analyst.** Plans, installs, configure upgrades and maintain networks as needed. If the system requires it, they ensure that external communications and connectivity are available.
- **Computer Systems Security Officer (CSSO).** The CSSO has a requirement to review system change requests, review and in some cases coordinate the Change Impact Assessments, participate in the Configuration Control Board process, conduct and report changes that may be made that affect the security posture of the system.

4.0 CONCLUSION

The process of maintaining a fully implemented system requires significant input from various sources within the organization. More than any other portion of the system analysis and design, implementation requires an effective communication link between the various systems analysts within the system to ensure success.

5.0 SUMMARY

This unit has explained the various stages and operations of the maintenance phase of the system design process. The emphasis of this phase was to ensure that the users needs are met and the system continues to perform as specified in the operational environment. Additionally, as operations and maintenance personnel monitor the current system, they may become aware of better ways to improve the system and therefore make recommendations. Changes will be required to fix problems, possibly add features and make improvements to the system.

6.0 TUTOR MARKED ASSIGNMENT

- Explain the rationale of daily maintenance activities for the system
- What are the individual items in a checklist of system operations key tasks and activities?
- What is data and software administration?
- What is the rationale of the daily maintenance activities for the system?

UNIT 5 THE DESIGN PROCESS

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 The Process
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

In unit 21 we discussed system design tools. These were tools that enable the system analyst to, as it were, put down on paper, his ideas and help him create a map of the general information system he is creating. In this final module of the course, we are going to now look at the various additional programming tools that the analyst must employ to develop the Information System for an organization, which translated the system designs into user-friendly applications.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the waterfall SDLC development stages
- Explain the steps in application development
- Explain the advantages of the steps in application development

3.0 MAIN CONTENT

3.1 Overview

It was to identify sub-stages within each stage of the system development model of SDLC (go back to Unit 12 to refresh your studies on System Development Life Cycle, or SDLC). However, there is no general agreement on what sub-stages being followed in order to complete the development. Depending on the workload distribution among the

development team members, different approaches were established knowingly without misleading the flow as discussed herein. Even with the mentioned well formulated design, the development process was under influence of the difficulties unforeseen circumstances with and from the resources and program code, the task been changed in several modes but all the practices remained within.

3.2 The Process

Figure 29.1 recapitulates the “waterfall” SDLC key steps:

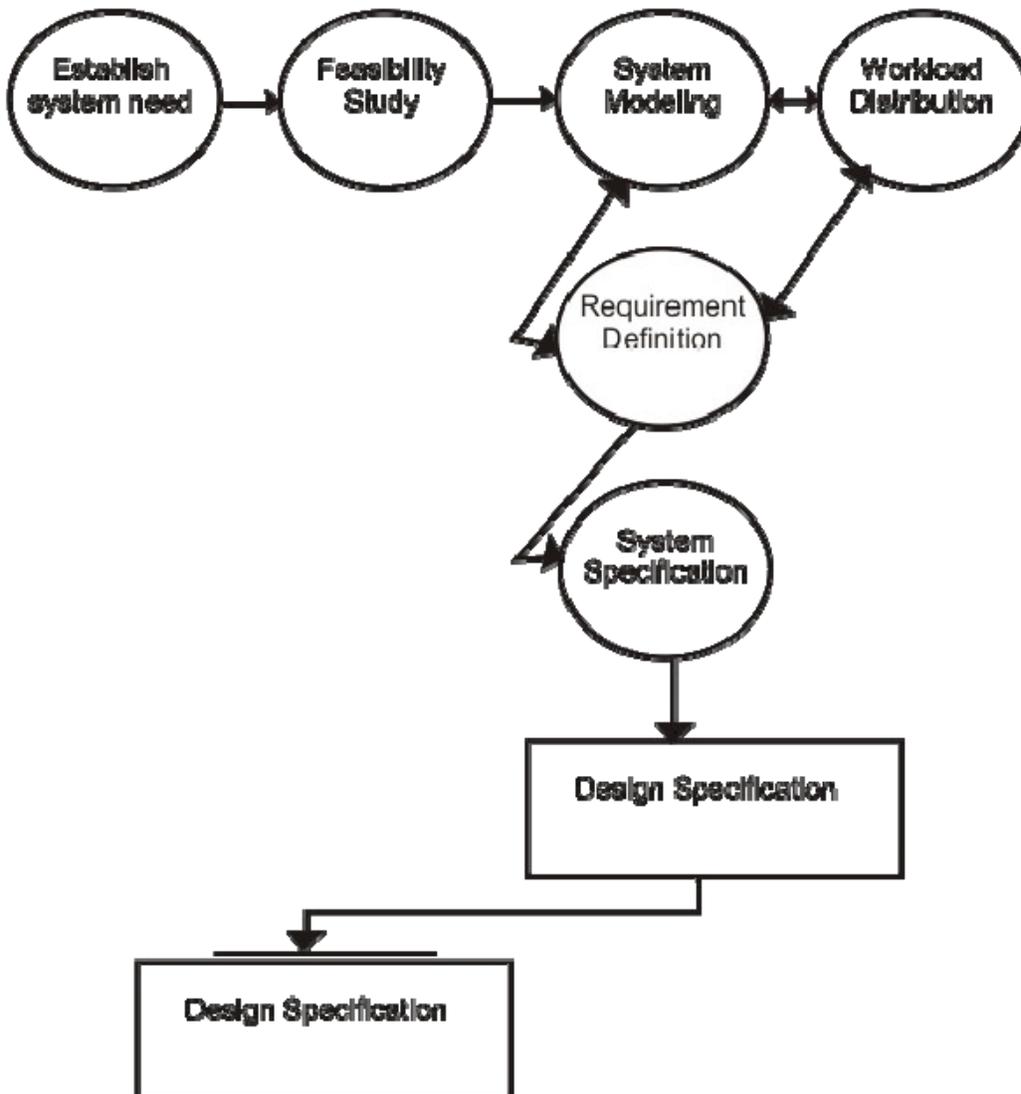


Fig 29.1. The process formulation

The process is a structured set of activities required to develop a system. A common process framework is established by defining a small number of framework activities that are applicable to all projects. Task sets are created under a common process framework and they contain a group of work tasks, project milestones, work products, deliverables and quality assurance points. These are important in implementing the project efficiently.

The procedures outlined are the traditional steps to developing application software. Today many organizations are using a newer approach to application development, called structured design. Developing useful computer programs and application software requires a substantial amount of detailed planning. Through years of experience, a series of steps and planned activities have been developed to maximize the likelihood of developing profitable and useful software. Many companies and computer manufacturers have developed books, seminars, cassettes, and other materials on the application development process with titles such as “Managing Application Development” and “Computer Problem Solving.” While the particular procedures have different names, the overall approach is basically the same. A brief description of the steps in application development and the people involved is shown in Table 1.

Step	People involved	Brief descriptions
1. Problem definition output and requirements.	User, analyst	Define problem and specify input
2. Analysis and design new	User, analyst	Analyze old system and design system.
3. Language selection	Analyst, Programmer	Determine best programming language.
4. Programming	Programmer	Develop the necessary programs.
5. Testing and debugging	Analyst, programmer	Make sure program

performs as
designed.

6.	Documentation	Analyst, programmer	Develop technical and user documentation.
7.	Implementation	User, analyst	Phase in new programs while phasing out old procedures.
8.	Maintenance	User, analyst, programmer	Enhance and improve programs.

The fundamental steps outlined in Table 1 offer a number of advantages over an approach without a formal procedure. These advantages include:

1. Better management control over the project.
2. The use of proven techniques and procedures.
3. Good use of data processing personnel.
4. Better estimates of project completion time.
5. Better estimates of total project costs,
6. Improved quality control over entire project.
7. Better program design, which leads to fewer problems with debugging and maintenance.
8. Greater likelihood of satisfying user needs and desires.

The importance of program design (problem definition, analysis, and design) cannot be overemphasized. It is easier, faster, and cheaper to design a good program than to try to fix and maintain a poorly designed one. The key to good design is planning. By spending more time in the design phase, it is normally possible to spend less time on coding, testing, and maintenance. In some cases, better program design can reduce total programming time and cost. In the 1960s, a typical application might take about six months to complete. Today, because applications are becoming longer and more complex, the typical application can take over three years to complete. Thus, time and cost savings from good planning and design is becoming more important.

It should be noted that one step does not have to be completely finished before the next step is started. For example, during analysis and design, materials are developed that are used in documentation. Furthermore, many programmers develop test data, during the initial steps that is used during the testing and debugging phase. In other cases, programmers perform all of

these steps for one part of one program, and then this process is repeated until the entire program is written. Application and program development is a continuous process. This is called the programming life cycle .

4.0 CONCLUSION

Like links in a chain, the eight steps of developing application software provide the connection between the potential of the computer hardware and the desires of the individuals using the computer system. When one step is not adequately completed, there is a weak link, and the computer system will probably not be able to perform to its potential in satisfying the needs and desires of those individuals using the system.

5.0 SUMMARY

In this unit, you have leant the process structure to system design.

6.0 TUTOR MARKED ASSIGNMENT

- Why is a formal structure favored in system design?
- Explain the steps in application development.
- Explain the advantages of the steps in application development.

MODULE 5 INFORMATION SYSTEM DEVELOPMENT TOOLKIT

- Unit 1 Key System Development Life Cycle
 Supporting Processes
- Unit 2 The process model
- Unit 3 Examples of programming language
- Unit 4 Risk Management
- Unit 5 IT development Case Study

UNIT 1 KEY SYSTEM DEVELOPMENT LIFE CYCLE SUPPORTING PROCESSES

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Project Planning
 - 3.2.1 Organizational Development
 - 3.2.2 Organizational Development
 - 3.2.3 Staffing
 - 3.2.4 Personnel Training
 - 3.2.5 Communication Channels
 - 3.2.6 Project Officer Charter
 - 3.2.7 Organizational Processes Development
 - 3.3 Acquisition Strategy Development
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References and Further Readings

1.0 INTRODUCTION

In **Unit 12**, we first encountered the concept of SDLC. In this unit, we are delving further by looking at the supportive processes that make the stages of SDLC effective.

2.0 OBJECTIVES

By the end of this unit, you should be able to

- Outline the processes that shape the SDLC life-cycle
- Create a System Boundary Document (SBD) for an SDLC project
- Acquire the skills needed for a project manager to work on SDLC projects
- Explain the process of Acquisition Strategy Development

3.0 MAIN CONTENT

3.1 Overview

This unit serves as an introduction to several key process areas that should be used throughout the system life cycle to manage the development and operations of information systems. Processes should address:

- Understanding Customer Needs and Expectations
- Deriving and Allocating Requirements
- Planning the Technical effort
- Analyzing Candidate Solutions
- Developing Physical Architectures
- Integrating Acquisition Disciplines
- Managing, Monitoring, and Controlling the Technical Effort
- Integrating the System
- Verifying and Validating the System
- Ensuring Quality
- Managing Configurations

- Managing Risk
- Managing Transition to Operational Use
- Managing System Support
- Defining and Improving Management and Engineering Processes

For this unit, processes are grouped in the general categories of project planning, requirements management, project tracking and oversight, contractor management, verification and validation, quality assurance, change management, and risk management.

3.2 Project Planning

For each information system project, the agency should designate a responsible organization and assign that organization sufficient resources to execute the project. Additionally, the agency may tentatively identify the operating organization if different from the project organization.

The project organization should perform the initial project planning described below and present the following System Boundary Document (SBD) elements to the appropriate level for approval:

1. Project Office Charter and Resource Requirements
2. Acquisition Strategy Paper
3. Acquisition Project Baseline
4. Project Schedule
5. Project Cost Estimate

Once approved, these documents should provide the basis for internal project management as well as agency oversight. These SBD elements should be reviewed and updated, as appropriate, prior to initiating each system development life cycle phase.

3.2.1 Organizational Development

Development of large information systems requires unique management skills and experience as well as underlying knowledge of the technologies involved in the project. The designated project manager for large information system projects should possess, as a minimum, the following knowledge, skills and abilities:

- A degree or higher in engineering, computer science, or technical field related to the project.

- Demonstrated ability to manage development projects, to include accountability for project cost, schedule, and technical performance.
- Knowledge of agency development policies and procedures.
- Demonstrated ability to manage contracted efforts.
- Demonstrated ability to represent the project at all levels of the government as well as to the public.

3.2.2 Organizational Alignment

The project office may be placed at any location or level of the agency as appropriate. In all cases, the designated project manager should not be the warranted contracting officer for any associated contracts.

3.2.3 Staffing

The project office organization should be staffed with the appropriate number and kinds of personnel experienced in the domain of the application being acquired. These individuals may be directly assigned to the project office or matrixed from supporting organizations. Depending on the project size and complexity, project office personnel should include individuals trained and experienced in technical management (engineering, computer science, configuration management, quality assurance, logistics, etc.), business management (finance, cost-estimating, business management, etc.), and contract management. Support contractors may augment project office staffing, provided all applicable public law and Federal regulations are applied to the efforts (e.g., inherent governmental activities, non-personal services, etc.). Additionally, project office personnel should be supplied with all needed supplies, equipment, tools, and services needed to accomplish the project objectives.

Information technology development projects, like all acquisition development projects, require personnel with management and technical skills that are congruent with the acquisition strategy selected for the project. Selecting an acquisition strategy that requires skills and experience not present in the available staffing introduces major risks.

