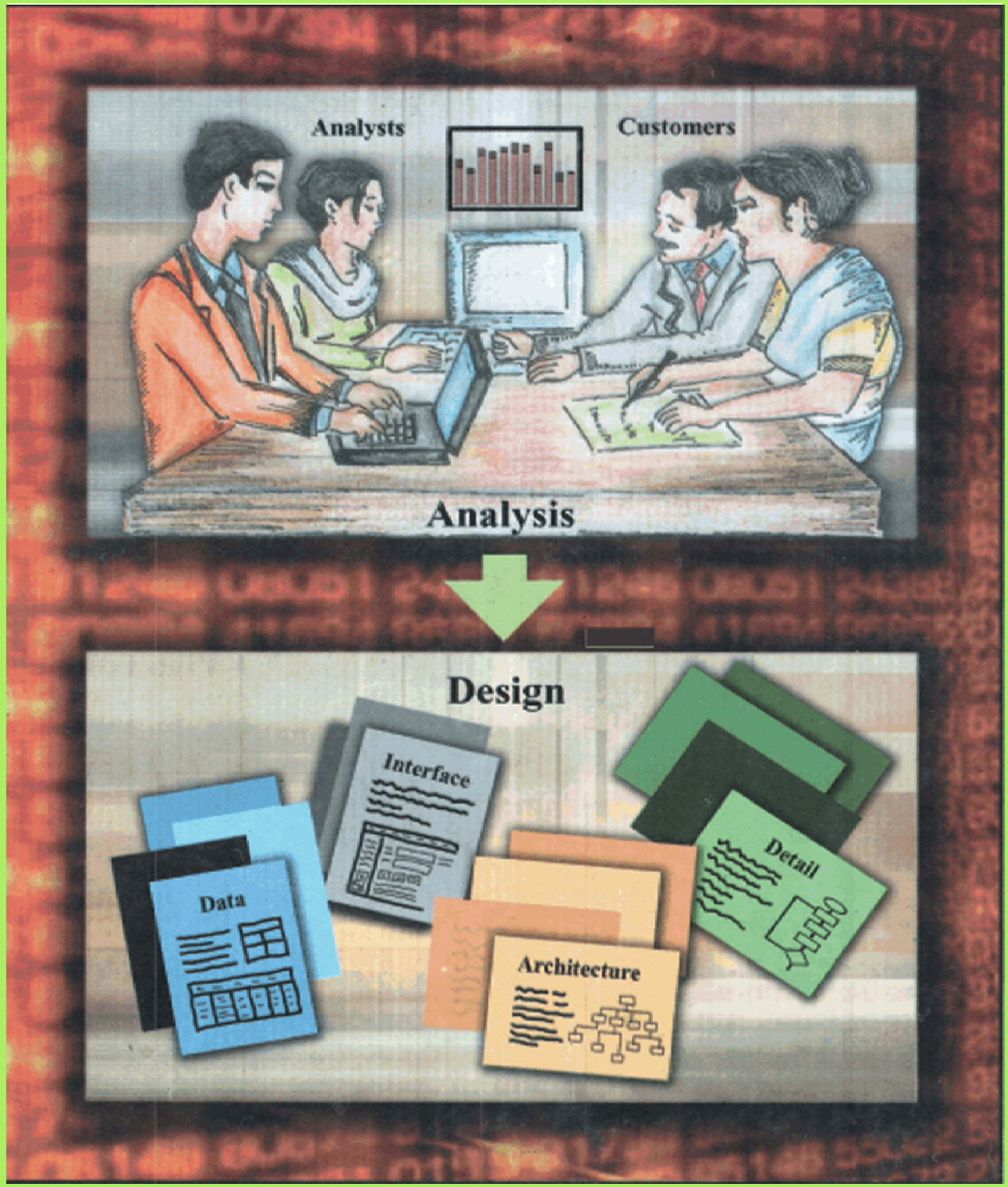
Indira Gandhi National Open University
School of Computer and Information Sciences

MCS-014
SYSTEM ANALYSIS AND DESIGN



INTRODUCTION TO SYSTEMS DEVELOPMENT

1

Indira Gandhi
National Open University
School of Computer and
Information Sciences

MCS-014
SYSTEMS ANALYSIS AND DESIGN

Block

1

INTRODUCTION TO SYSTEMS DEVELOPMENT

Programme / Course Design Committee

Prof. Sanjeev K. Aggarwal, IIT, Kanpur
Prof. M. Balakrishnan, IIT , Delhi
Prof. Harish Karnick, IIT, Kanpur
Prof. C. Pandurangan, IIT, Madras
Dr. Om Vikas, Sr. Director, MIT
Prof P. S. Grover, Sr. Consultant
SOCIS, IGNOU

Faculty of School of Computer and Information Sciences
Shri Shashi Bhushan
Shri Akshay Kumar
Prof Manohar Lal
Shri V.V. Subrahmanyam
Shri P. Venkata Suresh

Block Preparation Team

Prof. M.P.Goel (Content Editor)
Head
Department of Computer Science
Rukmini Devi Institute of Advanced Studies
New Delhi

Ms.Tamanna Siddiqui
Dept. of Computer Science
Jamia Hamdard
New Delhi

Ms.Huma Anwar
Dept. of Computer Science
Institute of Management & Research
Ghaziabad

Shri P. Venkata Suresh
SOCIS, IGNOU

Prof. M.R.Dua
New Delhi

Prof. Sunaina Kumar
SOH, IGNOU

} Language Editors

Course Coordinator : P. Venkata Suresh

Block Production Team

Shri H.K Som, SOCIS

Acknowledgements

To all the faculty of SOCIS, IGNOU for their comments on the course material.

August, 2004

ISBN-81-266-1354-8

Further information on the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110 068.

Printed and published on behalf on the Indira Gandhi National Open University, New Delhi by The Director, SOCIS.

COURSE INTRODUCTION

This course is on the analysis and design of systems.

An organization which did not computerize its activities or an organization which needs to expand its existing systems should have staff who are having knowledge of various systems that are possible along with approaches to development. Systems Analyst plays a key role in the development of the system. He/She develops important specifications and is the life line for the success of the proposed system. There are various ways of developing systems, though most of the phases are same. Appropriate SDLC has to be chosen for the development of a successful system. Documentation is an important part of any project and it should be developed in parallel to the project. There are various standards of documentation. Also, the quality of software is impacted by the quality of the documentation.

When the systems are planned for an organization, as the first step, systems study is done. Requirements are gathered through various techniques such as Interviews, Group Discussions, and Presentations etc. One of the most important steps is the performance of cost benefit analysis. If the system development is not going to lead to any savings in cost or steps are not taken to save the cost , then the development of the system is unnecessary. In this course, we also discuss various principles of design as well as different modelling techniques such as Data flow diagrams and E-R diagrams.

Any system that is developed should be user friendly. The end users should like the use of the system and they should also be able to learn the operation of the system easily. For this reason, normally, any system will consist of forms designed to input data and reports are generated for the assistance of decision makers. Apart from the above, this course also covers the aspects of design of physical files and databases. This course also focuses on CASE tools for the development of systems. Various components of CASE are discussed including the issues related to Visual CASE tools.

After the development of the information system, it is to be implemented. The system should be thoroughly tested for implementation in real life. This course also discusses various threats that are faced by computer systems along with the possible solutions to these problems. Also, the role of management information systems in an organization is discussed along with various categories of information systems that are used in organizations.

This course consists of 4 blocks and is organized in the following manner:

Block-1 covers the fundamentals of Systems Analysis and Design, importance and role of a Systems Analyst , process of systems development and concepts related to documentation of systems.

Block-2 covers the issues related to systems planning, modular and structured design of systems along with the modeling and design of systems.

Block-3 covers the design of forms and reports, design of physical files & databases and usage of CASE tools for the development of systems.

Block-4 covers the issues related to systems implementation and maintenance, audit and security of computer systems and management information systems.

BLOCK INTRODUCTION

This block is on the issues related to the fundamentals of systems development.

When an organization intends to develop systems, the staff that have executive powers should have knowledge of various types of systems that can be developed. Taking stock of the business of the organization as on the date of consideration for systems development is one way and including future possible business activities is an extended way. Sometimes, depending on the types of systems available, an organization may opt for an advanced system keeping the future requirements of the organization in view. This block covers various systems that can be developed such as Real time systems, Distributed systems etc.

Systems Analyst plays a major role in the development of a system for an organization. There might be analysts with in the organization, if it is large enough. In such cases, some of the work, such as preparation of Customer requirement specification, System requirement specification etc. are developed by the organization itself. In case the development work is outsourced, the analyst in the development organization as well as source organization will together develop the required documents. In the case of small an organization which doesn't have their own systems analysts, total work related to development of specifications is handled by the development organization. In this block, we discuss the need of systems analyst to the businesses, his/her role and skills required to be a systems analyst.

There are different phases in the process of systems development. Also, there are different models which sequence these phases in different ways. So, an appropriate model has to be chosen for the development of the system. We discuss various phases as well as products of SDLC phases. Also, various approaches to the systems development are discussed.

Any document that is developed during the development of systems becomes part of documentation. Also, documentation is not a separate phase which is started after/before another phase. It commences with the project. It has to go in parallel. We discuss various standards of documentation and also focus on good practices for documentation.

This block consists of 4 units and is organized as follows:

Unit-1 deals with the fundamentals of the systems. It also introduces Real time systems and Distributed systems to the learner. Also, various approaches for development of information systems are discussed.

Unit-2 deals with the profession of systems analyst. The need for a systems analyst to the businesses, his/her role, and qualifications are covered in this unit.

Unit-3 deals with the process of systems development. Various phases and products of SDLC are discussed along with the approaches to the development. Also, a Case study has been included.

Unit-4 deals with the documentation of systems. This unit covers fundamentals of documentation along with various types and standards available. Also, the issue of quality of software from the context of documentation is discussed.

UNIT 1 INTRODUCTION TO SAD

Structure **Page No.**

1.0 INTRODUCTION

In general, a System is based on Input-Process-Output (IPO model). Manual work can be replaced by computerized system for accuracy and speed of processing. So, before the development of any computerized system, developers should also understand all basic concepts about the system. To develop a system, a standard Methodology must be considered. Different approaches are available for the development of a system. Selecting the best approach is the responsibility of systems analyst and this selection is based on the requirements of end user, problem definition and the infrastructure provided. Standard principles should be followed for the development of good quality software.

1.1 OBJECTIVES

After going through this unit, you should be able to:

- learn the concepts related to Systems;
- know about Real Time Systems;
- know about Distributed Systems; and
- learn the process of developing a successful system.

1.2 FUNDAMENTALS OF SYSTEMS

System is a word derived from the Greek word ‘Systema’ which means an organized relationship among components.

A System may be defined as orderly grouping of interdependent components linked together according to a plan to achieve a specific goal. Each component is a part of total system and it has to do its own share of work for the system to achieve the desired goal.

An **Information system** is an arrangement of people, data, processes, information presentation and information technology that interacts to support and improve day-to-

day operations in a business as well as support the problem solving and decision making needs of management and users.

The characteristics of a System are as follows:

- **Organization** implies structure and order. It is an arrangement of components that helps to achieve objectives.
- **Interaction** refers to the procedure in which each component functions with other components of the system.
- **Interdependence** means that one component of the system depends on another component.
- **Integration** is concerned with how a system is tied together. It is more than sharing a physical part. It means that parts of system work together within the system even though each part performs a unique function.
- **Central Objective** is quite common that an organization may set one objective and operate to achieve another. The important point is that the users must be aware about the central objective well in advance.

1.2.1 Important Terms Related to Systems

Purpose, Boundary, Environment, Inputs, and Outputs are some important terms related to Systems.

- A System's **purpose** is the reason for its existence and the reference point for measuring its success.
- A System's **boundary** defines what is inside the system and what is outside.
- A System **Environment** is everything pertinent to the System that is outside of its boundaries.
- A System's **Inputs** are the physical objects and information that cross the boundary to enter it from its environment.
- A system's **Outputs** are the physical objects and information that go from the system into its environment.

1.2.2 Classification of Systems

Systems may be classified as follows:

a) Formal or Informal
b) Physical or Abstract
c) Open or Closed
d) Manual or Automated.

a) A **Formal System** is one that is planned in advance and is used according to schedule. In this system policies and procedures are documented well in advance. A real life example is to conduct a scheduled meeting at the end of every month in which agenda of the meeting has already been defined well in advance.
An **Informal System** is the system that is not described by procedures. It is not used.

According to a schedule. It works on as need basis. For example, Sales order processing system through telephone calls.

b) **Physical Systems** are tangible entities that may be static or dynamic.
Computer Systems, Vehicles, Buildings etc. are examples of physical systems.
Abstract systems are conceptual entities.

Example: Company

c) **Open System** is a system within its environment. It receives input from environment and provides output to environment**.**
Example: Any real life system, Information System, Organization etc.

Closed System: It is isolated from environment influences. It operates on factors within the System itself. It is also defined as a System that includes a feedback loop, a control element and feedback performance standard.

Figure1.1 shows a Closed loop system. *Performance Standard* is defined as objective that the System has to meet. A *Feedback loop* is defined as a portion of the System that enables the System to regulate itself. Signals are obtained from the System describing the System Status and are transmitted to the Control Mechanism. A *Control Element* compares the output with the performance standard and adjusts the system input accordingly.

Figure 1.1: Closed loop system(refer to Fig 1.1 in unit-1-pg-1.jpg)

d) **Manual and Automated systems**: The system, which does not require human intervention is called Automated system. In this system, the whole process is automatic.

Example: Traffic control system for metropolitan cities.

The system, which requires human intervention, is called a **Manual System**. Example: Face to face information centre at places like Railway stations etc.

1.2.3 Real Life Business Subsystems

A Subsystem is a component of a System, even though it can also be considered as a system in its own right. Consider a manufacturing firm. It consists of five subsystems namely, Product design, Production, Sales, Delivery and Service. .

The boundary is between the firm and its environment. In this system, all the subsystems work together to achieve a goal.

1.3 REAL TIME SYSTEMS

A real time system describes an interactive processing system with severe time limitations. A real time system is used when there are rigid time requirements on the flow of data. A real time System is considered to function correctly only if it returns the correct result within imposed time constraints. There are two types of Real Time systems. They are :

- **Hard Real Time Systems** which guarantee that critical tasks are completed on time.
- **Soft Real Time Systems which** are less restrictive type of real time systems where a critical real time task gets priority over other tasks, and retains the priority until it completes them. Systems that control scientific experiments, medical imaging systems, industrial control systems and some display systems are real time systems.

