Unit 1 Information Systems

Lesson Structure

1.0 Objective

After going through this unit you will learn:

- Fundamentals of system
- Classification of system
- Real and distributed systems
- Development of a successful system
- Structured analysis and design approach and
- Joint application development

1.1 Introduction

In a broad term we can say that system is a group or set of components that interact and work together to complete a specific task. A system is based on input process and output. System may be computerized or manual. To develop a computerized system the developers should 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.

1.2 Fundamentals of System

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 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. Following are the characteristics of a system.

Organization

Organization implies structure and order. It is the arrangement of components that helps to achieve objectives. A computer system is designed around an input device, a central processing unit, an output device and one or more storage units. When linked together they work as a whole system for producing information.

Interaction

Interaction refers to the procedure in which each component functions with other components of the system. In a computer system the central processing unit must interact with the input device to solve a problem. In turn, the main memory holds programs and data that the ALU uses for computation. The interrelationship between these components enables the computer to perform.

Interdependence

Interdependence means that part of the organization or computer system depend on one another. They are coordinated and linked together according

to a plan. One subsystem depends on the input of another subsystem for proper functioning. That is, the output of one subsystem is required input of another subsystem.

Integration :

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 partperforms a unique function.

Central Objective :

Central Objectives quite common that an organization may set one objective and operate to achieve another. The important point is that the users must know the central objective of a computer application early in the analysis for a successful design and conversation.

1.3 Important Term Related to System

Purpose:

Purpose is the reason for existence of a system and the reference point for measuring its success.

Boundary:

Boundary defines what is inside the system and what is outside.

Environment:

Environment is everything pertinent to the system that is outside of its boundaries.

Inputs:

Inputs are the physical objects and information that cross the boundary to enter it from its Environment.

Outputs:

Outputs are the physical objects and information that go from the system into its environment.

1.4 Classification of Systems

Systems may be classified as follows:

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

a. Physical or Abstract :

Physical systems are tangible entities that may be static or dynamic in operation. For example, the physical parts of computer center are the offices, desk, and chairs that facilitate operation of the computers. They can be seen

and counted; they are static. In contrast, a programmed computer is a dynamic system. Data, programs, output, and applications change as the user's demands or the priority of the information requested changes.

Abstract systems are conceptual or nonphysical entities. They may be as straightforward *as* formulas of conceptualization of physical situations. A model is a representation of a real or a planned system. The use of models makes it easier for the analyst to visualize relationships in the system under study. The objective is to point out the significant elements and the key interrelationships of as complex system.

b. Open or Closed

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. 7 **Introduction to SAD**

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.

c. Formal or Informal

A *formal information system* is based on the organization represented by the organization chart. The chart is a map of positions and their authority relationship, indicated by boxes and connected by straight lines. It is concerned with the pattern of authority, communication, and work flow. Information is formally disseminated in instructions memos, or reports from top management to the intended user in the organization. This structure also allows feedback up the chain of command for the follow-up.

The *informal information system* is a power structure designed to achieve company goals. An organization's emphasis on control to ensure performance tends to restrict the communication flow among employees, however. As a result, an informal information system develops. It is an employee-based system designed to meet personnel and vocational needs and to help solve, work-related problems. It also funnels information upward through indirect channels. In this respect, it is a useful system because it works within the framework of the business and its stated policies.

d. Manual or Automated

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.

1.4.1 Real Life Business Subsystem

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.

1.4.2 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.

1.4.3 Distributed System

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 smallmicroprocessors, 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

1. Resource sharing
2. Computation speedup
3. Reliability
4. Communication

The five Layers of Distributed System architecture are:

1. Application Logic Layer : includes all the logic and processing required to support the actual business application and rules

Example: Calculations.

2. Data Manipulation Layer : includes all the command and logic required to store and retrieve data to and from the database.

3. Data Layer : is actual stored data in the database.

4. Presentation Layer : is the actual user interface. The inputs are received by this layer and the outputs are presented by this layer.

5. Presentation Logic layer : includes processing required to establish user interface.

Example: Editing input data, formatting output data.