3.2.4 Personnel Training

All assigned and supporting project office personnel should be provided continuous opportunities for training in their technical, business, and contract management functions. Additionally, the project office should provide assigned personnel with training specific to the project requirements, with emphasis on training in the area of advanced technologies being applied to the project.

3.2.5 Communication Channels

Communications channels should be established through the agency organization to the agency IRM Office, as needed, to provide for internal agency tracking of project progress.

Organizational placement should allow sufficient visibility of project needs to allow agency resources to be effectively applied to the project.

The management of the project organization and individual reporting channels should be in accordance with published agency directives. Additionally, for large information system projects, the designated project manager should have a formal, unrestricted communications channel to the agency Information Resource Manager.

3.2.6 Project Office Charter

Effective with the identification of the business need (in the System Concept Development Phase) for a project, the agency should designate the developing project organization. For a new large project, this may require the creation of a new organizational element. For smaller projects and modifications to existing systems, the project may be assigned within a current organizational element. In all cases, the identification of a distinct project organization should also include the identification of a designated project manager who carries both the responsibility and accountability for project execution. The project manager is responsible for creating the SBD. This package, when approved by the project manager, the resource provider, and the decision authority, should constitute a contract between the project manager and the decision authority as detailed in the project office charter. All project reporting, tracking, and oversight should reference this package. The SBD should also contain specific requests for the removal (waivers) of constraints within the purview of the decision authority.

3.2.7 Organizational Processes Development.

To provide a management structure for the project, the project office should adapt, adopt, or create written processes and procedures for recurring project office activities. This guide defines several of the critical processes essential to the successful project execution. These include requirements management, project tracking, contractor management, verification and validation, quality assurance, change management, and risk management. For agencies with mature acquisition organizations, this activity may be confined to following existing general processes or adapting an existing project's processes to the new project.

3.3 Acquisition Strategy Development

Acquisition strategy is a combination of business and technical concepts designed to meet the stated business need within any specified constraints. It is the framework for managing all phases of the life cycle and provides the underlying strategy for all program plans and activities.

While the strategy may evolve over time, it should contain elements from across all life cycle phases, including disposal.

4.0 CONCLUSION

System analysis and development is an involved process, and early identification of the various levels of input into the process ensures an effective program development. Again, this brings to fore, the strategic importance of systems thinking as a more user-friendly than the traditional systems development strategies.

5.0 SUMMARY

In this unit you have learnt the further integrated nature of systems thinking. We are now beginning to approach the code-writing stages of system development, and this unit has taken you through some of the supportive inputs into the process.

6.0 TUTOR MARKED ASSIGNMENT

- What is a System Boundary Document, and what is its role in SDLC project?
- What are the range of skills and knowledge one must require to be a project manager on a SDLC project?

UNIT 2 THE PROCESS MODEL

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Programming language
 - 3.3 Early language
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

The key to making the computer system perform business applications is application software. Whether an organization makes its own or buys it, application software must be developed. As a systems analyst (go to **Module 2, Unit 6** to refresh your memory about system analysts) you will be directly involved in developing application software. Thus you should know how computer programs are written and used. With the increased use of personal computers, you may be writing short programs for your own use. Many managers who know how to program in a high-level language are finding they are able to write programs that help them do their job. Since application software is vital to every organization, you should have an understanding of what a computer program is and how it is written.

2.0 OBJECTIVES

By the end of this unit, you should be able to

- Explain the waterfall model of SDLC development
- Explain the role of programming languages in system design

3.0 MAIN CONTENT

3.1 Overview

Programs can be written in machine language, assembler language, symbolic languages, high-level language, or other types of languages. Over the years, there has been a trend toward writing application software in high-level languages. A programming language should possess the following characteristics to be considered high-level:

1. The language is relatively independent of a given computer system.
2. Each statement in the language translates into several instructions in machine language.
3. The language is natural and uses abbreviations and words used in everyday communication.
4. The language is independent of machine language instructions and other pieces of system software except for the language compiler.
5. The language is not experimental in nature and exists on more than one computer system.

Since there are over 200 high-level programming languages that meet the above qualifications, it would be impossible to investigate all of them in any detail. This should be covered by your other courses on computer programming. The purpose of this module is to discuss the general approaches to using programming languages as tools for developing information systems.

3.2 Programming languages

As we have seen in this course, the development of a large piece of software — the ultimate end-goal of system design — is an engineering task, to be approached with the same care as the construction of a big building, for example, and with the same attention to cost, reliability, and maintainability of the final product. Again as we have seen in earlier units, the software-engineering process is usually described as consisting of several phases, variously defined but in general consisting of: (1) identification and analysis of user requirements, (2) development of system specifications (both hardware and software), (3) software design (perhaps at several successively more detailed levels), (4) implementation (actual coding), (5) testing, and (6) maintenance.

Since the early 1980s, increasingly sophisticated tools have been built to aid the software developer and to automate as much as possible, the

development process. Such computer-aided software engineering (CASE) tools span a wide range of types, from those that carry out the task of routine coding when given an appropriately detailed design in some specification language, to those that incorporate an expert system to enforce design rules and eliminate software defects prior to the coding phase.

As the size and complexity of software grow, the concept of reuse has become increasingly important in software engineering, since it is clear that extensive new software cannot be created cheaply and rapidly without incorporating existing program modules (subroutines, or pieces of computer code). One of the attractive aspects of object-oriented programming is that code written in terms of objects is readily reused. As with other aspects of computer systems, reliability—usually rather vaguely defined as the likelihood of a system to operate correctly over a reasonably long period of time—is a key goal of the finished software product. Sophisticated techniques for testing software have therefore been designed. For example, a large software product might be deliberately “seeded” with artificial faults, or “bugs” if they are all discovered through testing, there is a high probability that most actual faults likely to cause computational errors have been discovered as well. The tools for designing end-user systems are programming languages. Let us look at a little history first.

3.3 Early languages

Programming languages are the languages in which a programmer writes the instructions that the computer will ultimately execute. The earliest programming languages were assembly languages, not far removed from the binary-encoded instructions directly executed by the machine hardware. Users soon (beginning in the mid-1950s) invented more convenient languages.

COBOL, FORTRAN, and their descendants, such as Pascal and C, are known as imperative languages, since they specify as a sequence of explicit commands, how the machine is to go about solving the problem at hand; this is not very different from what takes place at the machine level. Other languages are functional, in the sense that, programming is done by calling (i.e., invoking) functions or procedures, which are sections of code executed within a program. The best-known language of this type is **LISP** (List Processing), in which all computation is expressed as an application of a function to one or more “objects.” Since LISP objects may be other functions as well as individual data items (variables, in mathematical terminology) or data structures, a programmer can create functions at the appropriate level of abstraction to solve the problem at hand. This feature

has made LISP a popular language for artificial intelligence applications, although it has been somewhat superseded by logic programming languages such as **Prolog** (Programming in Logic). These are nonprocedural or declarative languages, in the sense that, the programmer specifies what goals are to be accomplished but not how specific methods are to be applied to attain those goals. Prolog is based on the concepts of resolution (akin to logical deduction) and unification (similar to pattern matching). Programs in such languages are written as a sequence of goals. A recent extension of logic programming is constraint logic programming, in which pattern matching is replaced by the more general operation of constraint satisfaction. Again, programs are a sequence of goals to be attained, in this case the satisfaction of the specified constraints.

4.0 CONCLUSION

This unit has introduced you to the programming language environment, which the systems designer must familiarize himself with if he or she is to design a new information system.

5.0 SUMMARY

The programming environment has dramatically evolved over the years, from the stages when programming is like magic — with arcane formulas and rituals, to a canned programming environment where the new languages do not even require coding on the part of the programmer. This unit is the tip of the larger programming iceberg to follow.

6.0 TUTOR MARKED ASSIGNMENT

- Explain the waterfall model of SDLC development
- Explain the role of programming languages in system design

UNIT 3 EXAMPLES OF PROGRAMMING LANGUAGES

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Assembly language
 - 3.2.1 Fortran
 - 3.2.2 Cobol
 - 3.2.3 PL/I
 - 3.2.4 Basic
 - 3.2.5 APL
 - 3.2.6 Pascal
 - 3.2.7 Ada
 - 3.2.8 C
 - 3.2.9 LISP
 - 3.2.10 Fourth-generation languages (AGLs)
 - 3.2.11 Object-oriented language
 - 3.3 High-level language
 - 3.4 Rapid Application Development Language
 - 3.4.1 Background
 - 3.4.2 The Benefits of RAD
 - 3.4.3 DSDM – a RAD Methodology
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

There are many programming languages for different purposes. Some of the more important ones are discussed here.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- List the major programming languages
- Explain the merits of each programming language

3.0 MAIN CONTENT

3.1 Overview

The programming languages are the main tools of the new age system designer. While the traditional tools of flowchart designs provide the “old school” programmer-analyst with the tools to visualize an information system, the newly accessible programming languages such as Microsoft’s Visual Basic provide a more dynamic way of developing system applications.

3.2 Assembly languages

Although computers from one manufacturer tend to have the same machine language, those from different manufacturers do not. Accordingly, different computers have different assemblers and assembly languages. In addition to the conversion of the mnemonic operation code and decimal operand in each instruction into machine code, most assembly languages have functions to facilitate programming, such as combining a sequence of several instructions into one pseudo-instruction. Machine code instructions are then generated from this pseudo-instruction. Programming in assembly languages requires a solid knowledge of computer architecture and is more time-consuming than programming in high-level languages.

3.2.1 FORTRAN

FORTRAN (Formula Translation) was developed as a language for numerical analysis computation by John W. Backus and others at IBM and was announced by IBM in 1957. It has been revised several times since then. Even though other languages, such as C, are becoming popular for scientific and engineering computations, FORTRAN is still the language of choice for numerical analysis. In order to extend its applicability to scientific computations beyond numerical analysis, facilities for handling structured data, dynamic data allocation, recursive calls, and other features were added in the version released in 1990, FORTRAN 90.

3.2.2 COBOL

COBOL (Common Business-Oriented Language) is the most popular language in the business community, including banks and insurance companies. Computer users and manufacturers joined with the U.S.

Department of Defense to set up a common programming language for business applications and formed the Conference on Data Systems Languages (CODASYL) in 1959. CODASYL created COBOL to fulfill two major objectives: portability (the ability of programs to be run with minimum modification on computers manufactured by different companies) and readability (the ease with which programs can be read like ordinary English sentences). COBOL has been revised several times since 1959. It can be more easily understood by business people than can other languages, and programs written in COBOL are quite portable.

3.3.3 PL/I

PL/I is a complex language that was suggested by SHARE (a group of IBM computer users) and IBM in 1963. It was initially called NPL (New Programming Language) but later renamed PL/I. IBM announced the first manual of PL/I in 1965. The American National Standards Institute (ANSI) and other organizations have revised it several times since then. PL/I was designed for both scientific/engineering and business problems by integrating the features of FORTRAN, COBOL, and ALGOL, which were popular languages at that time. Many different types of data can be processed, and a large range of arithmetic and other operations are available. In recent years PL/I has been used less frequently.

3.3.4 BASIC

BASIC (Beginner's All-Purpose Symbolic Instruction Code) is a general-purpose programming language developed by John G Kemeny and Thomas E. Kurtz at Dartmouth College, Hanover, N.H., in the mid-1960s. It is one of the simplest high-level languages and can be learned with relative ease, even by schoolchildren and novice programmers. Since about 1980, BASIC has become popular for use on personal computers.

3.3.5 APL

APL (A Programming Language) is based on the book, A Programming Language published in 1962 by Kenneth E. Iverson of IBM. In 1968 IBM announced APL/360 as its first version. APL was initially used for scientific and engineering problems, but, since IBM introduced APLSV (APL Shared Variable) in 1973, APL can handle general files and consequently has been adopted for business programs as well. Statements are expressed with simple notations, and their operational functions are powerful.

3.3.6 Pascal

Pascal is a language developed by Niklaus Wirth of the Federal Institute of Technology, Zurich, Switzerland. in the late 1960s. It was intended to be a good educational tool for the systematic teaching of programming and to have fast, reliable compilers. Since 1974 the Pascal compiler developed by Wirth has been available to the public and has been used by many universities. Pascal strongly influenced many languages developed later, such as Ada. The language specifications of Pascal are concise, making it easier to learn than many other high-level languages. Complex data structures and algorithms can be described concisely by Pascal, and its programs are easy to read and to debug.

3.3.7 Ada

Ada is a high-level language whose development was initiated in 1975 by the U.S.Department of Defense. Ada was intended to be a common language that could be used on the department's computers, which were produced by many different manufacturers. The first version of the language specification was completed in 1979. It was formally named Ada in honor of Augusta Ada King, Countess of Lovelace and daughter of Lord Byron. (She had worked as an assistant to Charles Babbage in the development of his Analytical Engine and is often credited as being the world's first computer programmer.) Ada is similar to Pascal but contains many additional features that are convenient for the development of large-scale programs. Because of its abundant features, however, ordinary users may feel awkward using Ada. Thus, Ada has not been widely used in programs other than those for the Department of Defense.