Check Your Progress 1

Select the appropriate choice given under each question.

1. _____ are comprehensive, multiple-step approaches to system development that will guide your work and influence the quality of your final product.

   a) Techniques
   b) Tools
   c) Methodologies
   d) Data flows.

2. The person in an organization who has the primary responsibility for systems analysis and design is _____.

   a) The end user
   b) The systems analyst
   c) The internal auditor
   d) Business manager.

3. An overall strategy to information systems development that focuses on the ideal organization data, rather than where and how data are used best defines the ____.

   a) Process-oriented approach
   b) Data-organization approach
   c) Data-oriented approach
   d) Information-oriented approach.

4. Which of the following is not a true statement concerning the differences between the process-oriented and data-oriented approaches to systems development?

   a) The process oriented approach has limited design stability
   b) Much uncontrolled data duplication exists with the data oriented approach
   c) The data oriented approach designs data files for the enterprise
   d) None of the above.

5. Non-information system professionals in an organization who specify the business requirements and who use software applications are called:

   a) Programmers
   b) Network managers
   c) Code designers
   d) End users.

6. Which of the following is not one of the four classes of information systems?

   a) Transaction processing systems
   b) Decision support systems
   c) Expert systems
   d) Production systems.

1.4 DISTRIBUTED SYSTEMS

A Distributed System in which the Data, Process, and Interface component of information System are distributed to multiple locations in a computer network. Accordingly, the processing workload required to support these components is also distributed across multiple computers on the network. In this system, each processor has its own local memory. The processors communicate with one another through various communication lines, such as high buses or telephone lines. The processors in a distributed system may vary in size and function. They may include small

microprocessors, workstations, minicomputers, and large general-purpose computer systems. The implementation of a distributed system is complicated and difficult, but still is in demand. Some of the reasons are that modern businesses are already distributed. So, they need distributed solutions. In general, solutions developed using a distributed systems paradigm are user-friendlier. They have the following advantages:

- Resource sharing
- Computation speedup
- Reliability
- Communication.

The five Layers of Distributed System architecture are:

- **Presentation Layer** is the actual user interface. The inputs are received by this layer and the outputs are presented by this layer.
- **Presentation Logic layer** includes processing required to establish user interface. Example: Editing input data, formatting output data.
- **Application Logic Layer** includes all the logic and processing required to support the actual business application and rules
  Example: Calculations.
- **Data Manipulation Layer** includes all the command and logic required to store and retrieve data to and from the database.
- **Data Layer** is actual stored data in the database.

Check Your Progress 2

1. Define the following terms:

   - Information system
   - Strategic information
   - Tactical information
   - Operational information
   - Repository

2. What is the difference between information requirements determination and specification?
   ……
   ……
   ……
   ……
   ……

3. What are the characteristics of a good information system?
   ……
   ……
   ……

1.5 DEVELOPMENT OF A SUCCESSFUL SYSTEM

The success of any system depends on the approach of building it. If the development approach is right, the system will work successfully. Figure 1.2 depicts a System Development Life Cycle. System development life cycle (SDLC) is a standard methodology for the development of Information System. It mainly consists of four phases: System Analysis, System Design, System Construction & Implementation and System Support. Every phase consist of inputs, tasks and outputs. Traditional SDLC was strictly sequential. The developers first complete the previous phase then start the

next phase. But now concept of Repository is introduced in SDLC and it is known as FAST methodology where work is done across shared repository. It means that all the inputs and outputs of phases must be stored in the repository. At any time developers can backtrack to previous phase and they can also work on two phases simultaneously.

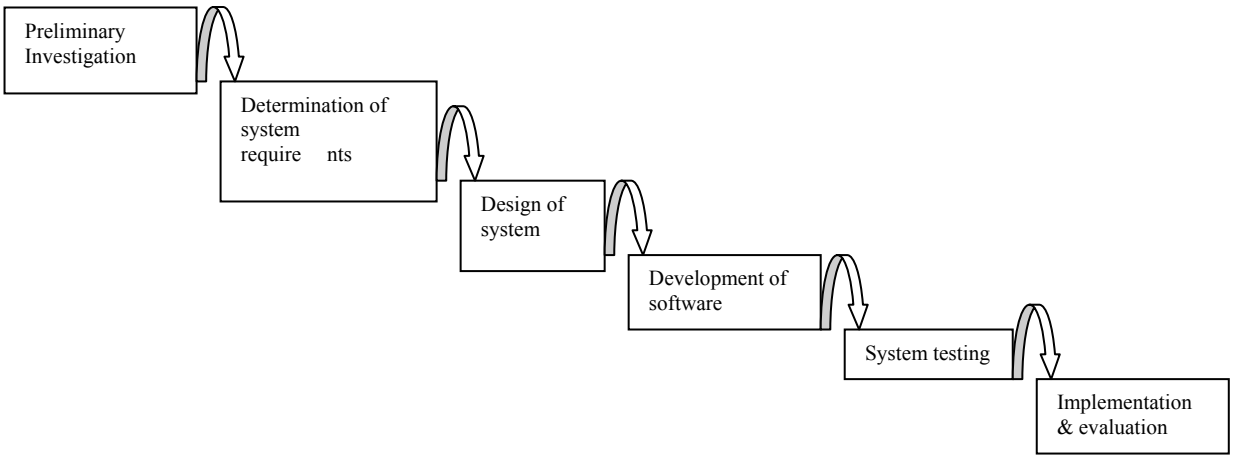


Figure 1.2: System Development Life Cycle

For making a successful system, the following principles should be followed:

(1) Both customers and developers should be involved for accuracy in the information.

(2) A problem solving approach should be adopted. The classic problem solving approach is as follows:

a) Study, understand the problem and its context
b) Define the requirements of a solution
c) Identify candidate solutions and select the best solution
d) Design and implement the solution
e) Observe and evaluate the solution's impact and refine the solution accordingly.

(3) Phases and activities should be established.

(4) For consistent development of a system, some standards should be established.

These standards are:

Documentation standards: It should be an ongoing activity during the system development life cycle.

Quality Standards: Checks should be established at every phase for ensuring that the output of every phase meets the business and technology expectations.

Automated Tool standards: Hardware and software platforms should be finalized for the development of Information system. Automated tool standards prescribe technology that will be used to develop and maintain information systems and to ensure consistency, completeness, and quality.

(5) Development of information system should be considered as capital investment: The developer of an information system should think about several solutions of a particular problem and every solution should be evaluated for cost-effectiveness and risk management. *Cost-effectiveness* is defined as the result obtained by striking a balance between the cost of developing and operating an

information system and the benefits derived from that system. Risk management is defined as the process of identifying, evaluating and controlling what might go wrong in a project before it becomes a threat to the successful completion of the project or implementation of the information system.

Multiple feasibility checkpoints should be built into system development methodology. At each feasibility checkpoint, all costs are considered sunk (i.e. not recoverable). Thus, the project should be re- evaluated at each checkpoint to determine if it remains feasible to continue investing time, effort, and resources. At each checkpoint, the developers should consider the following options:

- Cancel the project if it is no longer feasible.
- Re-evaluates and adjusts the cost and schedule if project scope is to be increased.
- Reduce the scope if the project budget and schedule are frozen and not sufficient to cover all the project objectives.

(6) *Divide and Conquer* approach is the way of making a complex problem easier. In this approach, the larger problem (System) is divided into smaller problems (Subsystem).

(7) For development of a successful system, the system should be designed for growth and change. When the System is implemented, it enters the operations and support stage of Life Cycle (Please refer to figure 1.3).

Figure 1.3: System Maintenance (refer to Fig 1.4 in unit-1-pg2.jpg)

During this stage, the developers encounter the need for changes that range from correcting simple mistakes to redesigning the system to accommodate changing technology to making modifications to support changing user requirements. These changes direct the developers to rework formerly completed phases of the life cycle.

1.6 VARIOUS APPROACHES FOR DEVELOPMENT OF INFORMATION SYSTEMS

Various approaches are available for development of Information Systems. They are:

- **Model Driven**: It emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.
- **Accelerated approach**: A prototyping approach emphasizes the construction of model of a system. Designing and building a scaled-down but functional version of the desired system is known as Prototyping. A prototype is a working system that is developed to test ideas and assumptions about the new system. It consists

of working software that accepts input, perform calculations, produces printed or display information or perform other meaningful activities.

- **Joint Application Development**: It is defined as a structured approach in which users, managers, and analysts work together for several days in a series of intensive meetings to specify or review system requirements. In this approach, requirements are identified and design details are finalized.

1.6.1 Structured Analysis and Design Approach

The goal of structured system analysis and design is to reduce maintenance time and effort. Modeling is the act of drawing one or more graphical representations of a System. Model driven development techniques emphasize the drawing of models to help visualize and analyze problems, define business requirements and design Information systems. The first model driven approach is Structured Analysis and Design approach.

Structured Analysis is a development method for the analysis of existing manual systems or automated systems, leading to development of specifications (expected functionality or behaviour) for proposed system. The objective of structured analysis approach is to organize the tasks associated with requirement determination to provide an accurate and complete understanding of a current situation. The major tasks of structured system analysis approach are:

- Preliminary Investigation
- Problem Analysis
- Requirement Analysis
- Decision Analysis.

It is a process-centred technique that is used to model business requirements for a system. Structured analysis introduced a process-modeling tool called the *Data flow diagram*, used to illustrate business process requirements. With the help of DFD, the systems analyst can show the system overview. Data modeling tools such as *Entity relationship diagrams* are used to illustrate business data requirements. With the help of ERD, the analyst, can show database overview.

Structured Design utilizes graphic description (Output of system analysis) and focuses on development of software specifications. The goal of structured design is to lead to development of programs consisting of functionally independent modules that perform relatively independently of one another. It is a specific program design technique, not a comprehensive design method. Thus it does not specify file or database design, input or output layout or the hardware on which the application will run. It provides specification of program modules that are functionally independent.

It is a process-centred technique that transforms the structured analysis models into good software design models. Structured Design introduced a modeling tool called Structure Charts. They are used to illustrate software (program) structure to fulfil business requirements. Structure charts describe the interaction between independent module and the data passing between the modules. These module specifications can be passed to programmers prior to the writing of program code. In structure chart the whole application is divided into modules (set of program instructions) and modules are designed according to some principles of design. These principles are:

Modularity and partitioning: Each system should consist of a hierarchy of modules. Lower level modules are generally smaller in scope and size compared to higher level modules. They serve to partition processes into separate functions.

Coupling: Modules should be loosely coupled. It means that modules should have little dependence on other modules in a system.

Cohesion: Modules should be highly cohesive. It means that modules should carry out a single processing function.

Span of control: Modules should interact with and manage the functions of a limited number of lower level modules. It means that the number of called modules should be limited (in a calling module).

Size of Module: The number of instructions contained in a module should be limited so that module size is generally small.

Shared use of Functions: Functions should not be duplicated in separate modules may be shared. It means that functions can be written in a single module and it can be invoked by any other module when needed.

1.6.2 Prototype

A prototyping approach emphasizes the construction model of a system. Designing and building a scaled-down but functional version of a desired system is the process known as Prototyping. A prototype is a working system that is developed to test ideas and assumptions about the new system. It consists of working software that accepts input, performs calculations, produces printed or displayed information or performs other meaningful activities. It is the first version or iteration of an information system i.e. an original model. Customer evaluates this model. This can be effectively done only if the data are real and the situations are live. Changes are expected as the system is used. This approach is useful when the requirements are not well defined. A prototype is usually a test model. It is an interactive process. It may begin with only new functions and be expanded to include others that are identified later. The steps of Prototyping process are depicted in Figure 1.4. They are:

- Identify the user's known information requirements and features needed in the system.
- Develop a working prototype.
- Revise the prototype based on feedback received from customer
- Repeat these steps as needed to achieve a satisfactory system.

Actual development of a working prototype is the responsibility of a systems analyst. The difference between a prototype model and an actual information system is that, a prototype will not include the error checking, input data validation, security and processing completeness of a finished application. It will not offer user help as in the final system.

But, sometimes, the prototype can evolve into the product to be built. The prototype can be easily developed with tools of fourth generation languages (4GL's) and with the help of Computer Aided Software Engineering (CASE) tools. Prototyping approach is a form of rapid application development (RAD).

Figure 1.4: Prototype approach (refer to Fig 1.5 in unit-1-pg2.jpg)

1.6.3 Joint Application Development

It is defined as a structured approach in which users, managers, and analysts work together for several days in a series of intensive meetings to specify or review system requirements. The important feature of JAD is joint requirements planning, which is a process whereby highly structured group meetings are conducted to analyze problems and define requirements.

The typical participants in a JAD are listed below:

JAD session leader: The JAD leader organizes and runs the JAD. This person is trained in group management and facilitation as well as system analysis. The JAD leader sets the agenda and sees that it is met. The JAD leader remains neutral on issues and does not contribute ideas or opinions but rather concentrates on keeping the group on the agenda, resolving conflicts and disagreements, and soliciting all ideas.

(1) **Users:** The key users of the system under consideration are vital participants in a JAD. They are the only ones who have a clear understanding of what it means to use the system on a daily basis.

(2) **Managers:** The role of managers during JAD is to approve project objectives, establish project priorities, approve schedules and costs and approve identified training needs and implementation plans.

(3) **Sponsors:** A JAD must be sponsored by someone at a relatively high level in the company i.e. the person from top management. If the sponsor attends any session, it is usually at the very beginning or at the end.

(4) **Systems Analysts:** Members of the systems analysis team attend the JAD session although their actual participation may be limited. Analysts are there to learn from customers and managers, but not to run or dominate the process.

(5) **Scribe:** The scribe takes down the notes during the JAD sessions. This is usually done on a personal computer or a laptop. Notes may be taken using a word processor. Diagrams may directly be entered into a CASE tool.

(6) **IS staff** like systems analysts, other IS staff such as programmers, database analysts, IS planners and data centre personnel may attend to learn from the discussions and possibly to contribute their ideas on the technical feasibility of proposed ideas or on technical limitations of current systems.

The following are the various benefits of Joint Application Development:

- actively involves users and management in project development,
- reduces the amount of time required to develop a system, and
- incorporates prototyping as a means for confirming requirements and obtaining design approvals.