1.5 Development of a Successful System

Development of a successful system depends on the approach of building it. if the ap[roach is right, the system will work successfully.

System development life cycle is a methodology for development of information system. System development is a process consisting of two major steps of system analysis and design, starts when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The systems development life cycle is classically thought of as the set of of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities.

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation and maintenance

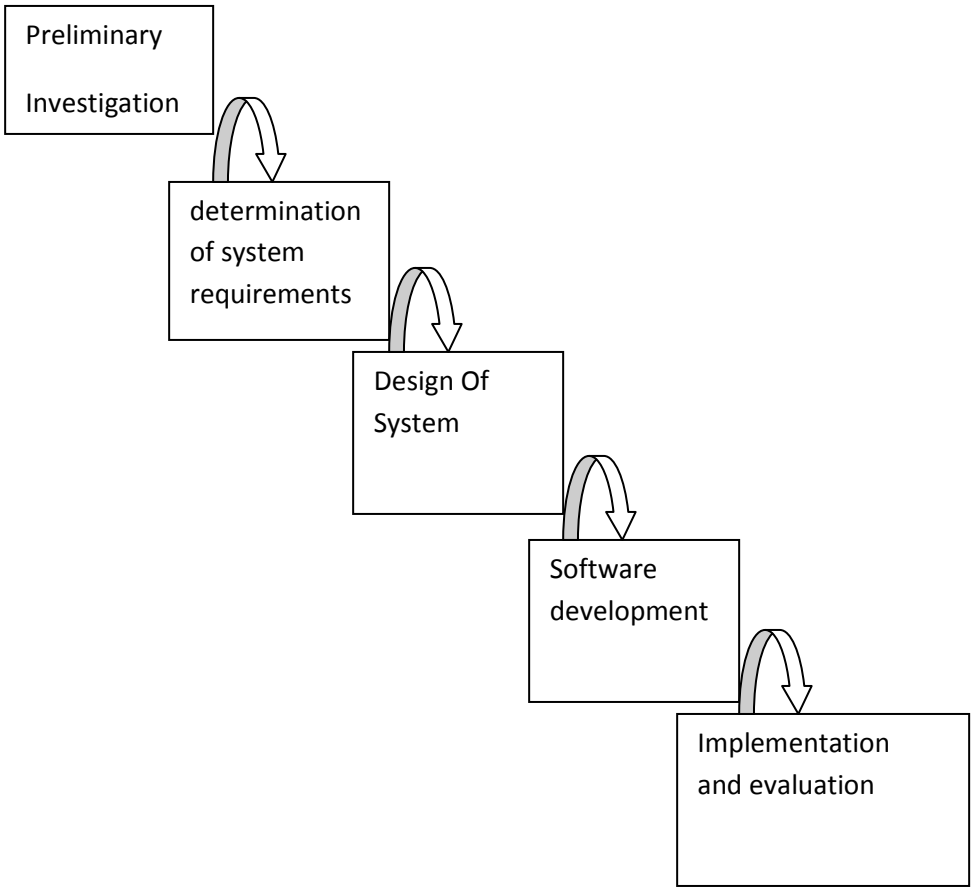

Figure 1 : System Development Life Cycle

For 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. 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:

l **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.

l **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.

l **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

Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specifications that are easily understandable to the user. The traditional approach focuses on cost / benefit analysis, project management, hardware and software selection, and personnel considerations. In contrast, structured analysis considers new goals and structured tools for analysis.

- The new goals specify the following.
- Use graphics wherever possible to help communicate better with user.
- Differentiate between logical and physical systems.
- Build a logical system model to familiarize the user with system characteristics and interrelationships before implementation.

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 is a data flow based methodology. the approach begins with a system specification that identifies inputs and outputs and describes the functional aspects of the system. The specification, then are used as a basis for the graphic representation. the next step is the definition of the modules and their relationships to one another in a form called a structure chart, using a data dictionary and other structured tools.