3.2.8 C

Although C is considered to be a high-level language, it has many low-level features, such as the ability to directly handle addresses and bits. C is, nonetheless, highly portable. It was developed by Dennis M. Ritchie of AT&T Bell Laboratories in 1972. The operating system UNIX has been written almost exclusively in C; previously, operating systems were almost entirely written in assembly or machine code. C has been extensively used on personal and larger computers.

3.3.9 LISP

LISP (List Processor) is a language that is powerful in manipulating lists of data or symbols rather than processing numerical data. In this sense, LISP is unique. It requires large memory space and, since it is usually processed by an interpreter, is slow in executing programs. LISP was developed in the late 1950s and early 1960s by a group headed by John McCarthy, then a professor at the Massachusetts Institute of Technology. At that time, LISP was radically different from other languages, such as FORTRAN and ALGOL. Several versions have been developed from the LISP 1.5 introduced by McCarthy; Common LISP, released in 1984, is becoming the de facto standard of LISP.

3.3.10 Fourth-generation languages (4GLs)

These are closer to human language than other high-level languages. 4GLs are intended to be easier for users than machine languages (the first generation), assembly languages (the second generation), and high-level languages (the third generation). Many 4GLs actually incorporate third-generation software as well. 4GLs are used primarily for database management. A typical 4 GL statement is FIND ALL RECORDS WHERE NAME IS "SHEHU". FOCUS, developed by Information Builders, Inc., is a database management system that includes a 4GL. It is popular among COBOL users because FOCUS is similar to COBOL and can be easily used with COBOL programs. Structured Query Language (SQL) and Query By Example (QBE), both produced by IBM, are other examples of 4GLs.

3.3.11 Object-oriented languages

An important trend in programming languages is support for data encapsulation, or object-oriented code. Data encapsulation is best illustrated by the language Smalltalk, in which all programming is done in terms of so-called objects. An object in Smalltalk or similar object-oriented languages consists of data together with the procedures (program segments) to operate on that data. Encapsulation refers to the fact that an object's data can be accessed only through the methods (procedures) provided. Programming is done by creating objects that send messages to one another so that tasks can be accomplished cooperatively by invoking each others' methods. This object-oriented paradigm has been very influential. For example, the language C, which was popular for engineering applications and systems development, has largely been supplanted by its object-oriented extension C++.

An object-oriented version of BASIC, named Visual BASIC, is available for personal computers and allows even novice programmers to create interactive applications with elegant graphical user interfaces (GUIs). One of the most powerful examples of this development tool is Microsoft Visual Basic.

In 1995 Sun Microsystems, Inc., introduced Java, yet another object-oriented language. Applications written in Java are not translated into a particular machine language but into an intermediate language called Java Bytecode, which may be executed on any computer (such as those using UNIX, Macintosh, or Windows operating systems) with a Java interpretation program known as a Java virtual machine. Thus Java is ideal for creating distributed applications or Web-based applications. The applications can reside on a server in Bytecode form, which is readily downloaded to and executed on any Java virtual machine. In many cases it is not desirable to download an entire application but only an interface through which a client may communicate interactively with the application. Java applets (small chunks of application code) solve this problem. Residing on Web-based servers, they may be downloaded to and run in any standard Web browser to provide, for example, a client interface to a game or database residing on a server.

3.4 High-level languages

At a still higher level of abstraction lie visual programming languages, in which programmers graphically express what they want done by means of icons to represent data objects or processes and arrows to represent data flow or sequencing of operations. As of yet, none of these visual programming languages has found wide commercial acceptance. On the other hand, high-level user-interface languages for special-purpose software have been much more successful; for example, languages like Mathematica, in which sophisticated mathematics may be easily expressed, or the “fourth generation” database-querying languages that allow users to express requests for data with simple English-like commands. For example, a query such as “Select salary from payroll where employee = ‘Shehu’”, “written in the database language SQL (Structured Query Language), is easily understood by the reader. The high-level language HTML (Hypertext Markup Language) allows nonprogrammers to design Web pages by specifying their structure and content but leaves the detailed presentation and extraction of information to the client’s Web browser.

3.5 Rapid Application Development Languages

Rapid Application Development or RAD is a new (highly interactive) systems development approach whereby qualitatively better systems can be realized in a shorter working less time and at lower cost compared to the traditional one. System development projects are famous for not meeting their delivery dates. Even worse, the system may prove insufficient to meet actual business needs once it is delivered. Strangely enough though, businesses are often inclined to accept non-perfect solutions if they are delivered on time. The Rapid Application Development (RAD) approach was developed invented to help cope with these problems.

3.5.1 Background

The techniques which collectively are known as RAD were first formalized in the methodology called RIPP (Rapid Iterative Production Prototyping) at Dupont in mid-1980s. James Martin extended the work done at Dupont and elsewhere into a larger more formalized process, later called Rapid Application Development in which he coined the term RAD.

Between 1970 and 1990, most systems development took place in an environment which— though complex—was far simpler technically than the world offered to us by modem technology architectures. The business problems solved by those systems were also relatively simple, and the personnel who worked on the design and implementation of the systems were almost entirely technical personnel.

The world is a different place now. When you add up the technical complexity of client-server systems which cross multiple operating system platforms, the business complexity of the issues those systems address, and the much wider organizational involvement imposed by user-driven requirements definition processes, you end up with a situation that's far too tangled for technicians to unravel unaided. Systems development in this world demands the cooperation of people from many different parts of an organization. Such fluid movement of people through a project is common practice in most consulting organization, but it's far from the norm in many maintenance-oriented shops.

3.5.2 The Benefits of RAD

- **Shorter development.** Using RAD, the operational version of an application is available much earlier than with a conventional

development. Average delivery time-scales are in the range of three - six months.

- **Business focus.** The key emphasis in RAD is fulfilling business need. This is frequently expressed in terms of the 80/20 rule. The assumption is that 80% of an application can be delivered in 20% of the time. The remaining 20% of an application frequently amounts to 'copper-plating' - putting a technical finish on the software product. In RAD, technological or engineering excellence is important, but is seen as coming second after business excellence.
- **Lower cost.** Because RAD produces systems more quickly and to a business focus, this approach tends to produce systems to a lower cost. This is particularly important at a time where business is looking to the IS function to better justify its investment in information technology.
- **Stakeholder commitment.** It is claimed that RAD projects engender a greater level of commitment from stakeholders, both business and technical, than conventional development. Users are seen as gaining more of a sense of 'ownership' of a system. Developers are seen as gaining more satisfaction from producing successful systems quickly.

3.5.3 DSDM - a RAD methodology

In 1992, a number of experienced RAD developers came together and decided to develop a new methodology, to combine the best elements of existing methods and practical experience.

This methodology became the Dynamic System Development Method (DSDM). A consortium of several different companies was set up to maintain it, with DCE as a member. In 1995 DCE introduced the methodology in the Benelux and founded the DSDM consortium there with six other companies.

4.0 CONCLUSION

This unit has taught you about the major programming languages, including the newly accessible Rapid Application Development languages, as tools for system design.

5.0 SUMMARY

Thus with such a large array of programming languages at their disposal, system analysts are spoiled for choice in the tools they can use to develop their new systems. The final choice will be dictated by expertise, time, resources and the nature of the information system being developed.

6.0 TUTOR MARKED ASSIGNMENT

- What is the purpose of a high-level programming language?
- Briefly describe the development and use of the BASIC programming language? “How are BASIC programs normally executed? Describe how you would run a BASIC program on a computer system.
- What was FORTRAN originally developed to do?
- Briefly describe the development of the COBOL language. What is (ANS) COBOL?
- What are the four major divisions of any COBOL program? What is a section, a paragraph, a sentence, and a clause?
- Briefly describe the purpose of the four divisions of a COBOL program.
- Contrast Pascal, RPG. PU1, and APL. Briefly describe the development and use of these programming languages.

UNIT 4 RISK MANAGEMENT

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Risk Identification
 - 3.3 Risk Analysis
 - 3.4 Risk Planning
 - 3.5 Risk Tracking
 - 3.6 Risk Control
 - 3.6.1 Risk Mitigation Techniques
 - 3.6.2 Risk Communication
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

Risk Management (RM) is the process of assessing risk, taking steps to reduce risk to an acceptable level and maintaining that level of risk. It also refers to the process of accepting, transferring, or mitigating risk.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Define the concept of risk in system development.
- Explain the mechanism of risk management.
- Categorize risk sources to information system projects.
- Explain the process of limiting risk to system projects.

3.0 MAIN CONTENT

3.1 Overview

Risk Management activities include documenting and identifying project risks; analysis, assessment, and prioritization of those project risks; and laying out plans to implement actions to reduce the project risks throughout the project's life cycle. Risk Management planning provides a control mechanism to monitor, report, and direct all risk mitigation activities. Risk management is initiated during the System Concept Development Phase and continues through all subsequent phases.

3.2 Risk Identification

Risk is an undesirable situation or circumstance, which has both a probability of occurring and a potential consequence to project success. Risk has an impact on cost, schedule, and performance. Risk identification is the process of identifying uncertainty within all aspects of a project. In other words, what might go wrong and what happens if it does. For most information system projects, these risks may be grouped in the following categories:

What are the categories of risks to information system projects?

- **Technical.** Risk associated with creating a new capability or capacity
- **Supportability.** Risk associated with implementing, operating, and maintaining a new capability
- **Programmatic.** Risk caused by events outside the project's control, such as public law changes
- **Cost and Schedule.** Risk where cost or schedule estimates are inaccurate or planned efficiencies are not realized

Risks should be identified continuously by project participants (at all levels) and the project management team should capture these risks in definitive statements of probability and impact. Lessons learned from previous projects may be a significant source for identifying potential risks on a new project.

3.3 Risk Analysis

Risk Analysis quantifies the identified risks and conducts detailed sensitivity studies of the most critical variables involved. The outcome of these analyses may be a quantified list of probabilities of occurrence and consequences that may be combined into a single numerical score. This single score allows project risks to be prioritized.

3.4 Risk Planning

Risk planning decides what to do about a project risk. Available actions are:

- Avoid the risk.
- Control the risk
- Assume the risk
- Transfer the risk

The action selected for each risk will depend on the project phase, the options that are available, and the resources that can be used for risk management. A majority of project activities involve tracking and controlling the project risk.

3.5 Risk Tracking

Risk tracking involves gathering and analyzing project information that measures risk. For example, test reports, design reviews, and configuration audits are risk tracking tools used by project management to assess the technical risk of moving forward into the next life cycle phase.

3.5 Risk Control

Risk control takes the results of risk tracking and decides what to do and then does it. For example, if a project design review shows inadequate progress in one area, the decision may be made to change technical approaches or delay the project.

3.5.1 Risk Mitigation Techniques

Risk mitigation techniques are used to control or transfer risk until an acceptable risk level is reached. The most common techniques are inherent in good management and engineering practice:

- Budget management reserve - mitigates cost risk.
- Schedule slack - mitigates schedule risk.
- Parallel development - mitigates technical risk.
- Prototyping - mitigates technical risk.
- Incentive fee and incentive-firm contract types - mitigates cost risk.
- Entrance and exit criteria for major design reviews - mitigates cost, schedule and technical risks.

3.5.2 Risk Communication

Risk information should be communicated to all levels of the project organization and to appropriate external organizations. This ensures understanding of the project risks and the planned strategies to address the risk. Risk information then feeds the decision processes within the project and should establish support within external organizations for mitigation activities. For example, an agency comptroller who understands the project risks is more likely to allow the project manager to have a management reserve within the project budget.

Communicating risk information in a clear, understandable, balanced, and useful manner is difficult. The ability to state the problem at hand clearly, concisely, and without ambiguity is essential. Ensure that the mitigation activities conveyed include alternatives, objectively stated justifications and trade off analyses. A well-planned and executed risk management program ensures the decision maker receives unbiased information - resulting in the best project decisions.

4.0 CONCLUSION

Risk is an inevitable factor in all management and development projects. However, appropriate evasive actions, and if risk does threaten the project, then damage-control techniques can be adopted to ensure survivability of the project.

5.0 SUMMARY

In this unit, you have learnt what can constitute a threat (risk) to an information project, and the various actions that can be taken to prevent the risk. You have also learnt the damage- control techniques that can be taken to enhance the survivability of a project exposed to unavoidable risk.

6.0 TUTOR MARKED ASSIGNMENT

- What are the evasive techniques of avoiding risk?
- At what stage is risk management introduced in system development?
- Explain the three risk damage-control or mitigation techniques.

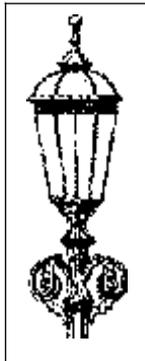
UNIT 5 IT DEVELOPMENT CASE STUDY

In this final unit, we will provide you with a case study of IT software developed by Nicholas Laudato and his colleagues at the University of Pittsburgh in the United States. The document provides you with an idea of how the various components of this course all fit together. At the end of the document is the contact emails for the team, in case you wish to delve further.

*** CASE STUDY

Applying Process Analysis and Prototype Techniques to Developing Media Service Software

Nicholas C. Laudato, Michael J. Arenth, and Patrick L. Hea University of Pittsburgh



Abstract

Forced to replace a legacy media services order entry system, the University engaged in a cross-functional process analysis effort, coupled with rapid application development techniques, to create a reengineered system compliant with established information architecture guidelines. This report discusses the methodologies employed to analyze, develop, and revise the system

Overview

In support of its centralized Instructional Media Services group, the University of Pittsburgh maintained a host-centric application to schedule classroom equipment, services, and personnel.