Check Your Progress 3

1. List the fundamental principles of S/W Development Life Cycle.

……

……

……

……

……

1.7 SUMMARY

In this unit, all the basic concepts that are necessary to understand the system, types and characteristics are given. Concepts about real and distributed systems are also discussed. Development of a successful information system depends on principles of SDLC. Different approaches are available for development of information systems. These are Model Driven (structure analysis and design approach), Accelerated (prototype) and JAD approach. Selection of the approach is based on the end user requirements, problem identified and infrastructure provided.

1.8 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. c
2. b
3. c
4. b
5. d
6. d

Check Your Progress 2

1. **Information system:** It is an arrangement of people, data, processes, information presentation and information technology that interacts to support and improve day-to-day operations of a business as well as support the problem solving and decision making needs of management and users.

   Strategic Information: This is the information needed for long range planning and top-level management of any organization that uses it. This information is unstructured or semi-structured and the volume of this information in any organization is very small.

   Tactical Information: This type of information is needed to take short term decisions to run the business efficiently and middle level managers' use it. The volume of tactical information is more than the volume of strategic information.

   Operational Information: This type of information is used for day-to-day operations of a business organization and first level management in the organization uses it. It is usually easy to obtain by straightforward clerical processing of data. The volume of this information is much more than volume of tactical information.

   Repository: It is defined as data store of accumulated system knowledge i.e. system models, detailed specifications, and any other documentation that has been accumulated during the system's development. This knowledge is reusable and critical to the production system's ongoing support. The repository is implemented with various automated tools and it is often centralized as an enterprise business and IT resource.

2. Information requirement determination attempts to find the strategic, tactical and operational information that is needed to effectively manage an organization. Information specification defines the manner in which the information will be presented and the analyzed data, it consists of.

3. The following are characteristics of a good information system:

   - meeting customer's requirements,
   - modular design, and
   - easily maintainable.

Check Your Progress 3

1. The following are the fundamental principles of SDLC:

   - Management and users should be involved because they can explain the problem that is to be taken for design and development in accurate manner.
   - A problem solving approach should be adopted
   - Phases and activities should be established.
   - Some standards should be established for consistent development of System.
   - Development of Information System should be considered as capital Investment.
   - Divide and Conquer approach should be adopted. It is one way of making complex problem easier.
   - For making a system successful, it should be designed for growth and change.

1.9 FURTHER READINGS

- Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman; *System analysis and design methods*; Tata McGraw-Hill ;Fifth Edition;2001.
- By Jeffrey A. Hoffer , Joey F. George , Joseph S. Valacich; *Modern Systems Analysis and Design*; Pearson Education; Third Edition;2002.

Reference Web sites

http://www.rspa.com

UNIT 2 SYSTEMS ANALYST – A PROFESSION

Structure **Page No.**

2.0 INTRODUCTION

The work of a systems analyst who designs an information system is the same as an architect of a house. Three groups of people are involved in developing information systems for organizations. They are managers, users of the systems and computer programmers who implement systems. The systems analyst coordinates the efforts of all these groups to effectively develop and operate computer based information systems.

Systems analysts develop information systems. For this task, they must know about concepts of systems. They must be involved in all the phases of system development life cycle i.e. from preliminary investigation to implementation. Success of development depends on skills and the dedication of Systems analysts.
Analysing, designing and implementing systems to suit organizational needs are the functions of systems analyst. S/he plays a major role in evaluating business benefits from computer technology. Systems analyst is basically a problem solver with unique skills. A systems analyst deals with people, procedures and technologies.

2.1 OBJECTIVES

After going through this unit, you should be able to :

- know the need of systems analyst in the business and in the development of information system;
- know the job responsibilities of systems analysts in the traditional business and in the modern business;
- know the role of systems analyst in a team for the benefit of the organization and success of developed information system;
- know different analytical skills that are important to systems development, including problem identification, problem solving and systems thinking;
- know the need for basic technical skills even when systems are developed using rapid prototyping and code generators;

- know the way systems analysts use their skills in managing resources, projects, risk, and change; and
- know the importance of interpersonal skills for a systems analyst, in communicating, working with teams, facilitating groups, and managing exceptions.

2.2 WHY DO BUSINESSES NEED SYSTEMS ANALYSTS?

A computerized system enables an organization to provide accurate information and respond faster to the queries, events etc. If a business needs computerized information system, a Systems Analyst is required for analysis and design of that system. Information systems evolved from the need to improve the use of computer resources for the information processing needs of business application. Customer defines the business problems to be solved by the computer. Project managers, Analysts, Programmers and Customers apply information technology to build information systems that solve those problems. Information technology offers the opportunity to collect and store enormous volume of data, process business transactions with great speed and accuracy and provide timely and relevant information for taking correct decision by management. This potential could not be realized without the help of a systems analyst since business users may not fully understand the capabilities and limitations of modern information technology. Similarly, computer programmers and information technologists do not fully understand the business applications they are trying to computerize or support. A communication gap has always existed between those who need computer based business solutions and those who understand information technology. Systems analyst bridges this gap.

2.3 USERS

System users are defined as the people who use information systems or who are affected by the information system on a regular basis i.e. capturing, validating, entering, responding to, storing and exchanging data and information apart from others. A common synonym is client. System users are concerned with business requirements. There are two main classes of system users and they are discussed below.

- **Internal Users** are employees of the business for which an information system is built. Examples are clerical and service staff, technical and professional staff, supervisors, middle level managers and executive managers. Remote and mobile users are the new class of internal users. They are geographically separated from the business. Examples of mobile users are sales and service representatives. An example of remote user is the person who is associated with telecommunication i.e. working from home. The person can be connected to the company's information system through modern communications technology.

- **External Users:** Modern information systems are now reaching beyond the boundaries of the traditional business to include customers and other businesses as system users. In business-to-business information systems, each business becomes an external user of the other business's information systems. For example, in the case of direct purchasing of product through the Internet, customer becomes an external user of the retailer's order processing information systems. Another example is that if a business connect their purchasing systems directly to the order processing systems of their suppliers then both become external users to each other.

2.4 ANALYSTS IN VARIOUS FUNCTIONAL AREAS

Today the systems analyst's job presents a fascinating and exciting challenge. It offers high management visibility and opportunities for important decision-making and creativity that may affect an entire organization.

2.4.1 Systems Analyst in Traditional Business

In the traditional business, information services are centralized for the entire organization or for a specific location. In this organization, the staff of information services (refer to Figure 2.1) report directly to the chief executive officer (CEO). The highest-ranking officer is sometimes called a chief information officer (CIO) and the rest of information services are organized according to the following functions or areas:

- **System Development** : In traditional business, systems analysts and programmers are organized into permanent teams that support the information systems and applications for specific business function. System development unit includes a *centre for excellence*, which is a group of experts (experienced systems analysts, system designers, and system builders) who establish and enforce methods, tools, techniques and quality for all system development projects.

- **Data Administration**: Data and other Information Resources of the organization are managed. This includes databases that are used by system developers to support applications. Systems analysts who are experts in data analysis can work here. These analysts are known as *Data Analysts*. They analyse database requirements, design and construct (sometimes) the corresponding databases.

- **Telecommunications**: Here, computer networks that play a critical role in the success of any business are designed, implemented and managed. Here, Network *analysts* perform many of the tasks as applied to designing local and wide area networks that will ultimately be used by systems and applications.

- **End-user Computing**: The growing base of personal computers and local area networks in the end user community are supported. This provides installation services, training and helps desk services. Analyst also provides standards and consulting to end users that develop their own systems with PC power tools such as spreadsheets and PC database management systems. In this centre, analysts may work as *End-user computing consultants.*

- **Computer Operations**: All of the shared computers including mainframes, minicomputers and other computers are put to operation and the same is coordinated. Systems Analysts may work as *Capacity Analysts* in this area.

Every analyst should know the management structure of a traditional information services organization.

2.4.2 Systems Analyst in Modern Business

Many medium-to-large information services units for the modern business have reorganized to be decentralized with a focus on empowerment and dynamic teams. In modern business, systems analyst may be reassigned to different projects at time to time. During the project, the systems analyst and other team members are directly accountable to the business unit for which the system is being developed. In this type of organization, the information services try to get closer to users and management to

improve services and value. Today's analysts should also know about a modern information services organization.

In modern business, two new trends are used for software development: outsourcing and consulting. *Outsourcing* is the act of contracting an outside vendor to assume responsibility for one or more IT functions or services. *Consulting* is the act of contracting with an outside vendor to assume responsibility for or participate in one or more IT projects.

Check Your Progress 1

1. What are the differences between problem identification and problem solving?
   ……
   ……
   ……
2. What is the difference between a logical system description and a physical system description?
   ……
   ……
   ……
3. Why is the development of information systems, sometimes done by an independent consultant?
   ……
   ……
   ……
4. Differentiate between systems analyst and a business analyst.
   ……
   ……
   ……

2.5 ROLE OF A SYSTEMS ANALYST

The success of an information system development is based on the role of Systems analyst. Among several roles, some important roles are described below:

- **Change Agent:** The analyst may be viewed as an agent of change. A candidate system is designed to introduce change and reorientation in how the user organization handles information or makes decisions. Then, it is important that the user accepts change. For user acceptance, analysts prefer user participations during design and implementation. Analyst carefully plans, monitors and implements change into the user domain because people inherently resist changes. In the role of a change agent, Systems Analyst may use different approaches to introduce changes to the user organization.
- **Investigator and Monitor:** A systems analyst may investigate the existing system to find the reasons for it's failure. The role of an investigator is to extract the problems from existing systems and create information structures that uncover previously unknown trends that may have a direct impact on organization. The role of a Monitor is to undertake and successfully complete a project. In this role, analysts must monitor programs in relation to time, cost and quality.

- **Architect** The analyst's role as an architect is liaison between the user's logical design requirements and the detailed physical system design. As architect the analyst also creates a detailed physical design of candidate systems. A systems analyst makes the design of information system architecture on the basis of end user requirements. This design becomes the blue print for the programmers.

- **Psychologist:** In system development, systems are built around people. The analyst plays the role of psychologist in the way s/he reaches people, interprets their thoughts, assesses their behaviour and draws conclusions from these interactions. Psychologist plays a major role during the phase of fact finding.

- **Motivator:** System acceptance is achieved through user participation in its development, effective user training and proper motivation to use the system. The analyst's role as a motivator becomes obvious during the first few weeks after implementation and during times when turnover results in new people being trained to work with the candidate system.

- **Intermediary:** In implementing a candidate system, the analyst tries to appease all parties involved. Diplomacy in dealing with people can improve acceptance of the system. The analyst's goal is to have the support of all the users. S/he represents their thinking and tries to achieve their goals through computerization.

These multiple roles require analysts to be orderly, approach a problem in a logical way, and pay attention to details. They prefer to concentrate on objective data, seek the best method, and be highly prescriptive. They appear to be cool and studious. They focus on method and plan, point out details, are good at model building, perform best in structured situations, and seek stability and order.

2.6 DUTIES OF A SYSTEMS ANALYSTS

The duty of a systems analyst is to coordinate the efforts of all groups to effectively develop and operate computer based information systems. The **duties** of a systems analyst are following:

- **Defining Requirements:** The most important and difficult duty of an analyst is to understand the user's requirements. Several fact-finding techniques are used like interview, questionnaire, and observation, etc.

- **Prioritising Requirements by Consensus:** There is a need to set priority among the requirements of various users. This can be achieved by having a common meeting with all the users and arriving at a consensus. This duty of systems analyst requires good interpersonal relations and diplomacy. S/he must be able to convince all the users about the priority of requirements.

- **Analysis and Evaluation:** A systems analyst analyses the working of the current information system in the organization and finds out the extent to which they meet user's needs. On the basis of facts and opinions, systems analyst finds the best characteristics of the new or modified system which will meet the user's stated information needs.

- **Solving Problems:** Systems analyst is basically a problem solver. An analyst must study the problem in depth and suggest alternate solutions to management. Problem solving approach usually incorporates the following general steps:

    - Identify the problem
    - Analyse and understand the problem
    - Identify alternative solutions and select the best solution.

- **Drawing up Functional Specifications:** The key duty of systems analyst is to obtain the functional specifications of the system to be designed. The specification must be non-technical so that users and managers understand it. The specification must be precise and detailed so that it can be used by system implementers.
- **Designing Systems:** Once the specifications are accepted, the analyst designs the system. The design must be understandable to the system implementer. The design must be modular to accommodate changes easily. An analyst must know the latest design tools to assist implementer in his task. An Analyst must also create a system test plan.
- **Evaluating Systems:** An analyst must critically evaluate a system after it has been in use for a reasonable period of time. The time at which evaluation is to be done, how it is to be done and how user's comments are to be gathered and used, must be decided by the analyst.

Check Your Progress 2

1. Are excellent programmers necessarily excellent systems analysts? Justify your answer.
   ……
   ……
   ……
2. List at least eight tasks performed by systems analysts.
   ……
   ……
   ……
3. List at least six attributes of a systems analyst.
   ……
   ……
   ……
   ……
4. Why should a systems analyst be able to communicate well?
   ……
   ……
   ……
   ……

2.7 QUALIFICATIONS OF A SYSTEMS ANALYST

A systems analyst must fulfil the following requirements:

- Working knowledge of information technology
- Computer programming experience and expertise
- General business knowledge
- Problem solving skills
- Communication skills
- Interpersonal skills
- Flexibility and adaptability
- Thorough knowledge of analysis and design methodologies.

In summary, the skills that are required may be classified into the following:

- Analytical skills
- Technical skills
- Management skills
- Interpersonal skills.

2.7.1 Analytical Skills

As the designation of person is Systems Analyst, possession of analytical skills is very important. Analytical skills can be classified into the following sets:

- System study
- Organizational knowledge
- Problem identification
- Problem analysis and problem solving.

System Study: The first important skill of systems analyst is to know about system. It means that Systems Analyst should be able to identify work assignment as a system. It involves identification of each of the system's characteristics such as inputs, outputs, processes etc. Information systems can be seen as subsystems in larger organizational systems, taking input and returning output to their organizational environments.

Data flow diagram clearly illustrates inputs, outputs, system boundaries, the environment, subsystems and inter-relationship. Purpose and constraints are much more difficult to illustrate and must therefore be documented using other notations. In total, all elements of logical system description must address all characteristics of a system.