Logical design proceeds from top down. general features, such as reports and inputs are identified first. Then each is studied individually and in more detail. Hence, the structured design partitions a program into small, independent modules. they are arranged in a hierarchy that approximates a model of the business area and is organized in a top down manner. Thus structured design is an attempt to minimize the complexity and make a problem manageable by subdividing it into smaller segments which is called modularization or decomposition. In this way, structuring minimizes intuitive reasoning and promotes maintainable provovable systems.

A design is said to be top down if it consists of a hierarchy of modules,with each module having a single entry and a single exit subroutine. The primary advantage of this design are as follows.

- Critical interfaces are tested first.
- Early versions of design, though incomplete, are useful enough to resemble the real system.
- Structuring the design, perse, provides control and improves morale.
- The procedural characterstics define the order that determines processing.

Principles of structured design

Modularisation

In structured design a program is segmented into small independent modules. These are arranged in a hierarchy that approximates a model of the business area and is organized in a top down manner with the details . in structure design we try to minimize the complexity of the problem and make it manageable by subdividing it into smaller sements which is called modularization.

Coupling

Coupling is the manner and degree of interdependence between software modules a measure of how closely connected two routines or modules are

the strength of the relationships between modules. Low coupling is often a sign of a well-structured computer system and a good design, and when combined with high cohesion, supports the general goals of high readability and maintainability.

Cohesion

Cohesion determines how closely the elements of a module are related to each other. Cohesion of a module represents how tightly bound the internal elements of the module are to one another. Cohesion of a module gives the designer an idea about whether the different elements of a module belong together in the same module. Modules should be highly cohesive.

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

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 Prototyping

Prototype is a working system that is developed to test ideas and assumptions about the new system. Like any computer based system, it consists of working software that accepts input, perform some operation on it and gives the output.

The prototype is actually a pilot test model. It is designed to be easily changed. Information gained through its use is applied to a modified design that may again be used as a prototype to reveal still better design information. The process is repeated as many times as necessary to reveal essential design requirements. In general, prototypes are considered to be most useful under following conditions.

- No system with the characteristics of the use of the one proposed has yet been constructed by the developers.
- The essential features of the system are only partially known, others are not identifiable even through careful analysis of reqirements.
- Experience in using the system will significantly add to the list of requirements the system should meet.

- Alternate versions of the system will evolve through experience and additional decelopment and refinement of its features.
- The system users will participate in the development process.

Typically, these are the steps in the prototyping process.

- 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.
- Use the prototype, nothing needed enhancements and changes. These expand the list of known system requirements.

As these steps suggest, prototyping is not a trial and error development process. Before starting the system design work, user and system analyst sit together to discuss to identify the requirements. These discussions from the basis for the construction of the prototype. System analyst is fully responsible for the development of the working prototype.

1.7 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.

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.

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.

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.

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.

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.

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 process.

1.8 Summary

In this unit we learned the basic concepts of system. Types and characteristics of system. We also learned about development of a successful information system. Prototype is a working system that is developed to test ideas and assumptions about the new system. Like any computer based system, it consists of working software that accepts input, perform some operation on it and gives the output. JAD 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. Selection of the approach is based on the end user requirements, problem identified and infrastructure provided.

1.9 Questions for Exercise

1. Define the following terms:
   i. Structured design
   ii. Structured Analysis
   iii. Real life business system
   iv. Distributed system
   v. protoyping
2. What are the characteristics of a good information system?
3. Explain the fundamental principles of Software Development Life Cycle.
4. What do you understand by joint Application Development? Explain

1.10 Suggested Readings

1. Alan Dennis, Barbara Haley Wixom; *Systems Analysis and Design;* John Wiley & Sons; 2002.
2. Joey George, J. Hoffer and Joseph Valacich; *Modern Systems Analysis and Design*; Pearson Education; Third Edition; 2001.
3. Elias M. Awad; Systems Analysis and Design; Galgotia Publications; Second Edition; 1997
4. Perry Edwards; Systems Analysis and Design; McGraw Hill Publication; 1993.

Referenced Link

1. http://systemanalysisanddesign.blogspot.in/
2. en.wikipedia.org/wiki
3. en.wikipedia.org/wiki/**Systems_development_life_cycle**
4. www.tutorialspoint.com

●●●