The software ran on an aging hardware/database platform with extremely limited capabilities and was not Y2K compliant. After an extensive systems investigation failed to identify viable commercial software options, a project was initiated in early 1997 to develop a new software solution.

The new software was designed to comply with the University's established information architecture guidelines. To ensure that the system would be

responsive to customer needs, a cross-functional project team engaged in a process analysis effort, while systems analysis staff simultaneously used rapid application development techniques to generate working prototype applications. The system developed as a result of this two-pronged methodology failed to provide a satisfactory solution.

Using a modified technique, a subsequent process analysis effort identified process disconnects, created a new target envisioned process, and guided the application development modifications. The resultant system successfully addressed the needs of the institution. This paper describes the analysis, development, and revision processes employed to bring the project to fruition

Background

The University of Pittsburgh is a state-related public research university and a member of the Association of American Universities (AAU). Founded in 1787, Pitt offers graduate, undergraduate, professional, and continuing education programs through 19 schools and four regional campuses. The University typically serves about 28,000 FTE in the fall term, and employs nearly 4,000 full- and part-time faculty members. Pitt's regional campuses are located between 30 and 170 miles from the main campus in Oakland, Pittsburgh's cultural and medical center. Among the notable sites on the Oakland campus is the 42-story Cathedral of Learning with its 28 nationality classrooms, each representing the heritage of one of the region's many ethnic groups.

The Center for Instructional Development & Distance Education (CIDDE) works directly with schools, departments, and individual faculty members to facilitate the academic goals of the University in the areas of instructional development and technology, media support, faculty development, and distance education. The Instructional Media Services (IMS) group within CIDDE is responsible for providing technology support for classroom activities and special events. IMS technicians provide operator and technical support for events in the University's media-enhanced facilities and also deliver equipment and materials to rooms that have no built-in technology. The IMS staff of 11 full-time and 25 student employees (part-time) supports over 11,000 media-related events each year.

The problem addressed in this report is the applicability of process analysis and rapid application development methodologies to the creation of responsive information system solutions in a complex university setting. The specific client/server application that was developed addresses the

management of the equipment, services, and staff associated with the delivery of instructional media services across the University.

Systems Investigation

To help manage their classroom media efforts during the 80's and 90's, the Instructional Media Services group used a host-centric application running on an AT&T 3B2 Unix minicomputer. This character-based application was home grown, developed using an early version of Informix, and accessible from VT220 terminals or from microcomputers running the Kermit terminal emulator. The application provided the functionality the department required, but had several problems:

- The system hardware was obsolete and out of maintenance.
- The application software was not Y2K compliant.
- The software had a variety of problems: its functionality was limited, data structures were inflexible and difficult to change, and end users had no control over the application and were not able to perform adhoc reporting.
- The application's character-based interface 'was not intuitive, required a steep learning curve, and was difficult. It for even the experienced operator to use,
- The system had significant performance and reliability problems.
- The original author was no longer employed at the University, and none of the existing information technology staff was familiar with the technologies employed.

By 1997, it was clear that the application had to be replaced. The initial investigation articulated the required functionality and desired features to address the limitations of the old software. An aggressive search of available software packages identified eight candidates, three of which were examined in detail, and one tried on a demo basis. When none of the candidates proved acceptable, a feasibility study was initiated to determine if the application could be written in-house.

Information Architecture Guidelines

In 1994 the University completed a project to define an enterprise-wide information systems architecture. The architecture is based on an information systems philosophy and set of related principles that articulate objectives and quality characteristics for the University Information System. The architecture is intended to guide the analysis, design, and decision-making relative to all aspects of information systems and processes. It determines the technological approach taken in defining components of the architecture and how they must operate, and provides a set of guidelines by which information system design decisions can be made.

For example, to comply with the enterprise architecture, applications should minimally:

- Capture data one time at its source.
- Facilitate flexibility and ease of adapting to changes in policy, to incremental improvements in processes, and advances in technology.
- Implement adopted standards, such as utilizing an SQL-compliant database
- Utilize the client/server model as the basic paradigm.
- Implement a common graphical user interface (GUI).

The authors valued the enterprise guidelines and were committed to building local applications and systems that complied with them.

Process Analysis

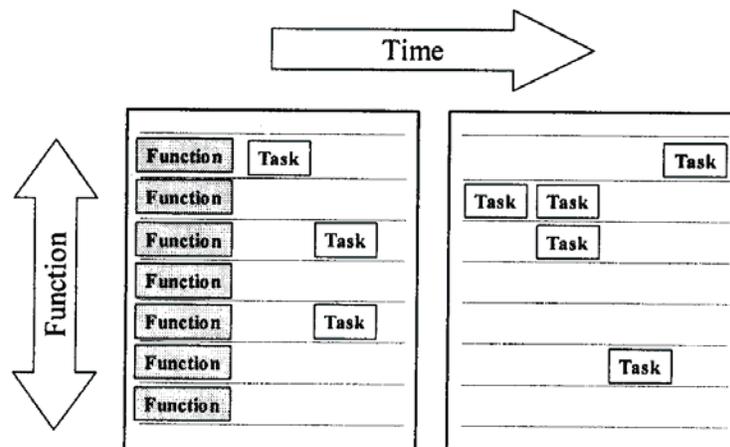
As recommended by the information systems architecture, business processes should be reviewed to determine the need for reengineering or for process improvement. The first step in process analysis is typically process mapping. The University of Pittsburgh uses an approach to process mapping based on the theory and research of Geary A. Rummler and Alan P. Brache, and modified by the work of Kathleen Shade and her colleagues in the Organizational Development department in the University's Human Resources unit.

Process mapping determines how a process currently works with the goal of identifying and implementing solutions to improve the efficiency or effectiveness of that process. Process mapping focuses on the flow of work and people through different areas and departments and the value each step

in the process adds to the final product or service. All of the steps in a process may be contained within one function (e.g. a computer programming process) but most processes are cross-functional, spanning the white spaces between the boxes on an organization chart.

Process mapping is accomplished in a series of meetings run by a facilitator who is expert in leading such sessions. After explaining the process mapping procedures, the facilitator leads the group in painting a picture of the process by identifying the tasks that comprise the process, identifying the responsible job functions of the process participants, and “mapping” the process as a series of steps over time. The resultant map is a two-dimensional grid showing the participants (by job function) on the vertical axis and time on the horizontal axis.

Figure 1: Format for Mapping Processes



A critical success factor for process mapping is ensuring that all of the appropriate stakeholders are represented in the process mapping team. Since all but the simplest processes are cross functional, a strong commitment from senior level management may be necessary to encourage cooperation. The media services process analysis included the following representative users from all steps in the process:

- Process Owner (Manager of Media Services)
- Media Services Staff Supervisors
- Order Entry Specialist

- Equipment Operators
- Inventory Manager
- Classroom Technology Engineer
- Systems Analyst/Programmer
- Billing Manager
- University Auditor
- Process Analysis Facilitator
- Unit Administrator

Customers were conspicuously missing from the cross functional team. In this case, the customers of the service are University faculty. Their needs and input were determined through a facilitated focus group meeting. The meeting was attended by a select group of faculty representing different categories of system patrons, including heavy users of technology, novice users, and faculty with special needs such as those teaching film studies, computer science, and fine arts.

The process team delineated several major processes and completed current process maps for each. The process mapping methodology enabled the participants to move, add, or delete tasks as necessary to ensure a consistent level of documentation, the activity was useful in enabling all team participants to understand and agree on all of the steps of the process, as it currently existed. One of the most difficult challenges in this phase of process mapping is to delineate the process as it currently is, rather than as the participants think it should be operating. For this reason, the current process map is often called the “is” map,

After the process maps were completed, the team began to identify every possible thing that could go wrong at each step throughout the process, These “things that go wrong,” called “disconnects,” were placed on the map and described separately. Figure 2 shows a partial excerpt from the “office deliveries” process, labeled with its process disconnects. Figure 3 shows a brief description of some of the identified disconnects.

Figure 2: Excerpt From Current Process Map

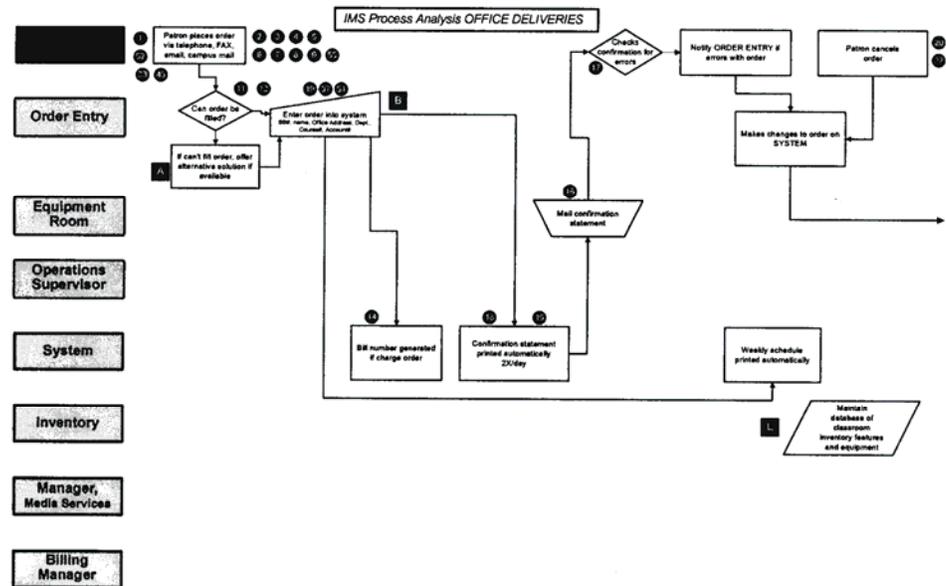


Figure 3: Excerpt From Disconnect List

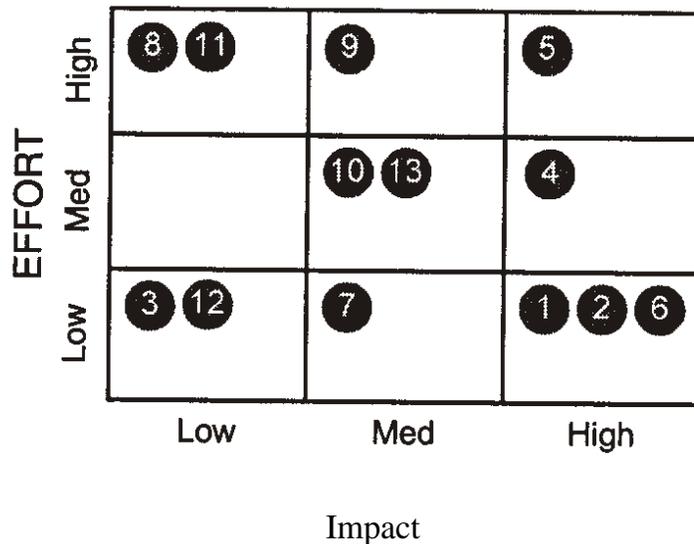
- Multiple people/areas may need to be consulted for more complex jobs.
- Insufficient information from patron to determine if job can be filled.
- Staff (Operations Supervisor, Inventory Manager, Coordinator) not available for consultation.
- Patron does not understand terminology.
- Patrons not familiar with resources and/or limitations of media services.
- Patrons place orders at last minute.
- Difficult to read patrons' writing, especially on Fax. Fax may not arrive.
- Patrons may want "personal" confirmation, so may not use www.
- Web page does not provide as much information as some might like (Film/Video catalog).

- System runs slowly.
- Patrons try to circumvent the system (appeal to supervisor if denied services),
- Patrons don't want to accept alternative of setting up equipment themselves,
- Patrons provide no account number or inaccurate account number.

Identifying and describing the process disconnects was another difficult challenge. The process mapping facilitator went to great lengths to deflect the natural defensiveness of the project participants and keep them on purpose. The process disconnects were subsequently analyzed for their causes and for their impact on the process. The disconnects were plotted onto a grid to reflect the effort required to address the disconnect versus the impact on the process if the disconnect were eliminated (see Figure 4).

The disconnect analysis, ordinarily a tool for process improvement activities, helped define areas that the newly defined process should address.