Organizational Knowledge: Whether a person is an in-house (in traditional organization) or contract software developer (in modern organization), s/he must understand how organization works. In addition s/he must understand the functions and procedures of the particular organization (or enterprise) s/he is working for. Selected areas of organizational knowledge for a systems analyst are given below:

(1) How work officially gets done in a particular organization: In this area, knowledge about the following is required:

- Terminology, abbreviations and acronyms
- Policies
- Standards and procedures
- Formal organization structure
- Job description.

(2) Understanding the organization's internal politics: In this area, knowledge is required about the following:

- Influence and inclinations of key personnel
- Finding the experts in different concerned subject areas
- Critical events in the organization's history
- Informal organization structure
- Coalition membership and power structures.

(3) Understanding the organization's competitive and regulatory environment: In this area, knowledge is required about the following:

- Government regulations
- Competitors from domestic and international fronts

- Products, services and markets
- Role of technology.

(4) Understanding the organization's strategies and tactics: In this area, the requisite knowledge is given below:

- Short as well as long term strategy and plans
- Values and missions.

Problem Identification: A problem can be defined as the difference between an existing situation and a desired situation. The process of identifying problem is the process of defining differences. So, problem solving is the process of finding a way to reduce differences. A manager defines differences by comparing the current situation to the output of a model that predicts what the output should be. In order to identify problems that need to be solved, the systems analyst must develop a repertoire of models to define the differences between what is present and what ought to be present.

Problem Analysis and Problem Solving: Once a problem has been identified, systems analyst must analyse the problem and determine how to solve it. Analysis entails more about the problem. Systems analyst learns through experience, with guidance from proven methods, the process of obtaining information from concerned people as well as from organizational files and documents. As s/he seeks out additional information, s/he also begins to formulate alternative solutions to the problem. The next step is that the alternatives are compared and typically one is chosen as best solution. Once the analyst, users and management agree on the general suitability of a solution (feasibility), they devise a plan for implementing it.

Herbert Simon has first proposed this approach. According to her/him, this approach has four phases namely intelligence, design, choice and implementation. This approach is similar to system development life cycle.

Intelligence: During this phase, all information relevant to the problem is collected.
Design: During this phase, alternatives are formulated.
Choice: During this phase, the best alternative solution is chosen.
Implementation: During this phase, the solution is put into practice.

2.7.2 Technical Skills

Many aspects of the job of systems analyst are technically oriented. In order to develop computer based information systems, systems analyst must understand information technologies, their potentials and their limitations. A systems analyst needs technical skills not only to perform tasks assigned to him/her but also to communicate with the other people with whom s/he works in systems development. The technical knowledge of a Systems Analyst must be updated from time to time.

In general, a Systems Analyst should be as familiar as possible with such families of technologies such as:

- Microcomputers, workstations, minicomputers, and mainframe computers,
- Programming languages,
- Operating systems, both for PC's and networks,
- Database and File management systems,
- Data communication standards and software for local and wide area networks,
- System development tools and environments (such as forms & report generators and graphical user interface design tools), and
- Decision support systems and data analysis tools.

S/he should know all of the above as well as modern methods and techniques for describing, modeling and building systems.

2.7.3 Management Skills

When a systems analyst is asked to lead a project team then management skills are required. Systems analyst needs to know the process of managing his/her own work and how to use organizational resources in the most productive ways possible. Self-management is important skill for an analyst. There are four categories of management skills:

- Resource management
- Project management
- Risk management
- Change management.

Resource Management: A systems analyst must know how to get the most out of a wide range of resources i.e. system documentation, information technology and money. A team leader must learn how to best utilize the particular talents of other team members. S/he must also be able to delegate responsibility, empower people to do the tasks they have been assigned.

Resource management includes the following capabilities:

- Predicting resource usage (budgeting)
- Tracking and accounting for resource consumption
- Learning how to use resources effectively
- Evaluating the quality of resources used
- Securing resources from abusive use
- Relinquishing resources when no longer needed and releasing the resources when they can no longer be useful.

Project Management: A *project* is defined as a sequence of unique, complex and connected activities having one goal or purpose and that must be completed by a specific time, within budgets and according to specifications.

Project management is defined as the process of scoping, planning, staffing, organizing, directing and controlling the development of acceptable system at minimum cost within a specified time frame. In the role of project manager, s/he first needs to decompose a project in to several independent tasks. The next step is to determine how the tasks are related to each other and who will be responsible for each task.

Risk Management: A risk is any unfavourable event or circumstance that can occur while a project is underway. If a risk comes true, it can hamper the successful and timely completion of a project. Therefore, it is necessary to anticipate and identify different risks, a project is susceptible to, so that contingency plans can be prepared in advance to control the effects of each risk. Once, risk to the project has been identified, project manager must be able to minimize the likelihood that those risks will actually occur. It also includes knowing where to place resources (such as people) where they can do the best and prioritising activities to achieve better productivity.

Change Management: Introducing a new or improved information system into an organization is a change process. In general people do not like change and tend to resist it. Therefore, any change in the way people perform their duties in an organization must be carefully managed. Change management is a very important skill for systems analyst. The systems analyst must know how to get people to make a smooth transition from one information system to another, giving up their old ways of

doing things and accepting new ways. Change management also includes the ability to deal with technical issues related to change, such as obsolescence and reusability.

2.7.4 Interpersonal Skills

Systems analyst works extensively with staff in key positions in an organization. So, interpersonal skills are necessary for success of him/her. These skills can be classified as:

- Communication skills
- Working alone as well as in a team
- Facilitating groups
- Managing expectations.

Communication skills: A Systems analyst should be able to communicate clearly and effectively with others. S/he must establish a good relationship with clients early in the project and maintain it throughout the project. Communication takes many forms from written to verbal to visual. The analyst must be able to master as many forms of communication as possible. Interpersonal communication subjects are:

- Business speaking
- Business writing
- Interviewing
- Listening
- Technical discussion
- Technical writing.

Working alone as well as in a team: A Systems analyst must be able to organize and manage his/her own schedule, commitments and deadlines because many people in the organization will depend on his/her individual performance, but systems analyst must work with the team towards achieving project goals. To work together effectively and to ensure the quality of the product, the team must establish standards of cooperation and coordination that guide their work. There are 12 characteristics of a high performance team that influence team work:

- Shared and elevated vision
- Sense of team identity: Result-driven structure
- Competent team members
- Commitment to the team
- Mutual trust
- Interdependency among team members
- Effective communication
- Sense of autonomy
- Sense of empowerment
- Small team size.

Facilitating groups: This skill is required when systems analyst works in Joint application development approach. In this approach systems analyst works with group during system development. Analysts use JAD sessions to gather systems requirements and to conduct design reviews. Systems analyst can be asked to work as a facilitator. Facilitation necessarily involves a certain amount of neutrality on the part of the facilitator. The facilitator must guide the group without being a part of the group and must work to keep the effort on track by helping the group resolve differences. Guidelines for a facilitator are given below:

- Purpose should be made clear
- Make sure that the group understands what is expected of them and of you
- Use physical movement to focus on yourself or on the group
- Reward group member participation with thanks and respect

- Ask questions instead of making statement
- Wait patiently for answers
- Be a good listener
- Encourage group members to feel ownership of the group's goal and of their attempts to reach those goals.

Managing expectations: System development is a change process, and members of any organization greet any organizational change with anticipation and uncertainty. Organizational members will have certain ideas about what new information system will be able to do for them. Ginzberg found that successfully managing user expectations is related to successful systems implementation. The systems analyst needs to understand the technology. S/he must understand the work flows that the technology will support and how the new system will affect them. The important ability of systems analyst is to communicate a realistic picture of the new system and what it will do for users and managers. Managing expectations begins with the development of the business case for the system and extends all the way through training people to use the finished system.

The relationship between a systems analyst's skills and systems development life cycle are depicted in figure 2.1.

Figure 2.1: SDLC and Skills of Systems Analyst (refer to Fig 2.3 in unit-2.jpg)

Check Your Progress 3

1. What are the business and technology trends that affect the players in the field of information systems?

   ……

   ……

   ……

2.8 SUMMARY

In this unit, we discussed the skills necessary for success as a systems analyst. An organization needs systems analyst for the replacement of existing system with computerized system. A systems analyst may work on a project basis or may be part of client's team as a permanent employee who works about changes to be implemented to the existing system in the client organization. A systems analyst takes various roles to work in a team for the benefit of the organization and to develop successful information systems. Some of the roles are: Change Agent, Investigator and Monitor, Architect, Psychologist, Motivator, Intermediary.

The requisite skills for systems analyst are analytical, technical, management and interpersonal.

2.9 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. In problem identification, a systems analyst compares the current situation in an organization to the desired situation. Problem identification involves measurement, not decision making. Problem solving is the process of finding one or more ways to reduce these differences and then select the best approach for implementation.

2. A *logical system description* portrays the purpose and function of the system without tying the description to any specific physical implementation. A *physical system description* of a system focuses on how the system will be materially constructed.

3. Time constraints, limited number of internal personnel and need for a better expertise are reasons for hiring an outside consultant.

4. A systems analyst facilitates most of the activities to develop or acquire an information system. S/he studies the problems and needs of an organization to determine how people, data, processes, communication and information technology can best accomplish improvement of the business.

   A business analyst is a systems analyst who specializes in business problem analyses and technology independent requirements analysis.

Check Your Progress 2

1. An excellent programmer is not necessarily an excellent systems analyst. A programmer is given clear specification (often in writing) and designs efficient and maintainable programs. S/he need not have good communication skills and inter-personal relations. S/he need not have knowledge of functionality of organizations.
   A programmer works with clear specifications whereas an analyst has to arrive at clear specifications from fuzzily stated requirements.

2. Tasks performed by systems analysts are:

   - Define requirements,
   - Draws up ordering of requirements,
   - Gather data, facts and opinion of users,
   - Analyses the existing systems in the organization and uses this knowledge to improve systems,

- Provides solutions to problems posed by management,
- Makes specification of information systems,
- Designs the information system, and
- Evaluates the information system.

3. The desirable attributes of a systems analyst:

   - Must know how organizations function,
   - Must know latest developments in computer hardware and software,
   - Can get along with diverse people from top level managers to the last level of employees,
   - Must be able to express himself and absorb information by being a good listener,
   - Must have an analytical mind, and
   - Must have a broad general knowledge.

4. S/he has to understand the users' requirements mostly by interviewing them and thus s/he has to ask the right questions, listen carefully and summarize the gist of conversation. S/he should also be able to present and explain orally to the users, the system designed by her/him and clarify doubts they may have after the oral presentation. His main job is to interact with the management, users' and the programmers. So, it is obvious that s/he must posses' good communication skills.

Check Your Progress 3

1. Players in the field of information systems are being affected by a number of business and technology trends including the following:

   - Total quality management, a comprehensive approach to facilitating quality improvements and management within a business.
   - Business process redesign, the study, analysis, and redesign of fundamental business processes to reduce costs and/or improve value added to the business.
   - Continuous process improvement, the continuous monitoring of business processes to reduce costs and add value.
   - Enterprise resource planning, the selection and implementation of a single vendor's fully integrated information system that spans most basic business functions required by a major corporation.
   - Electronic commerce, a new way of doing business that involves the conduct of both internal and external business over the internet, intranets and extranets.

2.10 FURTHER READINGS

- Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman; *System analysis and design methods*; Tata McGraw-Hill; Fifth Edition; 2001.

- By Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich; *Modern Systems Analysis and Design*; Pearson Education; Third Edition; 2002.

Reference Websites

- http://www.freeforessays.com
- http://www.rspa.com

UNIT 3 PROCESS OF SYSTEM DEVELOPMENT

Structure **Page No.**

3.0 INTRODUCTION

Information Systems Analysis and Design is a complex and stimulating process that is used to develop and maintain computer based information systems. The analysis and design of information systems are driven from an organizational point of view. An organization might consist of whole enterprise, specific departments or individual work groups. Information Systems Analysis and Design is, therefore, an organizational improvement process. Systems are built and rebuilt (enhanced) for organizational benefits. Benefits result by adding value during the process of creating, producing and supporting the organization's services and products. Thus, Information Systems Analysis and Design is based on the understanding of objectives, structure and processes of organization and the knowledge about the application of Information Technology for this purpose. Most organizations find it beneficial to use standard sets of steps, called a systems development methodology, to develop and support their information systems (IS). Like many processes, the development of Information Systems often follows a life cycle called Systems Development Life Cycle. For example, a product follows a life cycle when it is created, tested and introduced in the market. Its sale increases and goes to peak point and after that it declines and a new product or next version of the existing product is introduced in the market to replace it. SDLC is a common methodology for systems development in many organizations, consisting of various phases that mark the progress of system analysis and design effort.

3.1 OBJECTIVES

After going through this unit, you would be able to:

- define Information Systems Analysis and Design;
- describe the information systems development life cycle; and

- list alternatives to SDLC and compare the advantages and disadvantages of SDLC to its alternatives.

3.2 SYSTEMS DEVELOPMENT LIFE CYCLE

Most organizations find it beneficial to use a set of steps, called a systems development methodology, to develop and support their information system. Like many processes, the development of information system often follows a life cycle. The system development life cycle (SDLC) is a common methodology for system development in many organizations, featuring various phases that mark the progress of the system analysis and design effort.

Although any life cycle appears at first glance to be a sequentially ordered set of phases but actually it is not. The specific steps and their sequence are meant to be adapted as required for a project, consistent with management approach. For example, in any given SDLC phase, the project can return to an earlier phase, if necessary. If a commercial product does not perform well just after its introduction, it may be temporarily removed from the market and improved before being re-introduced. In the system development life cycle, it is also possible to complete some activities in one phase in parallel with some other activities of another phase. Sometimes, life cycle is iterative; that is, phases are repeated as required until a satisfactory and acceptable system is found. Such an iterative approach is special characteristic of rapid application development methods, such as prototyping. Some people consider life cycle to be spiral, in which we constantly cycle through the phases at different levels of detail. The life cycle can also be thought of a circular process in which the end of the useful life of one system leads to the beginning of another project that will develop a new version or replace an existing system altogether. However, the system development life cycle used in an organization is an orderly set of activities conducted and planned for each development project? The skills of a system analyst are required to be applied to the entire life cycle.