Figure 4: Disconnect Analysis



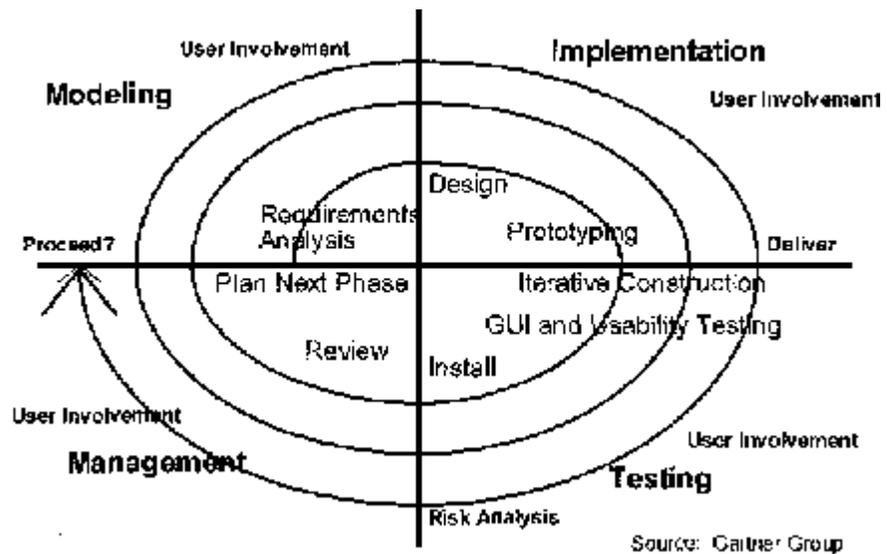
The last step in the process analysis was to create a new process map, called the “envisioned” or “should” map, that could guide system development decisions. The challenge the participants faced in mapping the envisioned process included the resistance to change natural to all participants in a

process and the difficulty in thinking “outside the box” as part of a general brainstorming session. The facilitator helped the group to consider suggestions that would ordinarily be summarily dismissed. The only negative aspect of the process mapping was that it took an inordinate amount of time due to difficulties in scheduling the facilitator.

Systems Analysis, Design and Implementation

Based on prior experience as members of the information architecture team, the authors were advocates of the rapid application development methodology. The systems analyst assigned to the project immediately began developing a prototype application and engaging end users in its assessment and refinement. Figure 5 depicts the PAD methodology, adopted from the Gartner Group. The most important feature of the methodology is the commitment to continuous user involvement throughout the application development life cycle.

Figure 5; Application Development Methodology



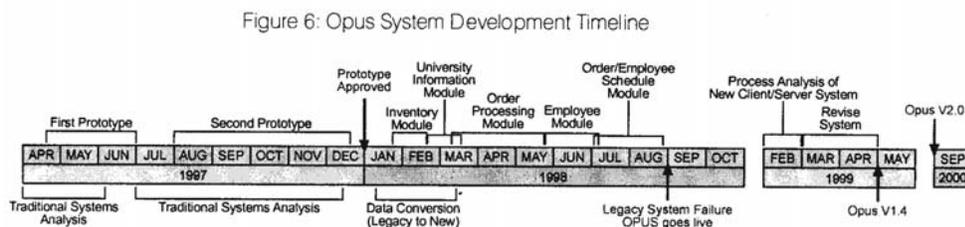
The prototyping technique involved first performing a review of the existing system and designing and normalizing the new database. Because of the rapidly changing nature of classroom technologies and the services associated with them, it was critical that the new system be designed with as much flexibility as possible. The system would need to be responsive to end users, and not require time-consuming development work to add new functionality and database elements.

To accomplish these goals, the application was designed to utilize “metadata,” or data about data. In the metadata approach, the content, appearance, and functionality of the application windows are driven by metadata elements and dynamically configured at run time, as opposed to being hard-coded in the application. The application therefore allows real-time additions and modifications to user forms without requiring further database work to add new classes of equipment.

The new application was named Opus, an acronym for “Order Processing System for University Media Services.” Opus is a client/server application, developed in Visual Basic, and running on a Windows NT server utilizing Microsoft SQL Server 6.5. The server is configured with mirrored disk and automated backup. Standard Windows NT and Windows 2000 PCs are used as clients, as well as a small number of Net PCs for the “will-call” counter.

The implementation planned for converting the legacy system data into the new system was to follow a parallel scheme. The parallel conversion method is staff intensive, as it requires duplicate data entry into both the old and new system, but it is the most reliable method for critical functions such as classroom support. Unfortunately, the legacy platform failed two weeks prior to the planned implementation date, and the beta version of Opus was forced into production prematurely (the “plunge” conversion method). Figure 6 depicts the development and implementation timeline.

Figure 6: Opus System Development timeline.



Systems Maintenance and Evaluation

When initial systems development was complete and the system became operational, attention was turned to identifying and correcting bugs and problems. After the system stabilized, it was clear that it did not live up to expectations. Users complained of performance problems and of awkward sequences of steps to perform some operations. The new system did not accurately implement the envisioned process map.

How had the system deviated from the vision? Perhaps engaging in application prototyping while simultaneously pursuing process engineering was a mistake. In retrospect, the stark contrast between the evolving GUI of the new prototype and the antiquated character-based legacy application had the effect of limiting the creativity of the project participants during the envisioning process. The new prototype was so much better than the legacy system that the users were willing to jump to the conclusion that it would meet all of their needs,

When it became clear that the new system had flaws, a second smaller process-mapping group was reconvened with a new facilitator. The facilitator reviewed the process analysis methodology with the participants and guided them through a mapping of the newly implemented system in a timely manner. This effort, conducted with a more mature and demanding group of users, produced an envisioned process map that guided the revision of Opus, phased over two releases.

The Resultant Media Order Entry System.

Opus provides a comprehensive media services management system. Its metadata driven design allows end users to redefine its functionality at both the database and application levels in real time. Its reports are available on demand via the on-line graphical interface and in ad hoc reports via a Microsoft Access interface to defined logical user views. Opus includes the following functionality:

- Real-time scheduling of equipment, media, and staff associated with campus locations,
- Maintenance of patron, staff, building, equipment, media, and classroom records.
- Management and tracking of equipment inventory, utilizing barcode scanning,

- Integration of an associated film/video collection database.
- Creation of tailored customer confirmations.
- Graphical scheduling of staff assignments.
- Collection, processing and reporting of staff payroll data.
- Creation of bills and financial reports.

Summary and Conclusion

The project was notable for several reasons. First, it was guided by principles and standards articulated in the University's established information architecture. These guidelines resulted in a multi-tiered client/server application designed using application metadata to create a flexible, maintainable, and customizable (in real-time) application. Second, the project utilized a process analysis methodology that mapped the existing business process, identified and analyzed process disconnects, and built an ideal envisioned process. Third, the project used rapid application development techniques to design a working prototype application. Finally, when these methodologies produced less than-satisfactory results, a revised process analysis effort led the project to a successful completion.

The project, through the use of sound architectural principles, prototyping techniques, and the successive application of process analysis methodologies, resulted in a successful implementation, fully meeting the requirements of all stakeholders in the process. The following lessons were learned from the project:

Process analysis can improve the responsiveness and functionality of application development efforts. Critical success factors include:

- Creating a cross-functional process analysis team.
- Obtaining senior management support and involvement.
- Performing the process analysis in a timely fashion.

Prototyping is an effective methodology for designing and implementing responsive and user-friendly systems. Critical success factors include:

- Development must be rapid and responsive to end user feedback.
- Communication between developers and end users must be frequent and rich,
- The timing of process analysis versus prototyping activities is crucial. Process analysis should precede prototyping.

About the Authors

Center for Instructional Development & Distance Education, University of Pittsburgh



Nick Laudato
Associate Director for
Instructional Technology
laudato@pitt.edu
(412) 624-3335



Michael Arenth
Assistant Director for
Instructional Media Services
arenth@cidde.pitt.edu
(412) 648-7236



Patrick Hetherington
Manager of Instructional
Computing
hetherington@cidde.pitt.edu
(412) 624-1113

MODULE 6 MANAGEMENT INFORMATION SYSTEM

Unit 1	Theory of Management
Unit 2	Organizational Concepts
Unit 3	Management Information Systems (MIS)
Unit 4	Risks Associated With MIS
Unit 5	Information systems and databases

UNIT 1 THEORY OF MANAGEMENT

CONTENTS

1.0	Introduction
2.0	Objectives
3.0	Main Body
3.1	Overview
3.2	Management of Organization
3.2.1	The Functional Approach to Management
3.2.2	The Strategic Information Centre
3.3	Design and Implementation Consideration
3.4	Problems of Management and Information Systems
4.0	Conclusion
5.0	Summary
6.0	Tutor Marked Assignment

1.0 INTRODUCTION

The system designs created by system analysts must operate within the confines of management structures. It is therefore vital for system analysts to have a grasp of the theory of management before conducting analysis and design. In any event, they will be dealing with managers a lot, so, understanding the mechanism of the management process makes it easier to anticipate the needs of managers in system designs.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the divisional structure of organizational management
- Explain the functional approach to management
- Describe the structure of the Strategic Information Center

- Explain the various design alternatives in an Information Support System setting

3.0 MAIN CONTENT

3.1 Overview

It is useful to systems analysts to understand the management process. To accomplish this, we will develop a theory of management that has been tested and validated in a number of case studies. This theory is helpful in designing and implementing management information and decision support systems.

3.2 Management of Organizations

To begin with, the management of an organization is normally divided into three levels. These levels are strategic, tactical, and operational. The strategic level of management consists of top-level managers, such as the president and vice presidents. The tactical level consists of middle-level managers and decision makers and the operational level consists of lower-level managers and decision makers. The source of data and information, the presentation of information, and the use of information vary from one level to the next.

Strategic or top-level managers should have primarily external sources of data; the information should be summarized, and the use of the information is primarily external. On the other hand, operational or low-level managers should have primarily internal sources of data, the presentation should be detailed, and the use of the information is normally internal. Tactical or middle managers are somewhat between the strategic and operational managers, concerning the source, presentation, and use of information.

Since a management information or decision support system helps managers perform various managerial functions, it is important to understand how the levels of management perform these functions; In general, managers are responsible for planning, organizing, staffing, directing, and controlling. How managers and decision makers perform these functions depends on the managerial level (strategic, tactical, and operational).

The three managerial levels have slightly different orientations for the five managerial functions. Strategic managers are involved with long-range planning, high-level organizing, high-level staffing, overall directing, and

overall control. Operational managers perform short-range planning, low-level organizing, low-level staffing, detailed directing, and detailed controlling. As before, tactical managers fall between strategic and operational managers.

The decision-making process also varies depending upon the level (strategic, tactical, or operational) of the particular manager. For strategic managers, the degree of problem structure is low, the time horizon is long, the amount of judgment is high, and each problem is usually unique. For operational managers, the degree of problem structure is high, the time horizon is short, the amount of judgment is low, and the decisions are usually routine. Tactical managers fall between these ranges in the decision-making process.

3.2.1 The Functional Approach to Management

During the initial development of MIS (Management Information System) and DSS (Decision Support System) concepts (see unit for further explanation of these concepts), the goal was to develop an integrated system. This system, theoretically, would be used by all managers and decision makers in the organization. Unfortunately, these systems were just too large and complex. As a result, many of the initial efforts ended in failure. This disappointing start prompted data processing personnel to seek better ways of developing these systems. Although there is no one best way, many organizations adopted the functional approach.

Most organizations are organized along functional lines or areas. This is usually apparent from the organization chart, which shows functional vice presidents under the president. Some of the traditional functional areas are: (1) accounting, (2) finance, (3) marketing, (4) personnel, (5) research and development, (6) legal services, (7) operations/ production management, and (8) information services. Furthermore, each of these areas require different information and support for decision making. In addition, each of these functional areas within the organization contains the various levels of management described previously in this unit. Thus, in addition to horizontally slicing management into three levels (strategic, tactical, and operational), there is also a need to vertically slice management into the various functional areas.

Using a functional approach, a different information or decision support system is developed for the various functional areas and the different levels of management within each functional area. Thus, there would be a system for top-level managers in accounting, middle-level managers in accounting,

lower-level managers in accounting, top-level managers in finance, middle-level managers in finance, lower-level managers in finance, and so on. Each system is designed to meet the particular needs of each group of managers.

When a functional approach is taken, it is necessary to try to relate or tie together the various management information and decision support systems. Otherwise, the **organization** might end up with a collection of disjointed and ineffective systems. With the increased use of data bases and data base management systems (DBMS), one way to unify and integrate the various systems is through a common data base. Data bases and data base management systems are discussed in the next supplement. Working with a common data base and a DBMS, it is possible to develop separate data bases and application programs for each functional managerial level.

3.2.2 The Strategic Information Center

In order to help managers get the information they need when they need it, the overall concept of a strategic information center has been developed. A strategic information center has the overall purpose of providing managers with the information needed to make effective decisions. The strategic information center can be located within or outside the data processing department. The sole function of the center members is to be resource personnel for top-level managers and decision makers.

Most strategic information centers seek information for managers that is both internal and external to the computer system. When data is stored on the computer, members of the strategic information center develop programs and commands that obtain the information that managers want. In some cases, the desired information is not contained in computer files. Then, center members seek other sources of information, such as annual reports, trade journals, legal documents, newspapers and magazines, and advice from experts and consultants.

3.3 Design and Implementation Considerations

In the above sections, we have emphasized that different functional areas and different managerial levels require different management information and decision support systems. The differences discussed in the above sections should be taken into account in designing and implementing a management information system or a decision support system.

There are many design alternatives that can be used in an information and decision support system. The system can be centralized, decentralized, or

distributed. A hybrid approach, where various components of the system can be either centralized, decentralized, or distributed is also possible. These components include hardware, software, personnel, data base, and control. For example, hardware and software can be decentralized, while personnel, data bases, and overall control are centralized. Another option would be to have the hardware and personnel decentralized while everything else is centralized. In other words, it is possible to centralize, decentralize, or distribute the hardware, software, data processing personnel, data bases employed, and overall control of the entire system.

The overall approach in designing and implementing an MIS or a DSS is the same for systems analysis and design. For a complete description of design and implementation procedures, refer to the unit on systems analysis and design. There are, however, problems and circumstances particularly important in the design and implementation of an MIS or a DSS.

3.4 Problems of Management and Information Systems

To begin with, there are a number of problems that have blocked the success of an MIS or a DSS. Some of these problems are:

1. Lack of goals and objectives for the MIS or DSS.
2. Lack of involvement of managers at all levels within the organization.
3. Too much emphasis placed on technical aspects of the system,
4. Not enough emphasis placed on human factors.
5. Over reliance on data processing personnel.
6. Lack of flexibility in the MIS or DSS.
7. False assumptions made by data processing professionals in developing the DSS or MIS.
8. Inability of data processing personnel to understand the needs of management during the design stage.
9. An inadequate or misdirected implementation of the MIS or DSS.
10. Inadequate control and ineffective evaluation and maintenance of the MIS or DSS after it has been installed.

One of the major problems in designing and implementing an MIS or a DSS has been due to human factors. Normally, there has been a resistance to these new systems and to change in general. Some managers have perceived these new systems as a threat to power, a threat to status and position, and a threat to financial and job security. Managers have also perceived these new systems as creating job and role uncertainties and ambiguities; changing the relationships between top- middle and lower-level managers, increasing job pressures, and increasing job complexity.

A successfully designed and implemented MIS or DSS overcomes the above problems or solves them before they become a serious threat. One solution suggested by many people is to get the managers involved in the new system. But involvement by itself is not enough. The involvement should directly tie them to the success of the system. One of the best approaches is to have the manager or user actually conceive or develop the solution. Managers should also be involved in evaluating the alternatives, selecting the best alternative, designing the new system, and implementing the new system. The overall objective is to make the new system the manager's system.

During implementation, the new system should be marketed and sold to managers like a new product is marketed and sold to consumers. Some companies have used brochures, posters, and informal seminars with great success to advertise, promote, and personally sell the new system to managers. The actual implementation of the new system should be slow, smooth, and not threatening. The people in charge of the implementation are change agents. They need to have both good interpersonal and organizational development skills and good technical skills. The implementation process requires three phases: (1) unfreezing, (2) moving, and (3) refreezing. Unfreezing is the process of removing or eliminating the old procedures and systems; moving is the phase of having managers get used to the new system; and refreezing is the process of establishing new habits and behaviors concerning the use of the new system.

4.0 CONCLUSION

We have seen therefore how management information or decision support system helps managers perform various managerial functions. It is important for systems analysts and designers to understand how the levels of management perform these functions so that a more effective system of management of information can be designed for their organizations. In general, managers are responsible for planning, organizing, staffing,

directing, and controlling. They can only do this efficiently if an effective interface is created for them.

5.0 SUMMARY

In this unit, you have learnt about the divisional structure of organizational management. The various approaches to management have also been explained, which will provide the analyst with a better understanding of how to approach organizational system analysis. The unit has also explained the significance of strategic information center, and how it plays a vital role in providing managers with all the information they need, and when they need it. Finally, the unit explained the problems that have blocked the success of Management Information Systems in most organizations.

6.0 TUTOR MARKED ASSIGNMENT

- Distinguish between strategic and operational managers in an organization
- Explain the functional approach to management
- Describe the structure of the Strategic Information Center
- Explain the various design alternatives in an Information Support System setting
- Explain at least five problems management information systems in organizational structures.

UNIT 2 ORGANIZATIONAL CONCEPTS

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Organization Purpose
 - 3.3 Organization Culture
 - 3.4 Organization Structure
 - 3.5 Business Organizations
 - 3.6 Information Needs of Organizations
 - 3.7 The General Model of the Organization
 - 3.8 Organizational Models
 - 3.9 Organizational Factors
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

Organizations such as businesses and government agencies are good examples of the subsystems of society, which is their environment. Organizations themselves consist of many subsystems, such as departments, divisions, and other work groups. Organizations are examples of open systems, since they interface and interact with other systems in their environment. Organizations are also examples of adaptive systems, since they can modify themselves to meet the demands of a changing environment.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Define organizational purpose
- Explain the different types of organizational culture
- Outline the different departments of a business organization
- List the information needs of organizations

3.0 MAIN CONTENT

3.1 Overview

At this point, it is useful to apply system terminology to a typical organization. An organization is a formal collection of people and various other resources established to accomplish some set of goals. All organizations receive inputs (resources such as materials, people, money etc.) and produces outputs (goods and/or services) by using a suitable transformation mechanism to achieve some purpose/set of goals within some environment. An organization is part of many environments - physical (geography and ecological), social, political and technological. A changing state of the environment drives change in the organization to maintain consistency with purpose. An organization can affect its environment.

3.2 Organizational Purpose

Organizational purpose is the glue that holds the organization together. An organization will have several purposes: value to customers, providing excellent working conditions, high returns for shareholders, etc. An organizations manager's work to bring the system to a state that is consistent with its purpose/goal.

3.3 Organization Culture

Organization culture consists of the major understandings and assumptions for a business, corporation, or organization. It also includes common beliefs, values, and approaches to decision making. Culture can have a significant impact on the development and operation of information systems within the organization.

3.4 Organization Structure

Organization structure refers to organizational sub-units and how they are related and tied into the overall organization. How an organization is structured influences its information needs. Types of organization structure:

- **Traditional organization structure:** Divided according to function, line position and staff position. It has a hierarchical structure.

- **Flat organization structure:** a trend to reduce the number of management levels and span of control. Empowerment giving more power, responsibility, and authority to employees
- **Project organization structure:** centered on major products or services. Most project teams are temporary.
- **Team organization structure:** centered around work teams or groups
- **Multidimensional organization structure:** it may incorporate several structures at the same time, both traditional functional areas and major project units. Matrix structure.

3.5 Business Organizations

A business is an organizational system where economic resources (input) are transformed by various organizational processes (processing) into goods and services (output). A business organization exists to make money, to provide a service and to control people and resources. Organizations vary in size and in their aims and objectives e.g. business organizations trying to make a profit, charities trying to raise money, service organizations such as the Health Service, special interest groups trying to change public opinion etc.

To achieve its objectives, a business organization creates business work areas or sub-systems (departments, units, and teams) each of which is designed to pursue its own goals. Combining the work and the goals of these individual departments should result in the achievement of the overall organizational objective.

Many business organizations contain the following departments, which interact in order to achieve organizational goals:

- **Sales:** Task is to sell the products of the business.
- **Production:** Concerned with the internal processes of producing the goods and services and also ensuring that in this production process, raw materials and resources are being used in the most efficient manner.

- **Marketing:** Concerned with the external links to prospective customers and in determining the potential demand for a product through market research and advertising.
- **Purchasing:** Concerned with buying the necessary raw materials so that an organization can produce goods/services to meet the demands of its customers. This will involve forming contracts with suppliers and ensuring that materials are received on time and in the necessary quantities.
- **Accounting:** Financial accounting looks at what has been done with the money in the past while management accounting is concerned with deciding how capital can be best employed in the future.
- **Corporate Management:** Task is to co-ordinate all of the organization's functions. For example, in marketing and production there is frequently a conflict between. Production prefers a uniform output of a limited range of goods throughout the year, marketing on the other hand is likely to encourage a wide product base in the hope of attracting new customers.
- **Personnel:** Personnel is often responsible for manpower planning, recruitment and selection, education and training, payment, industrial relations etc.

Every area has links with some or all of the others areas. For example, in a manufacturing organization, stock control receives data, concerning materials, from the purchasing department and passes information, about components required, to production control. It is important that there is close liaison between the sales department and the production department to ensure that the sales team does not take more orders than the production team can produce goods.

Depending on the size of the organization, each department may be responsible for one particular aspect of the business, or in smaller companies, one person may be responsible for a number of different job functions. For example, in a one-man business, the owner will undertake production, sales, marketing, accounting etc. on his own. In a larger company, each of these functions may require a large number of people. Departments then need to set their own goals and objectives but subject to the organizational goals in order to avoid sub-optimization.

Management, either at departmental or organizational level, needs to receive feedback on the performance of the systems for which they are responsible. This feedback comes in the form of information on which management can act as necessary. All business areas have data and information requirements. The information system is a subsystem within a business. Information systems are a critical part of the profitability and survival of many organizations

3.6 Information Needs of Organizations

All organizations must record information about their operations. The quantity and type of information recorded and the methods used to record it depend on the size and nature of the business. Information systems provide the means for ensuring efficient and effective management of information. The sorts of information that organizations need are:

- **Transactions:** Done for every transaction in the organization. Transactions are usually standard for each event/customer/etc. Includes very detailed specific information and usually done by relatively unskilled staff. Sales example: need to record details of every sales transaction to keep track of income.
- **Managing/Controlling:** is based on analysis of aggregate information from many transactions. It uses overall trends or patterns, not detailed specific information for individual transactions. Often involves recognizing and responding to unusual or unexpected results i.e. something has gone wrong, so action is needed. Done on a short or medium-term basis - daily/weekly/monthly and usually done by managers or skilled staff. Finance example: monitoring cash flow, ensuring accounts (receivable and payable) are being kept under control.
- **Planning:** Based on analysis of aggregate information of many types - transaction, business, industry, economy, etc. Needs long-term trends or patterns and requires prediction of future trends, etc. Done on a medium or long-term basis -quarterly/yearly/5 year and done by senior management. Stock control example: determining most suitable stock management policies for optimum stock turnaround/minimum wastage/etc e.g. just-in-time techniques, automatic ordering systems.

Note the number and variety of information needs the business may have. Each may need a system of its own. Note that although the basic

information requirements may be the same, every organization will have a unique blend of specific needs and ways of meeting them. You can have 'standard' systems (packages), but every business will have its own variations. Therefore the system developer must analyze the information needs individually for every system.

Only when you understand the business and its information flows can you know what data needs to be input, stored, processed and output; what interactions there are between different parts of the system; what interactions the system has with the outside world, etc.

If you don't know what sorts of information the business needs and how it uses it, how can you know what sort of systems to build to help it! Business people know their business well, but they don't know about technology. Technologists know technology well, but they don't know about businesses. Analysts need to be able to fill the gap by knowing about both!

3.7 The General Model of the Organization

With all of the various flows through the organization and the complexity of the organization and its environment, it would seem impossible to develop a general model of the organization. Since we are interested primarily in data, information, and decision flows, we can de-emphasize or eliminate the other flows, such as material, manpower, and money flows. In addition, since we are interested in developing a general model of the organization, we can de-emphasize or eliminate detail from the model. We should stop here to make an important point.

3.8 Organizational Models

A model is developed for a purpose. In this case, we want to develop a general model of the organization to study data, information, and decision flows. Thus, we will include only as much detail as is necessary to analyze the problem or situation. With this in mind, a generalized model of the organization can be developed. See Fig.8.1.

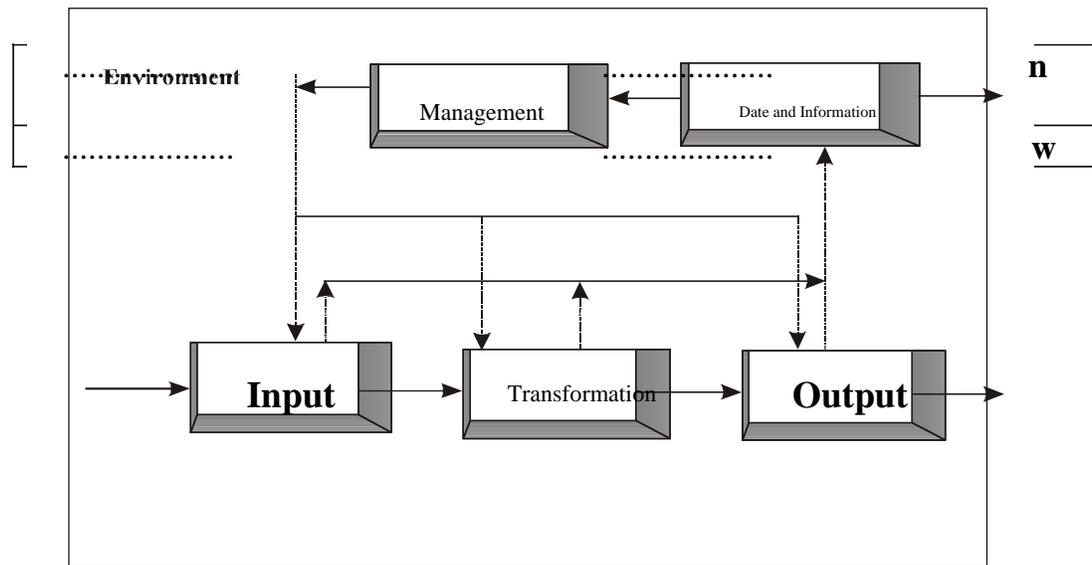


Fig 8.1: General model of the organization

As seen in Fig. 8.1, materials, manpower, money, and so forth, are input from the environment, go through a transformation process, and are output to the environment. The outputs are usually goods and services. Data flows from the inputs, the transformation process, and the outputs to the data and information processor. If a computer is used, the information and data processor is another name for the computer system. This data is then manipulated by the information and data processor to produce meaningful information used by managers to make decisions. These decisions can affect inputs, the transformation process, and outputs. These decisions, for example, could be to use different raw materials (inputs), to acquire a new piece of manufacturing equipment (transformation process), or to open new channels of distribution for the products produced (outputs).