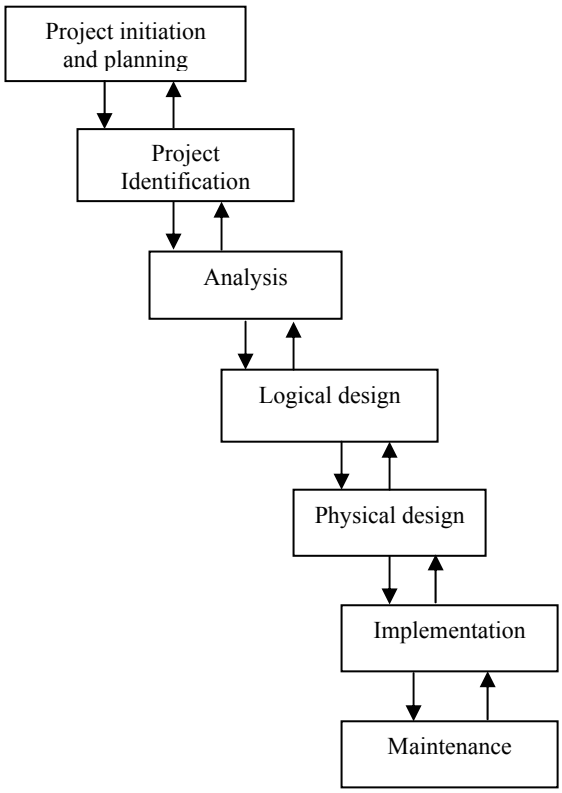


Figure: 3.1: Phases of System Development Life Cycle

Every custom software producer will have its own specific detailed life cycle or system development methodology. Even if a particular methodology does not look like cycle, you will discover that many of SDLC steps are performed, SDLC techniques and tools are used.

In order to make this unit generic, we follow a rather general life cycle model, as described in figure 3.1.

This model resembles a staircase with arrows connecting each step to the step before and to the step after it. This representation of the system development life cycle (SDLC) is sometimes referred to as the "waterfall model". We use this SDLC as one example of methodology but more as a way to arrange the steps of systems analysis and design. Each phase has specific outputs and deliverables that feed important information to other phases. At the end of each phase, system development project reaches a milestone and, as deliverables are produced, parties outside the project team often review them.

3.3 PHASES OF SDLC

SDLC consists of mainly seven steps. These are:

1. Project Identification and Selection
2. Project Initiation and Planning
3. Analysis
4. Logical Design
5. Physical Design
6. Implementation
7. Testing.

3.3.1 Project Identification and Selection

The first phase in the SDLC is called project identification and selection. In this phase, the user identifies the need for a new or improved system. In large organizations, this identification may be part of a systems planning process. Information requirements of the organization as a whole are examined, and projects to meet these requirements are proactively identified. The organization's information system requirements may result from requests to deal with problem in current system's procedures, from the desire to perform additional tasks, or from the realization that information technology could be used to capitalize on an existing opportunity. These needs can then be prioritised and translated into a plan for the Information System department including a schedule for developing new major systems. In smaller organizations, determination of which systems to develop may be affected by user request submitted as the need for new or enhanced systems arises as well as from a formal information planning process. In either case, during project identification and selection, an organization determines whether or not resources should be devoted to the development or enhancement of each information system under consideration. The outcome of the project identification and selection process is a determination of which systems development projects should be undertaken by the organization at least in terms of an initial study.

3.3.2 Project Initiation and Planning

The second phase is project initiation and planning. The problems that are identified should be investigated and a decision to implement the information system or not for the organization should be taken. A critical step at this point is determining the scope of the proposed system. The project leader and initial team of system analysts also produce a specific plan for the proposed project, which the team will follow using the

remaining SDLC steps. Now, this baseline project plan customizes the standardized SDLC and specifies the time and resources needed for its execution.

The formal definition of a project is based on the likelihood that the organization's information system department is able to develop a system that will solve the problem or use the opportunity and determine whether the costs of developing the system outweigh the benefits it could provide. The final presentation with the subsequent project phases is usually made by the project leader and other team members to someone in management or to a special management committee with the job of deciding which projects the organization will undertake.

3.3.3 Analysis

Analysis is the next phase. During this phase, the analysis has several sub-phases. The first is requirements determination. In this sub-phase, analysts work with users to determine the expectations of users from the proposed system. This sub-phase usually involves a careful study of current systems, manual or computerized that might be replaced or enhanced as part of this project. Next, the requirements are studied and structured in accordance with their inter-relationships and eliminate any redundancies. Third, alternative initial design is generated to match the requirements. Then, these alternatives are compared to determine which alternative best meets the requirement in terms of cost and labour to commit to development process.

In this phase, feasibility study of the proposed system is also performed. Various types of feasibilities are:

- Technical feasibility
- Economic feasibility
- Behavioural feasibility
- Operational feasibility
- Legal feasibility
- Time feasibility.

If the proposed system is not feasible to develop, it is rejected at this very step. The output of the analysis phase is a description of (but not are detailed design for) the alternative solution recommended by the analysis team. Once, the recommendation is accepted by those with funding authority, you can begin to make plans to acquire any hardware and system software necessary to build or operate the system proposed.

System Design: After analysis phase is complete, design of the system begins. The design consists of logical and physical design of the system. The fourth and fifth phases are devoted to design of the new and enhanced system. During design, you and the other analysts convert the description of the recommended alternative solution into logical and then physical system specifications. You must design all aspects of the system from input and output screens to reports, databases, and computer processes. Design occurs in two phases, viz., logical design and physical design.

3.3.4 Logical Design

Logical design is not tied to any specific hardware and systems software platform. Theoretically, the system could be implemented on any hardware and systems software. The idea is to make sure that the system functions as intended. Logical design concentrates on the business aspects of the system.

3.3.5 Physical Design

In physical design, the logical design is turned into physical or technical specifications. For example, you must convert diagrams that map the origin, flow, and

processing of data in a system into a structured systems design that can then be broken down into smaller and smaller units known as modules for conversion to instruction written in a programming language. You design various parts of the system to perform the physical operations necessary to facilitate data capture, processing, and information output. During the physical design, the analyst team decides the programming language in which the computer instructions will be written in, which database system and file structure will be used for the data, the platform that will be used and the network environment under which the system will be run. These decisions finalize the hardware and software plans initiated at the end of the analysis phase. Now, proceedings can be made with respect to acquisition of any new technology not already present in the organization. The final product of the design phase is the physical system specification in a form ready to be turned over to programmers and other system builders for construction. The physical system specifications are turned over to programmers as the first part of the implementation phase.

Check Your Progress 1

1. What is the difference between Project Identification and Project Initiation?
   ……
   ……
   ……
2. What is the difference between analysis and design?
   ……
   ……
   ……
3. Why do most of the projects die after feasibility phase?
   ……
   ……
   ……

3.3.6 Implementation

During implementation, you turn system specification into working system that is tested and put into use. Implementation includes **coding, testing and installation**. During coding, programmers write programs that make up the system. During testing, programmers and analysts tests the individual programs and the entire system in order to find and correct errors. During installation, the new system becomes a part of the daily activities of the organization. Application is installed or loaded, on existing or new hardware and users are introduced to new system and trained. The analysts begin planning for testing and installation as early as the project initiation and planning phase, since testing and installation require extensive analysis in order to develop the right approach.

Installation of the system can be done in the following three ways:

- **Direct conversion:** In this type of conversion, the software is directly installed at user's site.
- **Parallel conversion:** In this type of conversion, both the old and new systems are run in parallel for some time. After monitoring the new system for a reasonable period of time and if it is performing well, then, the new system is implemented replacing the old one.

- **Phased conversion:** In this type of conversion, the system is installed module by module.

Implementation activities also include initial user support such as the finalization of *documentation, training programs, and ongoing user assistance*. Note that documentation and training programs are finalized during implementation. Documentation is produced throughout the lifecycle. Implementation can continue for as long as the system exists since ongoing user support is also part of implementation. Despite the best efforts of analysts, managers, and programmers, however, installation is not always a simple process. Many well-designed systems can fail if implementation is not well managed. The management of implementation is usually done by the project team.

3.3.7 Maintenance

The final phase is maintenance. When a system is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions. Also, the organization's requirements with respect to the system change with time. During maintenance, programmers make the changes that users ask for and modify the system to reflect and support changing business conditions. These changes are necessary to keep the system running and useful. Maintenance is not separate phase but a repetition of the other lifecycle phases required to study and implement the needed changes. Thus, maintenance is an overlay to the life cycle rather than a separate phase. The amount of time and effort devoted to maintenance depends a great deal on the performance of the previous phase of life cycle. There comes a time, however, when an information system is no longer performing as desired, when maintenance cost becomes prohibitive, or when the organization's needs has changed substantially. Such problems are an indication that it is the time to begin designing the system's replacement, therefore, completing the loop and starting the life cycle over again. Often, the distinction between the major maintenance and new development is not clear, which is another reason why maintenance often resembles the lifecycle itself.

Maintenance is of three types:

Corrective maintenance: In this type, the errors that creep into the system are removed. Hence the name *corrective maintenance*.

Adaptive maintenance: It is done to adapt with the changing external factors. For example, if the government rules change regarding the Dearness Allowance from 52% to 58%, then the changes have to be made in the Information System to adapt with the changing scenario**.**

Perfective maintenance: This is done to satisfy the users' requirements to make the system more and more perfect.

The SDLC is a highly linked set of phases where output of one phase serves as input to the subsequent phase. Throughout the systems development life cycle, the systems development project needs to be carefully planned and managed. Therefore, the larger the project, the greater is the need for project management.

3.4 PRODUCTS OF SDLC PHASES

- *Project Identification and Selection*: Priorities for systems and project, architecture for data, networks, hardware and Information System Management are the result of the associated system.

- *Project Initiation and Planning*: Detailed work plan for project, specification of system scope and high level system requirements, assignment of team members and other resources.
- *Analysis*: Description of current system, need to enhance or replace current system, explanation of alternative systems and justification of alternatives.
- *Logical Design:* Functional and detailed specification of all system elements (data, process, input and output).
- *Physical design*: Technical, detailed specifications of all system elements, i.e., programs, files, network, system software, etc. and acquisition plan for new technology.
- *Implementation*: Code, documentation, training programs and support capabilities.
- *Maintenance*: New version of software with associated updates of documents, training and support.

3.5 APPROACHES TO DEVELOPMENT

In the continuing effort to improve the systems analysis and design process, several approaches have been developed. Attempts to make system development less of an art and more of a science usually referred to as engineering techniques, are applied to system development. We will discuss prototyping, followed by introduction to joint application design and participatory design.

3.5.1 Prototyping

Designing and building a scaled down but fundamental version of a desired system is known as prototyping. A prototype can be built with any computer language or development tool to simplify the process. A prototype can be developed with some fourth generation languages (4GLs) such as query, screen and report design tools of a data base management system (DBMS), and with tools called computer aided software engineering (CASE) tools.

Using prototyping as a development technique, the analyst works with user to determine the initial or basic requirements of the system. The analyst then builds a prototype. When the prototype is completed, the user works with it and tells the analyst what they like and do not like about it .The analyst uses this feedback to improve the prototype and take the new version back to the user. This process is iterated until the users are satisfied. Two key advantages of the prototyping technique are the large extent to which proto typing involves the user in analysis and design and its ability to capture requirements in concrete rather than verbal or abstract form. In addition to being used stand-alone, prototyping can also be used to augment the SDLC.

Prototyping is a form of rapid application development or RAD. The fundamental principle of any RAD methodology is to delay producing detailed system design document until the user requirements are clear. The prototype serves as the working description of needs. RAD methodologies emphasize gaining user acceptance of the human system interface and developing core capabilities as quickly as possible.

3.5.2 Joint Application Design

In the late 1970s, systems development personnel at IBM developed a new process for collecting information system requirements and reviewing systems designs. The process is called Joint Application Design (JAD). The basic idea behind JAD is to bring structure to the requirements determination phase of analysis and to the reviews that occur as part of design. Users, managers, and system developers are brought together for a series of intensive structured meetings run by a JAD session leader who

maintains the structure and sticks to the agenda. By gathering the people directly affected by an Information System in one room at the same time to work together to agree on system requirements and design details, time and organizational resources are better put to use. An added advantage is that, group members are more likely to develop a shared understanding of what the information system is supposed to do.

3.5.3 Participatory Design (PD)

Participatory Design (PD) represents a useful alternative approach to the SDLCPD emphasizes the role of the user much more than other techniques do. In some cases, PD may involve the entire user community in the development process. Each user has an equal share in determining system requirements and in approving system design. In other cases, an elected group of users control the process. These users represent the larger community. Under PD, systems analysts work for the users. The organization's management and outside consultants provide advice rather than control. PD is partly a result of the role of labour and management in the workplace where labour is more organized and is more intimately involved with technological changes.

Check Your Progress 2

1. Designing and building a scaled down but fundamental version of a desired system is the process known as _____

2. Prototyping is a form of _____

3. Implementation includes _____

3.6 CASE STUDY

The problem is to computerize Library of XYZ College. In this library, all transactions are handled manually. Registers are maintained to record the details of the books, information about the members of the library and to manipulate the issue and return of books. The data entry and recovery procedures are all manual and this takes a lot of time and energy to browse through the pages of the register for locating the relevant information.

This current manual system of the library is very tough and time-consuming and chances of getting errors gets very high. This method is not trustworthy.
This problem can be solved in the following steps:

Project Initiation and Planning

The specific services provided by our project will, of course, differ from other projects. Understanding the reasons behind the development of this project gives an appreciation of what our project does.

The main objective behind this project is:

- To provide the user with an easy and fast interface.
- To see that information handling is very easy and fast.
- Easy updation and modification of data.
- The basic aim of the project is to automate the basic functions of the library:
    - To handle the Book Details.
    - To record and handle the Member Details.
    - To handle issue, return of books and to keep details about the books given for reading.

Analysis

Is it feasible to automate the system? The three major areas to determine the feasibility of project are given below:

- **Technical Feasibility:** The current level of technology can support the proposed system. The proposed software is enabled to meet all the objectives of the system and the output received would be more efficient. So, the project is technically feasible.

- **Economic Feasibility:** The proposed system needs to get hardware and software installed. The short-term costs are overshadowed by the long-term gains. The management in question can invest in the system and is in condition to pay for the cost of system's study, cost of employee's time involved in the study and the cost of development of software. Thus, project is economically feasible.