You will also note the data and information processor produces data that goes back to the environment. This data is usually business documents and transactions. Examples of this data output are paychecks, invoices, bills, checks to suppliers, information to stockholders, and tax returns.

3.8 Organizational Factors

In addition to the general model of the organization, there are several organizational factors important in the design and development of an information or a decision support system. Some of the more important factors include:

- Purpose of the organization.
- Size of the organization,
- Structure of the organization.
- Philosophy of the organization.

4.0 CONCLUSION

An organization is a dynamic and complex system. Money, manpower, materials, machines and equipment, data, information, and decisions are constantly flowing through the organization. The organization is operating in a dynamic, complex environment. The environment is defined as everything not included as part of the organization. In other words, the organization and its environment make up the entire universe.

5.0 SUMMARY

In this unit you have learnt about organizations, a necessary steps towards understanding their structures and function, and consequently how they can be broken down into component structures for easier information access development strategies.

6.0 TUTOR MARKED ASSIGNMENT

Explain the relationship between the different departments of a business organization at:

- Explain at least three types of organizational structure
- Explain the three types of information that organizations need
- Using a clearly labeled diagram, develop a model of an organizational model

- List the organizational factors that are crucial in the design and development an information system.

UNIT 3 MANAGEMENT INFORMATION SYSTEMS (MIS)

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 MIS and DSS
 - 3.3 Emergence of DSS
 - 3.4 The Information Resource
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References and Further Readings

1.0 INTRODUCTION

We are now beginning to move towards an understanding of the fundamental concepts of real-world information systems in organization. The corporate structure that deals with the design and access to information is called Management Information Systems, or MIS.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Distinguish between Management Information Systems (MIS) and Decision Support Systems (DSS) “Explain the mechanism of Decision Support Systems
- Explain Information Resource Management (IRM)

3.0 MAIN CONTENT

3.1 Overview

The field of Management Information Systems (MIS) addresses the effective use of human and computer resources to realize important business objectives. MIS professionals are responsible for developing information systems that provide accurate and timely information to all

levels of decision-making in a business organization. MIS has evolved into an exciting area of study in which students are exposed to leading-edge technologies and trained in their deployment. With information systems permeating every functional area of business, this knowledge has become an essential component of a student's educational preparation.

3.2 MIS and DSS

A management information system (MIS) is an organized collection of people, procedures, and devices used to provide information. This information can relate to internal and external intelligence, and it can assist with planning, staffing, organizing, directing, and controlling. The overall purpose of a management information system is to provide the right information to the right manager or decision maker at the right time. As defined, a management information system is a specialized data processing system. All management information systems are also data processing systems, but only those data processing systems that provide useful information are considered management information systems.

A decision support system (DSS) is an organized collection of people, procedures, and devices used to support decision making. While decision support systems and management information systems appear to be the same, there are important differences. The major impact of a management information system is on structured tasks, and the major payoff has been to improve efficiency. In many cases, managers play a passive role in the development of the system. A decision support system has an impact on both structured tasks and unstructured tasks that require managerial judgment, and the major payoff has been to improve effectiveness. In addition, managers play an active role in the development and implementation of the system. A decision support system works from a managerial perspective, and it recognizes that different managerial styles and decision types require different systems. A decision support system can have a significant impact on the decision-making process, while a management information system does not. The overall emphasis is to support rather than replace managerial decision making.

3.3 Emergence of DSS

There is a great deal of excitement about the potential of decision support systems. New products are being developed that claim to be decision support systems, and academics are conducting numerous research projects to investigate this new computer-related tool. Due to problems with

traditional management information systems, there may be advantages to taking a fresh, new approach.

Decision support systems concentrate on a particular manager's decision-making style. The overall process involves investigating a particular problem, finding out how a manager solves this problem, seeing what actual information is needed, and then developing tools and techniques to generate the information and to support the decision-making process in general. In other words, the decision support system ties directly into an individual's decision-making style. In some cases, small systems are developed, run, and tested to make sure they are satisfying a decision maker's needs. This approach is called prototyping. With prototyping, a small prototype, or model, is made on the computer. This prototype is then tested and refined numerous times until a manager is satisfied with the decision support received from the computer system. This is unlike the traditional approach, which first generated information needs and then developed computer programs to retrieve the information without testing to make sure the information needs were legitimate.

This new excitement does not mean that all of the manager's problems will be solved by the development and use of a decision support system. Indeed, many data processing experts are skeptical of decision support systems. Other people feel that a decision support system is a lot of hype for a technique that is not radically different from an information system.

Regardless of the terminology used, there is a need for an effective system to help managers make good decisions. These systems must produce the right type and right amount of information. Too much information can be as dangerous as too little. During a crisis or critical corporate situation, all information should be turned off except that information directly dealing with the crisis, problem, or situation at hand.

3.5 The Information Resource

One of the most difficult problems facing MIS and DSS designers is managing a vast information resource. Many organizations have huge data bases containing millions of facts. In many of these organizations, all of this data and information is just sitting there. There is no effective way to gain access to it. In an attempt to manage data and information, a number of organizations have adopted the notion of information resource management (IRM). The idea is to consider information like any other resource in an organization. Like labor, capital, and raw materials, information needs to be

managed efficiently and effectively. Indeed, for many organizations, information can be the most valuable asset.

Even with the use of modern data base management systems, managing information can be difficult. Today it is impossible for most organizations to place all data on direct access storage devices and use data base management systems for immediate retrieval. Thus, it is important to identify information and data within the data base that is critical and requires immediate access. Academicians and professionals are investigating ways to identify important information in large data bases. This is called information requirements analysis. In the past, managers were asked to make a list of the information items important to their jobs. Unfortunately, many managers do not know what information they need. As a result, formalized procedures have been developed to determine information requirements for managers and decision makers. One such technique is critical success factors (CSF). This technique, developed at MIT, has shown early success in helping managers determine what information they need to perform their jobs. Typically, three to six hours of interviews are conducted in two or more separate sessions. The first interview session outlines a manager's goals and objectives. The important factors needed to achieve these goals are then determined. These factors are called critical success factors (which is how the technique was named). During the first interview session, the relationships between the goals and the CSFs are determined and clarified. During the second interview session, the results of the first interview are reviewed and refined. Then possible reports that can be generated to obtain corporate goals are discussed in depth. In some cases, a third or even fourth session may be required to obtain agreement on the critical success factors and the reports needed to help obtain corporate goals.

4.0 CONCLUSION

MIS and DSS provide mechanisms around which information, no matter how vast or complex in an organization, can be tamed and channeled into an effective weapon of organizational success and enhancement.

5.0 SUMMARY

In this unit you have learnt further techniques of organizational management, with particular emphasis on the emergence of Management Information Systems and Decision Support Systems. All these basically reflect the social interface of systems analysis and design.

6.0 TUTOR MARKED ASSIGNMENT

- Define MIS, and distinguish it from DSS
- What is the methodology of Decision Support Systems?

UNIT 4 RISKS ASSOCIATED WITH MIS

CONTENT

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Assessing Vulnerability To MIS Risk
 - 3.2.1 Accuracy
 - 3.2.2 Consistency
 - 3.2.3 Completeness
 - 3.2.4 Relevance
 - 3.3 Achieving Sound MIS
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment
- 7.0 References and Further Readings

1.0 INTRODUCTION

Perhaps one of the greatest barriers to full scale introduction of MIS and DSS techniques in organizations — and thus hampering the work of systems analysts — is the fear of the risk associated with information husbandry. In an increasingly competitive world, organization information needs to be secure so as not to reveal the vulnerabilities of the organization.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Describe the elements of a useable MIS
- Explain the sound methodology to ensure the security of MIS developed systems

3.0 MAIN CONTENT

3.1 Overview

Risk reflects the potential, the likelihood, or the expectation of events that could adversely affect earnings or capital. Management uses MIS to help in the assessment of risk within an institution. Management decisions based upon ineffective, inaccurate, or incomplete MIS may increase risk in a number of areas such as credit quality, liquidity, market/pricing, interest rate, or foreign currency. A flawed MIS causes operational risks and can adversely affect an organization's monitoring of its fiduciary, consumer, fair lending, Bank Secrecy Act, or other compliance-related activities.

Since management requires information to assess and monitor performance at all levels of the organization, MIS risk can extend to all levels of the operations. Additionally, poorly programmed or non-secure systems in which data can be manipulated and/or systems requiring ongoing repairs can easily disrupt routine work flow and can lead to incorrect decisions or impaired planning.

3.2 Assessing Vulnerability to MIS Risk

To function effectively as an interacting, interrelated, and interdependent feedback tool for management and staff, MIS must be "useable." The five elements of a useable MIS system are: **timeliness, accuracy, consistency, completeness, and relevance.** The usefulness of MIS is hindered whenever one or more of these elements is compromised. Timeliness, to simplify prompt decision making, an institution's MIS should be capable of providing and distributing current information to appropriate users. Information systems should be designed to expedite reporting of information.

The system should be able to quickly collect and edit data, summarize results, and be able to adjust and correct errors promptly.

3.2.1 Accuracy

A sound system of automated and manual internal controls must exist throughout all information systems processing activities. Information should receive appropriate editing, balancing, and internal control checks. A comprehensive internal and external audit program should be employed to ensure the adequacy of internal controls.

3.2.2 Consistency

To be reliable, data should be processed and compiled consistently and uniformly. Variations in how data is collected and reported can distort information and trend analysis. In addition, because data collection and reporting processes will change over time, management must establish sound procedures to allow for systems changes. These procedures should be well defined and documented, clearly communicated to appropriate employees, and should include an effective monitoring system.

3.2.3 Completeness

Decision makers need complete and pertinent information in a summarized form. Reports should be designed to eliminate clutter and voluminous detail, thereby avoiding “information overload.”

3.2.4 Relevance

Information provided to management must be relevant. Information that is inappropriate, unnecessary, or too detailed for effective decision making has no value. MIS must be appropriate to support the management level using it. The relevance and level of detail provided through MIS systems directly correlate to what is needed by the board of directors, executive management, departmental or area mid-level managers, etc. in the performance of their jobs.

3.3 Achieving Sound MIS

The development of sound MIS is the result of the development and enforcement of a culture of system ownership. An “owner” is a system user who knows current customer and constituent needs and also has budget authority to fund new projects. Building “ownership” promotes pride in institution processes and helps ensure accountability.

Although MIS does not necessarily reduce expenses, the development of meaningful systems, and their proper use, will lessen the probability that erroneous decisions will be made because of inaccurate or untimely information. Erroneous decisions invariably misallocate and/or waste resources.

This may result in an adverse impact on earnings and/or capital. MIS which meets the five elements of usability is a critical ingredient to an institution’s short- and long-range planning efforts. To achieve sound MIS, the

organization's planning process should include consideration of MIS needs at both the tactical and strategic levels. For example, at a tactical level MIS systems and report output should support the annual operating plan and budgetary processes. They should also be used in support of the long term strategic MIS and business planning initiatives. Without the development of an effective MIS, it is more difficult for management to measure and monitor the success of new initiatives and the progress of ongoing projects. Two common examples of this would be the management of mergers and acquisitions or the continuing development and the introduction of new products and services. Management needs to ensure that MIS systems are developed according to a sound methodology that encompasses the following phases:

- Appropriate analysis of system alternatives, approval points as the system is developed or acquired, and task organization.
- Program development and negotiation of contracts with equipment and software vendors.
- Development of user instructions, training, and testing of the system. Installation and maintenance of the system. Management should also consider use of "project management techniques" to monitor progress as the MIS system is being developed. Internal controls must be woven into the processes and periodically reviewed by auditors. Management also should ensure that managers and staff receive initial and ongoing training in MIS.

In addition, user manuals should be available and provide the following information:

- A brief description of the application or system.
- Input instructions, including collection points and times to send updated information.
- Balancing and reconciliation procedures.
- A complete listing of output reports, including samples.

Depending on the size and complexity of its MIS system, an institution may need to use different manuals for different users such as first-level users, unit managers, and programmers.

4.0 CONCLUSION

Vulnerability and risk are the twin worries of system designers and developers concerning their MIS designed systems. System analysts and designers pay considerable attention to the robustness of their systems in order to ensure a greater degree of confidence in the security of their designed systems.

5.0 SUMMARY

In this unit, you have learnt the various elements that are necessary to secure an effective and trustworthy MIS system.

6.0 TUTOR MARKED ASSIGNMENT

- What are the five elements of a useable MIS?
- How do these elements reduce the risk of MIS systems
- List the methodology of a “sound” MIS system
- What is the structure of an MIS user manual?