- **Operational Feasibility**: The current system faces a lot of problems which would be removed in the proposed system. The employees of the system will be free from the burden of the paper work and a lot of confusion. The employees are themselves interested in getting the manual system replaced by the automated one. The proposed system is user friendly. So, even a layman can use it. Thus, it is operationally feasible.

Design

Once it is found that the project development is feasible, Design has to be developed for the requirements listed in the analysis phase.

Data Dictionary

- A Data Dictionary is a catalogue of all elements in a system. It consists of data about data.
- It is a document that collects co-ordinates and confirms what specific data terms mean to different people in the team.
- It is important for the following reasons:
    - to manage the details,
    - communicate meaning,
    - document system features,
    - facilitate analysis, and
    - locate errors and omissions.

Consider a Library Information System. Our Data Dictionary record stores the following descriptions:

- ***Book_Details***

Stores information about books in the library. The table contains the following attributes:

Attributes	Stores
Book Id (primary key)	ISBN Number
Name	Name of the book.
Category	Category of the book.
Subject	Subject of the book.
Author	Author of the book.
Price	Price of the book.

Date	Date on which the book is added in the library.
Edition	Edition of the book.
Copies	Total no. of copies of the book present in the library.

- ***Book_Copy***

Stores information about various copies of a book available in the library.

Attributes	Stores
Book Id	ISBN Number.
Book Idg	Indent no. of the book whose multiple copies are present.

- ***Book_State***

Stores information regarding current status of the book.

Attributes	Stores
Book Id (primary key)	ISBN Number.
Status	It gives information about current status of the book. Status of a book can be –
	• I => Book is issued.
	• L => Book is lost.
	• P => Book is present.
	• M => Book is lost by member.
	• D => Book is deleted.

- ***Book_IR***

It gives information about the state of the book which is issued to a member.

Attributes	Stores
TID (primary key)	Identification no of issue / return transaction.
Book ID	ISBN Number.
MemID	Stores indent of member who issued the book.
Mode	Mode can be –
	• I => book is issued for home.
	• R => book is issued for reading in library.
State	State can be –
	• 0 => book is with the member.
	• 1 => member returned the book.
	• 2 => member lost the book.

- ***Book_Irdetail***

It gives information about issue/return and date/time of the issue/return of the book.

Attributes	Stores
TID	Identification number of the transaction of issue/return.
Issue_Date	Stores the date on which the book is issued.
Issue_Time	Stores the time of issue of the book to the member.
Return_Date	Stores the date on which a member returns the book.
Return_Time	Stores the time at which a member returns the book.

- ***Book_Delete***

It stores the information about the books that are deleted and the date on which it was deleted.

Attributes	Stores
BookId	ISBN Number.
Date	Stores the date on which the book was deleted.

- ***Book_Renewed***

It stores the information about the book, when it is brought back to the library and the date on which it was renewed.

Attributes	Stores
BookId	Stores the bookId of the book being deleted.
Date	Stores the date on which the book was resumed.

- ***Member_Details*** – It stores the information about members of the library

Attributes	Stores
MemId	Stores the ID of the member.
First	Stores the first name of the member.
Middle	Stores the middle name of the member.
Last	Stores the last name of the member.
Sex	Stores the gender of the member.
Address	Stores the address of the member.
City	Stores the city of the member.
State	Stores the state of the member.
Country	Stores the country of the member.
PIN	Stores the pin code of the member.
Age	Stores the age of the member.
Phone	Stores the phone number of the member.
Issue_limit	Stores the maximum number of books that can be issued to the member.
Date_of_Joining	Stores the date on which member enrols in the library.
Books_Issued	Stores the number of books issued to the member.

- ***Member_State***

Storing information about state of the member in the library.

Attributes	Stores
MemID	Stores the ID of the member.
State	It can have two values. • P => Member can access the library. • D => Member is deleted for library access.

- ***Member_Deleted***

It stores information about the member deleted and the date on which the member was deleted.

Attributes	Stores
MemID	Stores identification no of the member being deleted.
Date	Stores date on which member was deleted.

The Zero level DFD is depicted in Figure 3.2.

Figure 3.2: 0- level DFD of Library Information System (refer to Fig. 3.2 in unit-3.jpg)

Input Design

The input design of this project is as follows. Points considered for the design of 'easy to fill out' form are given below which conforms to the design of the project:

- Designing form with proper flow.
- Logical grouping of information.
- Labels holding suitable captions & textboxes to accept the data.
- Usage of other tools, such as radio buttons, checkboxes, combo boxes, etc. also serve purpose for the better recording, processing, storing and retrieval of information.
- The appearance of the form has been tried to be kept as attractive as possible to help in better and logical organization of details.
- Since we know good screen design like good form design is an important instrument for steering the course of work, our design of input is guided by the following six objectives:
    - Effectiveness
    - Accuracy
    - Ease to use
    - Consistency
    - Simplicity
    - Attractiveness.
- Our screens show only that data which is necessary for the particular action being undertaken.
- Screens are kept consistent by locating information in the same area each time a new screen is accessed.
- We have made it easy to move from one screen to another through the use of icons, which channels the way to other screens apart from direct access to screens through the main menu.
- Rather than jamming all data into one screen and cluttering up the screen, we have made use of multiple screens which add to the user appeal, thus are more productive and are prone to less errors.

Data Capture Information

- **Identification of data**

The identifying data item in each transaction record is called a "KEY". Therefore, details of member is identified by the unique code, the details of the books through a unique book code. Similarly, the issue / return transactions through the transaction code.

- **Details of the retrieval system**

This in with reference to the stored data that can be quickly retrieved from the system files. This is done when we perform search on a particular criterion to draw the records or details of the search parameters.

- **Output Design**

Users generally merit the system by its output. Thus, in order to create the most useful output, system analyst works closely with the user through the interactive process, until the result is considered to be satisfactory.

The objectives of the output design are:

- Serve the intended purpose.
- Output should satisfy the user.
- Assured output where it is needed.
- Output on time.
- Choose appropriate output methods.
    - Reports
    - Messages (on screen)
    - Document on help

Depending on the circumstances and the contents, the output may be displayed or printed. Output contents may originate from these sources:

- Retrieval from data stores.
- Transmission from a process or system activity.
- Directly From input source.

Keeping the above points in mind, we have taken best care to present our information with the most clear and readable output. Our details are convincing enough to make the decisions fast and accurate.

Our reports represent one feature of output to present the various details in discrete categories. These reports can be viewed on screen as well as can be kept as a hardcopy in the printed layouts. Our system produces following reports:

1. List of books
2. List of members
3. List of books issued
4. List of books returned.

- **Database Design**

The following are various entities along with attributes for the project:

- ***Book_Details*** – (BookId, Name, Author, Category, Subject, Price, Date, Copies, Edition).
- ***Book_Copy*** – (BookId, BookIDC).
- ***Book_IR*** – (TID, BookId, MemId, Mode, State).
- ***Book_TRDetail*** – (TID, Issue_Date, Issue_Time, Return_Date, Return_Time).
- ***Book_Delete*** – (BookId, Date).
- ***Book_Resumed*** – (Book_Id, Date).
- ***Book_State*** – (BookId, Status).

- ***Member_Details*** – (MemID, First, Middle, Last, Sex, Address, City, State, Country, Pin, Age, Phone, Issue Limit, Date_of_Joining, Books_Issued).
- ***Member_State*** – (MemID, State).
- ***Member_Delete*** – (MemID, Date).
- ***Member_Resumed*** – (MemID, Date).

After these steps, coding in any programming languages can be done and then the system will be tested against the requirements of the user. The tested system will be implemented either by direct conversion or by parallel conversion.

Check Your Progress 3

1. What are the phases involved in the analysis and development of the system?
   ……
   ……
   ……
2. What are the basic types of feasibilities used in system analysis and design?
   ……
   ……
   ……
3. "Implementation of system can be done in three different ways". Do you agree with this statement? Justify your answer.
   ……
   ……
   ……
4. What are various types of maintenance.
   ……
   ……
   ……

3.7 SUMMARY

In this unit, you learned about the basic framework that guides systems analysis and design, the systems development life cycle, with its seven major phases: project identification and selection, project initiation and planning, analysis, logical design, physical design, implementation, and maintenance. The life cycle has had its share of criticism, which you read about, and other frameworks have been developed to address the life cycle's problems. These alternative frameworks include: Prototyping (Rapid Application Development approach), Joint Application Design and Participatory Design.

3.8 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. The first phase in the SDLC is called project identification. In this phase, the user identifies the need for a new or improved system. Information requirements of the organization as a whole are examined, and projects to meet these requirements are proactively identified. In project initiation, the formal, but

preliminary investigation of the system problem or opportunity at hand and the presentation of reasons as of why the system should or should not be developed by the organization is done. A critical step at this point is determining the scope of the proposed system.

2. During analysis phase, the requirements are determined. In this phase, analysts work with users to determine what the users want from a proposed system. This phase usually involves a careful study of any existing systems, manual or computerized that might be replaced or enhanced as part of the project. Next, the requirements are studied and structured according to their inter-relationship and eliminate any redundancies. After this alternative initial design is generated to match the requirements then these alternatives are compared to determine which alternative best meets the requirements.

   After analysis phase is complete, design of the system begins. The design consists of logical and physical design of the system. During design, the analysis converts the description of the recommended requirements into logical and then physical system specifications. Design occurs in two phases, viz., logical design and physical design.
   Logical design concentrates on the business aspects of the system.
   In physical design, the logical design is turned into physical or technical specifications.

3. If the proposed system is not feasible to develop, it is rejected at this very step. In this phase, feasibility study of the proposed system is performed. The Cost Benefit Analysis of the proposed system is prepared by the system analyst in this phase and if the cost of the proposed system in terms of time, money and resources outweigh the benefits of the system, then the proposed system is rejected.

Check Your Progress 2

1. Prototyping
2. Rapid Application Design
3. Coding, testing and installation

Check Your Progress 3

1. The following phases are involved in the analysis and development of the system:

   - Project identification and selection
   - Project initiation and planning
   - Analysis
   - Logical design
   - Physical design
   - Implementation
   - Maintenance.

2. The six basic types of feasibilities used in system analysis and design are:

   - Technical feasibility
   - Economic feasibility
   - Behavioural feasibility
   - Operational feasibility
   - Legal feasibility
   - Time feasibility.

3. Yes. The implementation of system can be done in three different ways. They are:

- Direct conversion
- Parallel conversion
- Phased conversion.

4. Types of maintenance are:

   - Corrective maintenance
   - Adaptive maintenance
   - Perfective maintenance

3.9 FURTHER READINGS

- Kendall & Kendall; *Systems Analysis and Design*; PHI; Fifth Edition.
- Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman; *System analysis and design methods*; Tata McGraw-Hill; Fifth Edition; 2001.

Reference Websites

- http://www.rspa.com
- http://www.ieee.org

UNIT 4 INTRODUCTION TO DOCUMENTATION OF SYSTEMS

Structure **Page No.**

4.0 INTRODUCTION

Documentation is a process to help users of software and other people to use and interact with system. Effective and accurate documentation is very necessary for the success of any system. Documentation becomes part of each step of system development through out the process of system development even before the documentation starts officially. During the process of system development, study reports and other necessary information are documented to help people involved in system development to understand the process. There are many kinds of documentation namely analysis documentation, design documentation, interface documentation, internal program documentation and user-oriented documentation. Documentation should not be seen as a overhead for system development. Rather, documentation has to be seen as an investment for the development of high quality software.

4.1 OBJECTIVES

After going through this unit, you should be able to:

- understand the concept of documentation;
- understand the importance of documentation;
- learn about various documents and process of documentation;
- understand application of various standards of documentation processes;
- differentiate between various documentation processes;
- know relation between the documentation and quality of software; and
- understand the good practices for documentation process.

4.2 CONCEPTS AND PROCESS OF DOCUMENTATION

Documentation may be defined as the process of communicating about the system. The person who is responsible for this communication is called documenter. It may be

noted that documenter is not responsible for the accuracy of the information, and his job is just to communicate or transfer the information.

The ISO standard ISO/IEC 12207:1995 describes documentation "as a supporting activity to record information produced by a system development life cycle process."

Why documentation?

Documentation is needed because it is

- a means for transfer of knowledge and details about description of the system
- to communicate among different teams of the software project;
- to help corporate audits and other requirements of the organization;
- to meet regulatory demand;
- needed for IT infrastructure management and maintenance; and
- needed for migration to a new software platform.

Document communicates the details about the system targeted at different audience. It explains the system. The ever-increasing complexity of information system requires emphasis on well-established system of documentation. Every information system should be delivered along with an accurate and understandable document to those who will use the software.

Traditionally, documentation was done after the development of the software is completed. However, as the software development process is becoming complex and involved, documentation has become an integral part of each system development process. Documentation is now carried out at every stage as a part of development process. We will also discuss how documentation affects quality of the software later in this section.

When the process of documentation is undertaken as a separate process, it requires planning in its own right. The figure below shows, how at the development process, documentation is done alongside each step. Design and development activities of software depend on a certain base document. Documentation is to be carried out before actually implementing the design. In such a case, any flaw in design identified can be changed in the document thereby saving cost and time during implementation. If documentation is being developed for an existing software, then documentation is done along side the software development process.

The Process of Documentation

The following are various steps involved in the process of documentation:

Collection of source material: The very first step of any documentation process is to acquire the required source material for preparation of document. The material is collected including specifications, formats, screen layouts and report layouts. A copy of the operational software is helpful for preparing the documentation for user.

Documentation Plan: The documenter is responsible for preparation of a documentation plan, which specifies the details of the work to be carried out to prepare the document. It also defines and the target audience.

Review of Plan: The plan as set out in the process above is reviewed to see that material acquired is correct and complete.

Creation of Document: The document is prepared with the help of document generator.

Testing of Document: The document created is tested for usability as required by the target audience.

Maintain Document: Once the document is created and distributed, it must be kept up to date with new version of the software product. It must be ensured that the latest document is available to the user of the software.

Figure 4.1 depicts various stages of the process of documentation.

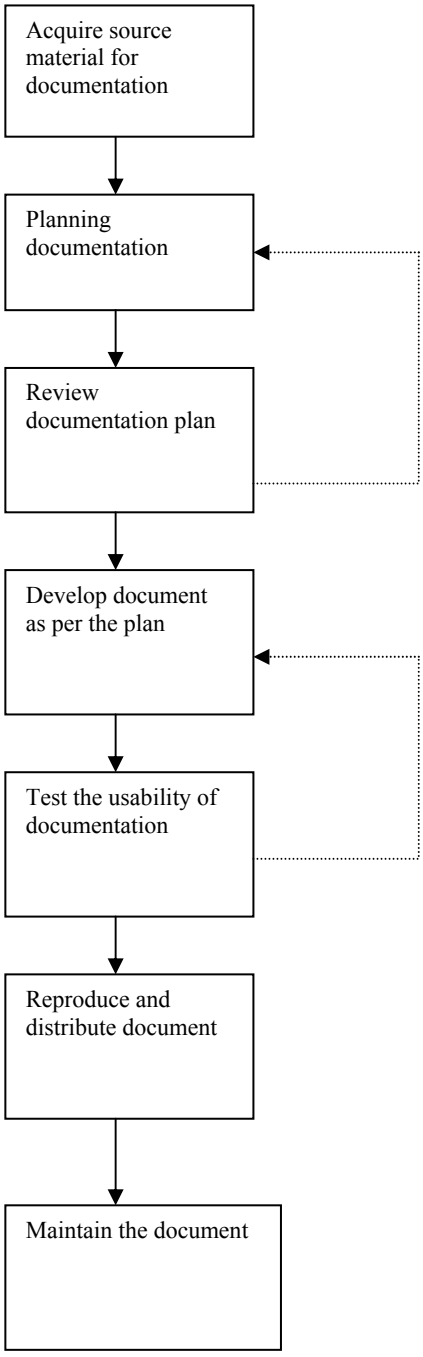


Figure 4.1: Stages of process of documentation

Check Your Progress 1

1. ISO standard ISO/IEC 12207:1995 describes documentation as a _____ activity.

2. The documenter is responsible for preparation of a documentation plan (Yes/No).

4.3 TYPES OF DOCUMENTATION

Any software project is associated with a large number of documents depending on the complexity of the project. Documentation that are associated with system development has a number of requirements. They are used by different types of audience in different ways as follows:

- They act as a means of communication between the members of development team
- Documents are used by maintenance engineer
- Documents are used by the user for operation of the software
- Documents are used by system administrator to administer the system.

4.3.1 System Requirements Specification

System requirement specification is a set of complete and precisely stated properties along with the constraints of the system that the software must satisfy. A well designed software requirements specification establishes boundaries and solutions of system to develop useful software. All tasks, however minute, should not be underestimated and must form part of the documentation.

Requirements of SRS: The SRS should specify only the external system behaviour and not the internal details. It also specifies any constraints imposed on implementation. A good SRS is flexible to change and acts as a reference tool for system developer, administrator and maintainer.

Characteristics of a System Requirements Specification (SRS)

1. All the requirements must be stated *unambiguously*. Every requirement stated has only one interpretation. Every characteristic of the final product must be described using a single and unique term.

2. It should be *complete*. The definition should include all functions and constraints intended by the system user. In addition to requirements of the system as specified by the user, it must conform to any standard that applies to it.

3. The requirements should be *realistic* and achievable with current technology. There is no point in specifying requirements which are unrealisable using existing hardware and software technology. It may be acceptable to anticipate some hardware developments, but developments in software technology are much less predictable.

4. It must be *verifiable and consistent*. The requirements should be shown to be consistent and verifiable. The requirements are verified by system tester during system testing. So, all the requirements stated must be verifiable to know conformity to the requirements. No requirement should conflict with any other requirement.

5. It should be *modifiable*. The structure and style of the SRS are such that any necessary changes to the requirements can be made easily, completely and consistently.

6. It should be *traceable* to other requirements and related documents. The origin of each requirement must be clear. The SRS should facilitate the referencing of each requirement for future development or enhancement of documentation. Each requirement must refer to its source in previous documents.

7. SRS should not only addresses the explicit requirement but also implicit requirements that may come up during the maintenance phase of the software. It must be *usable* during operation and maintenance phase. The SRS must address the needs of the operation and maintenance phase, including the eventual replacement of the software.

Rules for Specifying Software Requirements

The following are the rules for specifying software requirements:

- Apply and use an industry standard to ensure that standard formats are used to describe the requirements. Completeness and consistency between various documents must be ensured.
- Use standard models to specify functional relationships, data flow between the systems and sub-systems and data structure to express complete requirements.
- Limit the structure of paragraphs to a list of individual sentences to increase the tractability and modifiability of each requirement and to increase the ability to check for completeness. It helps in modifying the document when required.
- Phrase each sentence to a simple sentence. This is to increase the verifiability of each requirement stated in the document.

Structure of a Typical SRS Document:

1. Introduction
    - System reference and business objectives of the document.
    - Goals and objectives of the software, describing it in the context of the computer-based system.
    - The scope of the document.
2. Informative description about the system
    - Information flow representation.
    - Information content and structure representation.
    - Description of sub-systems and System interface.
    - A detailed description of the problems that the software must solve.
    - Details of Information flow, content, and structure are documented.
    - Hardware, software, and user interfaces are described for external system.
3. Functional Description of the system
    - Functional description.
    - Restrictions/limitations.
    - Performance requirements.
    - Design constraints.
    - Diagrams to represent the overall structure of the software graphically.
4. Test and validation criteria
    - Performance limitation, if any.
    - Expected software response.
    - It is essential that time and attention be given to this section.
5. Glossary
    - Definitions of all technical or software-specific terms used in the document.
6. Bibliography
    - List and reference of all documents that relate to the software.

7. Appendix

- Supplementary information to the specification.

4.3.2 System Design Specification

The system design specification or software design specification as referred to has a primary audience, the system implementer or coder. It is also an important source of information for the system verification and testing. The system design specification gives a complete understanding of the details of each component of the system, and its associated algorithms, etc.

The system design specification documents all as to how the requirements of the system are to be implemented. It consists of the final steps of describing the system in detail before the coding starts.

The system design specification is developed in a two stage process: In the first step, design specification generally describes the overall architecture of the system at a higher level. The second step provides the technical details of low-level design, which will guide the implementer. It describes exactly what the software must perform to meet the requirements of the system.

Tools for describing design

Various tools are used to describe the higher level and lower level aspects of system design. The following are some of the tools that can be used in the System Design Specification to describe various aspects of the design.

- **Data dictionary**

Definition of all of the data and control elements used in the software product or sub system. Complete definition of each data item and its synonyms are included in the data dictionary. A data dictionary may consist of description of data elements and definitions of tables.

Description of data element:

- Name and aliases of data item (its formal name).
- Uses (which processes or modules use the data item; how and when the data item is used).
- Format (standard format for representing the data item).
- Additional information such as default values, initial value(s), limitations and constraints that are associated with the data elements.

Table definitions

- Table name and Aliases.
- Table owner or database name.
- Key order for all the tables, possible keys including primary key and foreign key.
- Information about indexes that exist on the table.

Database schema: Database schema is a graphical presentation of the whole database. Data retrieval is made possible by connecting various tables through keys. Schema can be viewed as a logical unit from programmer's point of view.

E-R model: Entity-relationship model is database analysis and design tool. It lists real-life application entities and defines the relationship between real life entities that are to be mapped to database. E-R model forms basis for database design.

Security model: The database security model associates users, groups of users or applications with database access rights.

Trade-off matrix

A matrix that is used to describe decision criteria and relative importance of each decision criterion. This allows comparison of each alternative in a quantifiable term.

Decision table

A decision table shows the way the system handles input conditions and subsequent actions on the event. A decision table is composed of rows and columns, separated into four separate quadrants.

Input Conditions	Condition Alternatives
Actions	Subsequent action Entries

Timing diagram

Describes the timing relationships among various functions and behaviours of each component. They are used to explore the behaviours of one or more objects throughout a given period of time. This diagram is specifically useful to describe logic circuits.

State machine diagram

State machine diagrams are good at exploring the detailed transitions between states as the result of events. A state machine diagram is shown in figure 4.2.

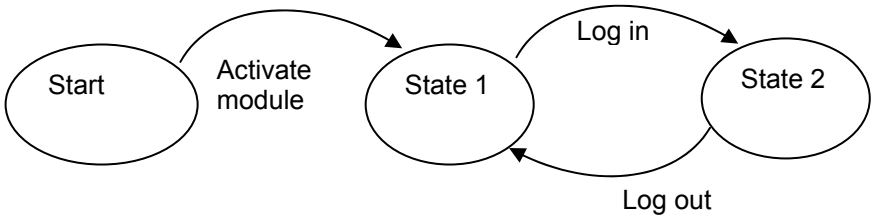


Figure 4.2 : A state machine diagram

Object Interaction Diagram

It illustrates the interaction between various objects of an object-oriented system. Please refer to figure 4.3. This diagram consists of directed lines between clients and servers. Each box contains the name of the object. This diagram also shows conditions for messages to be sent to other objects. The vertical distance is used to show the life of an object.

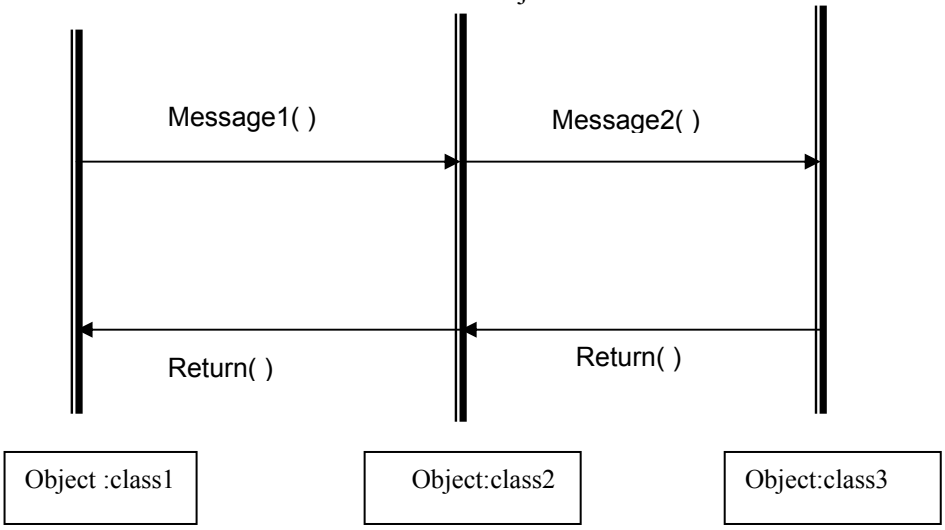


Figure 4.3: An object interaction diagram

A Flow chart shows the flow of processing control as the program executes. Please refer to figure 4.4.

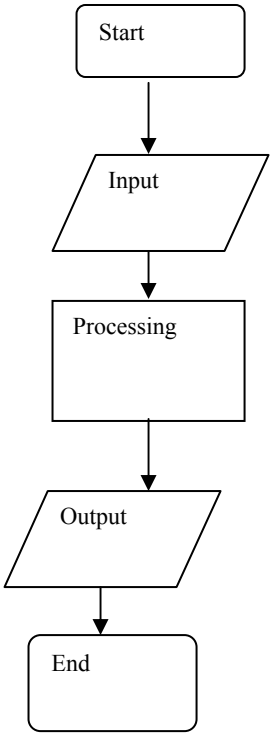


Figure 4.4 : Flow chart

Inheritance Diagram

It is a design diagram work product that primarily documents the inheritance relationships between classes and interfaces in object-oriented modeling. The standard notation consists of one box for each class. The boxes are arranged in a hierarchical tree according to their inheritance characteristics. Each class box includes the class name, its attributes, and its operations. Figure 4.5 shows a typical inheritance diagram.

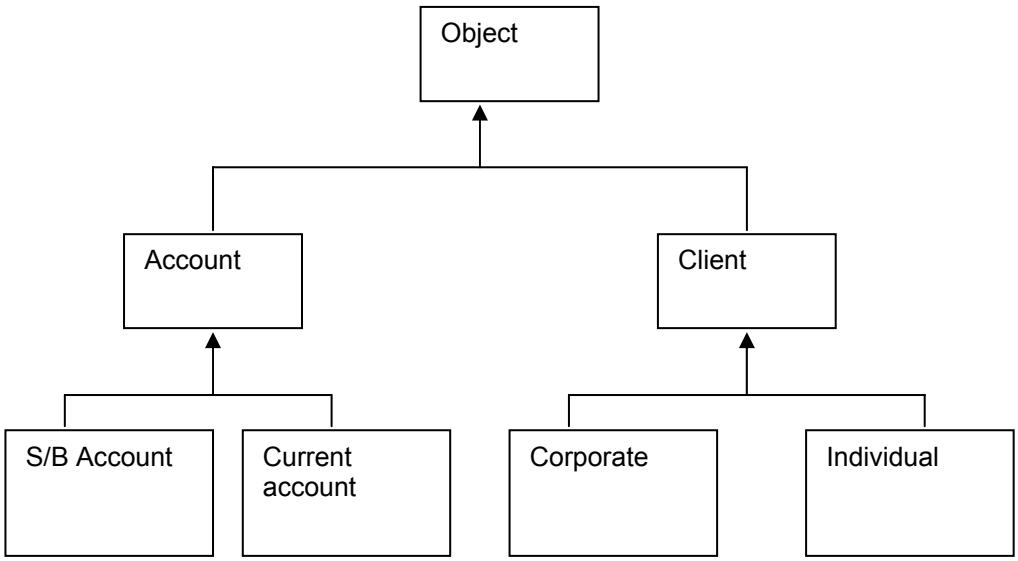


Figure 4.5: A typical inheritance diagram

Aggregation Diagram

The E-R model cannot express relationships among relationships. An aggregation diagram shows relationships among objects. When a class is formed as a collection of other classes, it is called an aggregation relationship

between these classes. Each module will be represented by its name. The relationship will be indicated by a directed line from container to container. The directed line is labelled with the cardinality of the relationship. It describes "has a" relationship. Figure 4.6 shows an aggregation between two classes (circle *has a* shape).

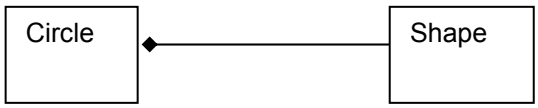


Figure 4.6 : Aggregation

Structure Chart

A structure chart is a tree of sub-routines in a program (Refer to figure 4.7). It indicates the interconnections among the sub-routines. The sub-routines should be labelled with the same name used in the pseudo code.