UNIT 5 INFORMATION SYSTEMS AND DATABASES

CONTENTS

- 1.0 Introduction
- 2.0 Objectives
- 3.0 Main Content
 - 3.1 Overview
 - 3.2 Types of database models
 - 3.3 Data Integrity
 - 3.4 Database Structure
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor Marked Assignment

1.0 INTRODUCTION

Another major thrust of systems design tools is the use of Database Management Systems (DBMS) to develop information systems. In Nigeria, these tools are more commonly used by programmers than other tools such as RAD. In part this is because of the simple information needs of Nigerian organizations, for which a simple DBMS was sufficient. The most famous DBMS used in Nigeria was Aston Tate's DBase III+, which eventually grew up to become Dbase 7.0 for Windows, although it has been discontinued.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the different types of database models
- Explain the database structure

3.0 MAIN CONTENT

3.1 Overview

File systems of varying degrees of sophistication satisfied the need for information storage and processing for several years. However, large enterprises tended to build many independent files containing related and even overlapping data, and data-processing activities frequently required

the linking of data from several files. It was natural, then, to design data structures and database management systems that supported the automatic linkage of files. Three database models were developed to support the linkage of records of different types. These are: (1) the hierarchical model, in which record types are linked in a treelike structure (e.g., employee records might be grouped under a record describing the departments in which employees work); (2) the network model, in which arbitrary linkages of record types may be created (e.g., employee records might be linked on one hand to employees' departments and on the other hand to their supervisors—that is, other employees); and (3) the relational model, in which all data are represented in simple tabular form.

3.2 Types of database models

In the relational model, the description of a particular entity is provided by the set of its attribute values, stored as one row of the table, or relation. This linkage of attribute values to provide a meaningful description of a real-world entity or a relationship among such entities forms a mathematical n -tuple; in database terminology, it is simply called a tuple. The relational approach also supports queries (requests for information) that involve several tables by providing automatic linkage across tables by means of a “join” operation that combines records with identical values of common attributes. Payroll data, for example, could be stored in one table and personnel benefits data in another; complete information on an employee could be obtained by joining the tables on the employee's identification number. To support any of these database structures, a large piece of software known as a database management system (DBMS) is required to handle the storage and retrieval of data (via the file management system, since the data are physically stored as files on magnetic disk) and to provide the user with commands to query and update the database. The relational approach is currently the most popular, as older hierarchical data management systems, such as IMS, the information management system produced by IBM, are being replaced by relational database management systems such as IBM's large mainframe system DB2 or the Oracle Corporation's DBMS, which runs on large servers. Relational DBMS software is also available for workstations and personal computers.

The need for more powerful and flexible data models to support nonbusiness applications (e.g., scientific or engineering applications) has led to extended relational data models in which table entries need not be simple values but can be programs, text, unstructured data in the form of binary large objects (BLOBs), or any other format the user requires. Another development has been the incorporation of the object concept that

has become significant in programming languages. In object-oriented databases, all data are objects. Objects may be linked together by an “is-part-of” relationship to represent larger, composite objects. Data describing a truck, for instance, may be stored as a composite of a particular engine, chassis, drive train, and so forth. Classes of objects may form a hierarchy in which individual objects may inherit properties from objects farther up in the hierarchy. For example, objects of the class “motorized vehicle” all have an engine; members of subclasses such as “bus” or “airplane” will then also have an engine. Furthermore, engines are also data objects, and the engine attribute of a particular vehicle will be a link to a specific engine object. Multimedia databases, in which voice, music, and video are stored along with the traditional textual information, are becoming increasingly important and also are providing an impetus toward viewing data as objects, as are databases of pictorial images such as photographs or maps. The future of database technology is generally perceived to be a merging of the relational and object-oriented views.

3.3 Data integrity

Integrity is a major database issue. In general, integrity refers to maintaining the correctness and consistency of the data. Some integrity checking is made possible by specifying the data type of an item. For example, if an identification number is specified to be nine digits, the DBMS may reject an update attempting to assign a value with more or fewer digits or one including an alphabetic character. Another type of integrity, known as referential integrity, requires that an entity referenced by the data for some other entity must itself exist in the database. For example, if an airline reservation is requested for a particular flight number, then the flight referenced by that number must actually exist. Although one could imagine integrity constraints that limit the values of data items to specified ranges (to prevent the famous “computer errors” of the type in which a N10 cheque is accidentally issued as N10,000), most database management systems do not support such constraints but leave them to the domain of the application program.

Access to a database by multiple simultaneous users requires that the DBMS include a concurrency control mechanism to maintain the consistency of the data in spite of the possibility that a user may interfere with the updates attempted by another user. For example, two travel agents may try to book the last seat on a plane at more or less the same time. Without concurrency control, both may think they have succeeded, while only one booking is actually entered into the database.

A key concept in studying concurrency control and the maintenance of database correctness is the transaction, defined as a sequence of operations on the data that transform the database from one consistent state into another. To illustrate the importance of this concept, consider the simple example of an electronic transfer of funds (say N500 using Western Union Money Transfer facility) from bank account A to account B. The operation that deducts N500 from account A leaves the database inconsistent in that the total over all accounts is N500 short. Similarly, the operation that adds N500 to account B in itself makes the total N500 too much. Combining these two operations, however, yields a valid transaction. The key to maintaining database correctness is therefore to ensure that only complete transactions are applied to the data and that multiple concurrent transactions are executed (under a concurrency control mechanism) in such a way that a serial order can be defined that would produce the same results. A transaction-oriented control mechanism for database access becomes difficult in the case of so-called long transactions—for example, when several engineers are working, perhaps over the course of several days, on a product design that may not reach a consistent state until the project is complete. The best approach to handling long transactions is a current area of database research.

As discussed above, databases may be distributed, in the sense that data reside at different host computers on a network. Distributed data may or may not be replicated, but in any case the concurrency-control problem is magnified. Distributed databases must have a distributed database management system to provide overall control of queries and updates in a manner that ideally does not require that the user know the location of the data. The attainment of the ideal situation, in which various databases fall under the unified control of a distributed DBMS, has been slowed both by technical problems and by such practical problems as heterogeneous hardware and software and database owners who desire local autonomy.

A closely related concept is interoperability, meaning the ability of the user of one member of a group of disparate systems (all having the same functionality) to work with any of the systems of the group with equal ease and via the same interface. In the case of database management systems, interoperability means the ability of users to formulate queries to any one of a group of independent, autonomous database management systems using the same language, to be provided with a unified view of the contents of all the individual databases, to formulate queries that may require fetching data via more than one of the systems, and to be able to update data stored under any member of the group. Many of the problems of distributed databases are the problems of distributed systems in general. Thus distributed

databases may be designed as client-server systems, with middleware easing the heterogeneity problems.

3.4 Database Structure

Databases are structured to facilitate the storage, retrieval, modification, and deletion of data in conjunction with various data-processing operations. Databases can be stored on magnetic disk or tape, optical disk, or some other secondary storage device.

A database consists of a file or a set of files. The information in these files may be broken down into records, each of which consists of one or more fields. Fields are the basic units of data storage, and each field typically contains information pertaining to one aspect or attribute of the entity described by the database. Using keywords and various sorting commands, users can rapidly search, rearrange, group, and select the fields in many records to retrieve or create reports on particular aggregates of data.

Database records and files must be organized to allow retrieval of the information. Early systems were arranged sequentially (i.e., alphabetically, numerically, or chronologically); the development of direct-access storage devices made possible random access to data via indexes. Queries are the main way users retrieve database information. Typically, the user provides a string of characters, and the computer searches the database for a corresponding sequence and provides the source materials in which those characters appear; a user can request, for example, all records in which the contents of the field for a person's last name is the word Shehu.

The many users of a large database must be able to manipulate the information within it quickly at any given time. Moreover, large business and other organizations tend to build up many independent files containing related and even overlapping data, and their data-processing activities often require the linking of data from several files. Several different types of database management systems have been developed to support these requirements: flat, hierarchical, network, relational, and object-oriented.

In flat databases, records are organized according to a simple list of entities; many simple databases for personal computers are flat in structure. The records in **hierarchical databases** are organized in a treelike structure, with each level of records branching off into a set of smaller categories. Unlike hierarchical databases, which provide single links between sets of records at different levels, **network databases** create multiple linkages between sets by placing links, or pointers, to one set of records in another; the speed and

versatility of network databases have led to their wide use in business. **Relational databases** are used where associations among files or records cannot be expressed by links; a simple flat list becomes one row of a table, or “relation,” and multiple relations can be mathematically associated to yield desired information. A popular example of this type of database program among Nigerian software developers is dBase.

Fig. 281 Visual dBase 5.5 Relational Database

Object-oriented databases store and manipulate more complex data structures, called “objects,” which are organized into hierarchical classes that may inherit properties from classes higher in the chain; this database structure is the most flexible and adaptable.

The information in many databases consists of natural-language texts of documents; number-oriented databases primarily contain information such as statistics, tables, financial data, and raw scientific and technical data. Small databases can be maintained on personal-computer systems and may be used by individuals at home. These and larger databases have become increasingly important in business life. Typical commercial applications include airline reservations, production management functions, medical records in hospitals, and legal records of insurance companies.

The largest databases are usually maintained by governmental agencies, business organizations, and universities. These databases may contain texts of such materials as abstracts, reports, legal statutes, wire services, newspapers and journals, encyclopedias, and catalogs of various kinds. Reference databases contain bibliographies or indexes that serve as guides to the location of information in books, periodicals, and other published literature. Thousands of these publicly accessible databases now exist, covering topics ranging from law, medicine, and engineering to news and current events, games, classified advertisements, and instructional courses. Professionals such as scientists, doctors, lawyers, financial analysts, stockbrokers, and researchers of all types increasingly rely on these databases for quick, selective access to large volumes of information.

4.0 CONCLUSION

In this unit, you have learnt about Database Management Systems and how they can be used to create massive access points for vast arrays of data.

5.0 SUMMARY

Despite their simplicity (perhaps because of their earlier reliance on pseudo-code) DBMS systems have gained a wide currency of acceptance in business establishments in Nigerian organizations. Nevertheless, they are powerful means of handling huge arrays of data, and their simplicity makes it possible for clients to modify the code to suit their growing needs.

6.0 TUTOR MARKED ASSIGNMENT

- Explain the different types of database models.
- Explain the database structure.

REFERENCES AND FURTHER READINGS

- A. Dolan, K.A.(1996) Business Computer Systems Design. Mitchell Publishing, Incorporated
- A. Ziya Akta, A.Z. (1986), Structured Analysis and Design of Information Systems. Prentice Hall PTR
- Andriole, S.J. 1990), Getting It Right!: Information System Design Principles for the 90s. A FCEA A International Press. Bronzite, M (2000), System Development: A Strategic Framework. Springer- Verlag New York, Incorporated Brown, D.B., and Jeffrey A. Herbanek, J.A. (1984), Systems Analysis for Applications Software Design. Holden-Day, Incorporated Bruce B. Campbell,
- B.B (1997), Health Management Information Systems in Lower Income Countries: An Analysis of System Design, Implementation and Utilization in Ghana and Nepal. Royal Tropical Institute Press
- Condon, R.J. (1984), Data Processing and System Analysis. Prentice Hall PTR Gibson, M.L. and Cary T. Hughes, C.T. (1994), Systems Analysis and Design: A Comprehensive Methodology with Case. Course Technology, Inc. Head, R.V. (1982), Strategic Planning for Information Systems. Wiley, John & Sons, Incorporated.
- Joseph J. Adamski, J.J., Shelly, GB., and Cashman, T. (1991), Systems Analysis and Design. Course Technology, Inc.
- Lavette, C.Jr., Christopher, T, and Pidgeon, W (1985), Structured Analysis Methods for Computer Information Systems. Pearson Publications Company.
- Leonid A. Kalinichenko, L.A .A. and J . Eder. (Eds), (1996), Advances in Databases and Information Systems: Proceedings of the Second International Workshop on Advances in Databases and Information Systems. Springer-Verlag New York, Incorporated
- Marshall, GR. (1986), Systems Analysis in the Small Systems Environment. Prentice Hall PTR

Skidmore, P (1992), *Introducing Systems Analysis*. Blackwell Publishers

Skidmore, S (1996), *Introducing Systems Design*. Blackwell Publishers

Weinberg, G.M. (2000) *General Principles of Systems Design* Dorset House Publishing

Wetherbe, J.C. (1979), *Cases in Systems Design*. West Publishing Company, College & School Division.

Whitten, J ., Bentley, L., and Dittman, K., *Systems Analysis and Design Methods*, McGraw-HillGraw 5th Edition, 2001.

Few Web-based Resources

- Model Based [system] Architecting and Software Engineering at http://sunset.iisc.edu/research/MBASE/mbase_team/Intro-To-MBASE.doc
- Information Systems at http://www.mislee.com/classes/6201/lecturere_notes/week03-handout-2pdf
- Systems Development Techniques at www.css.edu/depts/cis/Syllabi/07%20syllabus%20fD1.doc
- Systems Analysis and Design: www.ucalgary.ca/~mitchelv/A&D_02_Winter2002.doc
- The Information Systems Project: Applying Systems Analysis and Design Theory and Tools in an Organization at: <http://cbalabs.ou.edu/classfiles/BAD5932/Text/Outline.doc>
- Information System Development: <http://joshgray.myip.org/school/CECS231/Chapter%203%20Notes.doc>
- The Traditional System Life Cycle, at <http://uts.cc.utexas.edu/~mis374/mcleod/ch3.doc>
- Rapid Application Development (RAD), at [.www.dcs.napier.ac.uk/recall/rad/98radlectl.doc](http://www.dcs.napier.ac.uk/recall/rad/98radlectl.doc)

- Requirements Analysis, at www.csd.uwo.ca/courses/CS473b/termproject-phase3.doc
- “Reshaping the Enterprise through an Information Architecture and Process Reengineering.”

CAUSE/EFFECT, Volume 18, Number 4, Winter 1995, pp. 30-31, 36-42