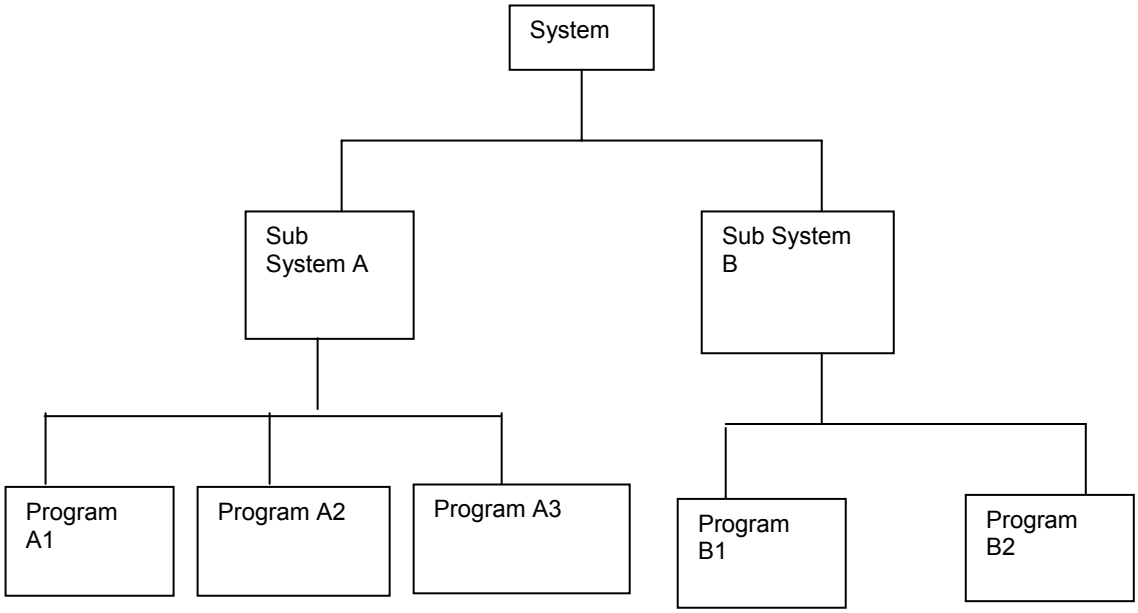


Figure 4.7 : A structures chart

Pseudocode

Pseudocode is a kind of structured English for describing algorithms in an easily readable and modular form. It allows the designer to focus on the logic of the algorithm without being distracted by details of language syntax in which the code is going to be written. The pseudocode needs to be complete. It describes the entire logic of the algorithm so that implementation becomes a routine mechanical task of translating line by line into source code of the target language. Thus, it must include flow of control.

This helps in describing how the system will solve the given problem. "Pseudocode" does not refer to a precise form of expression. It refers to the simple use of Standard English. Pseudocode must use a restricted subset of English in such a way that it resembles a good high level programming language. Figure 4.8 shows an example of pseudo code.

```
IF HoursWorked > MaxWorkHour THEN
        Display overtime message
ELSE
        Display regular time message
ENDIF
```

Figure 4.8 : Example of a Pseudocode

Contents of a typical System Design Specification document content

1. Introduction

   1.1 Purpose and scope of this document:
   Full description of the main objectives and scope of the SDS document is specified.
   1.2 Definitions, acronyms, abbreviations and references
   Definitions and abbreviations used are narrated in alphabetic order. This section will include technical books and documents related to design issues. It must refer to the SRS as one of the reference book.

2. System architecture description

   2.1 Overview of modules, components of the system and sub-systems

   2.2 Structure and relationships

   - Interrelationships and dependencies among various components are described.
   - Here, the use of structure charts can be useful.

3. Detailed description of components

   - Name of the component
   - Purpose and function
   - Sub-routine and constituents of the component
   - Dependencies, processing logic including pseudocode
   - Data elements used in the component.

4. Appendices.

4.3.3 Test Design Document

During system development, this document provides the information needed for adequate testing. It also lists approaches, procedures and standards to ensure that a quality product that meets the requirement of the user is produced. This document is generally supplemented by documents like schedules, assignments and results. A record of the final result of the testing should be kept externally.

This document provides valuable input for the maintenance phase.

The following IEEE standards describe the standard practices on software test and documentation:

1. 829-1998 IEEE Standard for Software Test Documentation
2. 1008-1987 (R1993) IEEE Standard for Software Unit Testing
3. 1012-1998 IEEE Standard for Software Verification and Validation

The following is the typical content of Test Design Document:

1. Introduction

Purpose

The purpose of this *document* and its intended audience are clearly stated.

Scope

Give an overview of testing process and major phases of the testing process. Specify what is not covered in the scope of the testing such as, supporting or not third party software.

Glossary

It gives definition of the technical terms used in this document.

References

Any references to other external documents stated in this document including references to related project documents. They usually refer the System Requirement Specification and the System Design Specification documents.

Overview of Document

Describe the contents and organization of the document.

2. Test Plan

A test plan is a document that describes the scope, approach, resources and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, and the person who will do each task, and any risks that require contingency planning.

2.1. Schedules and Resources

An overview of the testing schedule in phases along with resources required for testing is specified.

2.2. Recording of Tests

Specify the format to be used to record test results. It should very specifically name the item to be tested, the person who did the testing, reference of the test process/data and the results expected by the test, the date tested. If a test fails, the person with the responsibility to correct and retest is also documented. The filled out format would be kept with the specific testing schedule. A database could be used to keep track of testing.

2.3. Reporting test results

The summary of what has been tested successfully and the errors that still exist which are to be rectified is specified.

3. Verification Testing

3.1. Unit Testing

For each unit/component, there must be a test which will enable tester to know about the accurate functioning of that unit.

3.2. Integration testing

Integration test is done on modules or sub-systems.

4. **Validation Testing**

4.1. System Testing

This is the top level of integration testing. At this level, requirements are validated as described in the SRS.

4.2. Acceptance and Beta Testing

List test plans for acceptance testing or beta testing. During this test, real data is used for testing by the development team (acceptance testing/alpha testing) or the customer (beta testing). It describes how the results of such testing will be reported back and handled by the developers.

4.3.4 User Manual

This document is complete at the end of the software development process.

Different Types of User Documentation

Users of the system are not of the same category and their requirements vary widely. In order to cater to the need of different class of user, different types of user documentation are required. The following are various categories of manuals:

- Introductory manual: How to get started with the system?
- Functional description: Describes functionality of the system.
- Reference manual: Details about the system facility.
- System administrator guide: How to operate and maintain the system?
- Installation document: How to install the system?

The following is the typical content of User Manual

1. Introduction

1.1 Purpose

The purpose of this *document* and its intended audience is stated. If there is more than one intended audience, provide information in this section and direct the reader to the correct section(s) for his/her interest.

1.2 Scope of Project

Overview the product . Explain who could use the product. Overview the services that it provides. Describe limitation of the system. Describe any restrictions on using or copying the software and any warranties or contractual obligations or disclaimers.

1.3 Glossary

Define the technical terms used in this document. Do not assume that the reader is expert.

1.4 References

References to other documents cited anywhere in this document including references to related project documents. This is usually the only bibiliography in the document.

1.5 Overview of Document

The contents and organization of the rest of this document are described.

2. **Instructional Manual**

This section should be divided in the manner that will make it user friendly.

2.1 System Usage

Provide examples of normal usage. Images of the screendumps are very useful to provide a look and feel of the product. Provide any necessary background information. On-line help system is very common for systems today. Information on how to use the on-line help system, how to access it may be provided here.

3. **User Reference Manual**

3.1 List of Services

Provides an *alphabetical* listing of services provided by the system with references to page numbers in this document where the concerned service is described.

3.2 Error Messages and Recovery

Provides an *alphabetical* list of all error messages generated by the system and how the user can recover from each of these errors.

4. **Installation Information**

Installation information is provided including the operating environment.

Maintenance Manual

The supplier/developer of the software is sometimes different from the software maintenance agency. In such cases, the criticality of maintenance manual assumes a bigger role. Maintenance manuals provide precise information to keep your product operating at peak performance. Similar to installation manuals, these documents may range from a single sheet to several hundred of pages.

Check Your Progress 2

1. The system design specification is developed in _____ process.
2. Pseudo code is used to describe _____.
3. _____ provides information as to how to operate and maintain the system.

4.4 DIFFERENT STANDARDS FOR DOCUMENTATION

This software documentation standard is used in the organization for uniform practices for documentation preparation, interpretation, change, and revision, to ensure the inclusion of essential requirements of different standards. Sometimes, documentation as per various standards is stated in the contractual agreement between the software vendor and the customer.

This standard will also aid in the use and analysis of the system/sub-system and its software documentation during the system/software life cycle of a software project. Documentation comes in many forms, e.g., specifications, reports, files, descriptions, plans, source code listings, change requests, etc. and can be in electronic or paper form.

The Documentation Standard defines various aspects of documentation such as style, format, and the document revision/change process of these documents.

The International Standards, ISO/IEC 12207 – Software life cycle process, describes documentation as one of the supporting parallel process of software development process. It may be noted that this standard is not documentation standard but describes the process of documentation during the software development process. The following are other documentation standards:

1. ISO/IEC 18019: Guidelines for the design and preparation of user documentation for application software

This standard describes how to establish what information users need, how to determine the way in which that information should be presented to the users, and then how to prepare the information and make it available. It covers both on-line and printed documentation. It describes standard format and style to be adopted for documentation. It gives principles and recommended practices for documentation.

2. ISO/IEC 15910: Software user documentation process

This standard specifies the minimum process for creating user documentation for software that has a user interface, including printed documentation (e.g., user manuals), on-line documentation, help text and on-line documentation systems.

3. IEEE 1063: Software user Documentation

It provides minimum requirement for structure, information content and format for user documentation. It does not describe the process to be adopted for documentation. It is applicable for both printed and on-line documentation.

Components of software user documentation as described in IEEE 1063: Software user Documentation:

Components of software user documentation:

1. Identification data (e.g., Title Page)
2. Table of contents
3. List of illustrations
4. Introduction
5. Information for use of the documentation such as description of software etc.
6. Concept of operations
7. Procedures
8. Information on software commands
9. Error messages and problem resolution
10. Glossary (to make the reader acquainted with unfamiliar terms)
11. Related information sources
12. Navigational features
13. Index
14. Search capability (for electronic document).

Documentation involves recording of information generated during the process of software development life cycle. Documentation process involves planning, designing, developing, distributing and maintaining documents.

During planning phase, documents to be produced during the process of software development are identified. For each document, following items are addressed:

- Name of the document
- Purpose
- Target audience
- Process to develop, review, produce, design and maintain.

The following form part of activities related to documentation of development phase:

- All documents to be designed in accordance with applicable documentation standards for proper formats, content description, page number, figure/table.
- Source and accuracy of input data for document should be confirmed.
- Use of tools for automated document generation.
- The document prepared should be of proper format. Technical content and style should be in accordance to documentation standards.

Production of the document should be carried out as per the drawn plan. Production may be in either printed form or electronic form. Master copy of the document is to be retained for future reference.

Maintenance

As the software changes, the relevant documents are required to be modified. Documents must reflect all such changes accordingly.

Check Your Progress 3

1. _____ is used in the organization for uniform practices for documentation.
2. International Standard ISO/IEC 12207 is documentation standard (Yes/No).

4.5 DOCUMENTATION AND QUALITY OF SOFTWARE

Inaccurate, incomplete, out of date, or missing documentation is a major contributor to poor software quality. That is why documentation and document control has been given due importance in ISO 9000 standards, SEI CMM software Maturity model. In SEI CMM Process Model and assessment procedure, the goal is to improve the documentation process that has been designed. A maturity level and documentation process profile is generated from the responses to an assessment instrument.

One basic goal of software engineering is to produce the best possible working software along with the best possible supporting documentation. Empirical data show that software documentation products and processes are key components of software quality. Studies show that poor quality, out of date, or missing documentation are a major cause of errors in software development and maintenance. Although everyone agrees that documentation is important, not everyone fully realizes that documentation is a critical contributor to software quality.

Documentation developed during higher maturity levels produces higher quality software.

4.6 GOOD PRACTICES FOR DOCUMENTATION

1. *Documentation is the design document.* The time to document is before actually implementing any design. A lot of effort can be saved in such cases.
2. *Good documentation projects the quality of software.* Many people take poor, scanty, or illiterate documentation for a program as a sign that the programmer is sloppy or careless of potential users' needs. Good documentation, on the other hand, conveys a message of intelligence and professionalism. If your program has to compete with other programs, better make sure that your documentation is at least as good as your competitors.

Check Your Progress 4

1. SEI CMM is a _____ model.
2. One of the basic goals of software engineering is _____.

4.7 SUMMARY

Documentation is an integral part of software development process and should not be taken lightly. Incomplete and inaccurate documentation may pose serious hurdle to the success of a software project during development and implementation. The documentation process itself requires proper planning like the software development process. Various documents like System requirement specification (SRS), system design specification (SDS), test design document and user manuals are produced during the life cycle of a software development process. SRS documents the high level requirements of the system without going into details of implementation issues, whereas system design document describes how the requirements are finally to be implemented. It also describes the implementation issues with help of various system design tools. Test design document documents the requirement to be tested and procedure to be followed for testing. We have also discussed various ISO and IEEE standards on user documentation for uniform practices on documentation style and process. The relation of documentation with software quality has also been discussed.

4.8 SOLUTIONS/ ANSWERS

Check Your Progress 1

1. Support
2. Yes

Check Your Progress 2

1. Two stage.
2. Algorithms.
3. System administration guide.

Check Your Progress 3

1. Software documentation standard.
2. No.

Check Your Progress 4

1. Capability Maturity.
2. To produce accurately working software along with the best possible supporting documentation.

4.9 FURTHER READINGS

- Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich; *Modern Systems Analysis and Design*; Pearson Education; Third Edition; 2002.
- ISO/IEC 12207: *Software life cycle process*
- IEEE 1063: *Software user Documentation*
- ISO/IEC: 18019: *Guide lines for the design and preparation of user documentation for application software*

Reference Websites

- http://www.sce.carleton.ca/squall
- http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/documentation.html
- http://www.sei.cmu.edu/cmm/

SOCIAS-IGNOU/P.O.10.5T/MAY, 2004

ISBN-81-266-1230-4