Unit 2 System Development Life Cycle

Lesson Structure

2.0 Objective

After going through this unit you will learn:

- System development life cycle
- Different phases of SDLC
- Different approaches to development
- Role of a system analyst and
- Qualities and qualification of a system analyst

2.1 Introduction

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.

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.

2.2 System Development Life Cycle

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 activities that analysts, designers and users carry out to develop and implement an information system.

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.

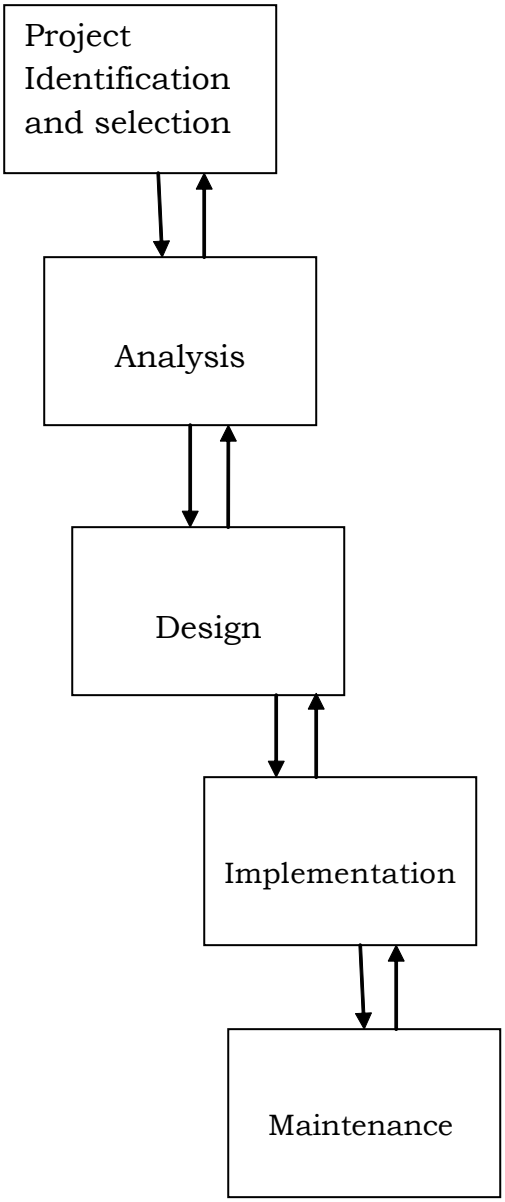

Figure 1: Phases of System Development Life Cycle

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.

2.3 Phases of SDLC

The system development life cycle consists of the following activities.

- Project Identification and Selection
- Project Initiation and planning
- Design of system
- Implementation
- and maintenance

2.3.1 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

2.3.2 Project Initiation and Planning

The next 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 theremaining SDLC steps. 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.

2.3.3 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
- Operational feasibility
- Behavioural feasibility
- Legal feasibility
- Time feasibility.
- Social feasibilty

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.

2.3.4 Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams.

2.3.5 Physical design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed as In Physical design, the following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing Requirements,
5. System control and backup or recovery.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

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 implementationphase

2.3.6 Implementation

Software developers may install purchased software or they may develop new, custom designed programs. The choice depends on the cost of each option, the time available to develop software and the availability of programmers. Generally it has been observed that programmers are part of permanent professional staff in a big organization. In smaller organization, without programmers, outside programming services may be hired or retained on a contractual basis. Programmers are also responsible for documenting the program, providing an explanation of how and why certain procedures are coded in specific ways. Documentation is essential to test the program and carry on maintenance once the application has been installed.

Implementation is the process of having systems personnel check out and put new equipment into use, trained users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two two persons. Sometimes, they will run both old and new system in parallel way to compare the results. In still other situations, system developers stop using the old system one day and start using the new one the next.

2.3.7 Maintanance

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 implements 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.

2.4 Products of SDLC Phases

These are the following products of SDLC Phases

1. 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. 2. 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.

2. Analysis: Description of current system, need to enhance or replace current system, explanation of alternative systems and justification of alternatives.

3. Logical Design: Functional and detailed specification of all system elements (data, process, input and output).

4. Physical design: Technical, detailed specifications of all system elements, i.e., programs, files, network, system software, etc. and acquisition plan for new technology.

5. Implementation: Code, documentation, training programs and support capabilities.

6. Maintenance: New version of software with associated updates of documents, training and support.

2.5 Approaches to Development

To improve the system analysis and design process, many approaches have been developed. 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 finalize

We will discuss prototyping joint application design and participatory design.

2.6 Prototyping

The third strategy for determining user information requirements is used when the user cannot establish information needs accurately before the information system is built. The reason could be the lack of an existing model n which to base requirements or a difficulty in visualizing candidate systems. In this case, the user needs to anchor on reallife systems from which adjustments can be made. Therefore, the iterative discovery approach captures an initial set of information requirements and builds a system to meet these requirements. As user gain experience in its use, they request

additional requirements or modifications (iterations), in the system in essence, information requirements are discovered by using the system. Prototyping is suitable in environments where it is difficult to formulate a concrete model for defining information requirements and where the information needs of the usr are evolving, such as in DSS. Which of the three strategies is selected depends on uncertainties in the process of determining information requirements – that is, uncertainly with respect to the stability of information requirements, the user's ability to articulate information requirements, and the ability of the analyst to elicit requirements and evaluate their accuracy. Thus, the asking strategy is appropriate for low- uncertainty information requirements determinations, whereas the prototyping strategy is appropriate for high uncertainty information requirements determination. The steps of Prototyping process 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.

2.7 Joint Application Design

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.

2.8 Participatory Design

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.

2.9 Role of a System 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:

l **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.

l **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.

l **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.

l **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.

l **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.

l **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

2.10 Quality and Qualification of a System Analyst

A systems analyst must fulfill 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:

1. Analytical skills
2. Technical skills
3. Management skills
4. Interpersonal skills.

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 24
- 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. 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.

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.

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.

2.11 Case Study

The problem is to computerize of ABC Hospital. In this hospital, all transactions are handled manually. Registers are maintained to record the details of the patient, information about the Doctors of the hospital and to manipulate the records and transactions. 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 hospital is very tough and time-consuming and chances of getting errors gets very high.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 hospital.
- To handle the patient details.
- To record and handle the doctor details.
- To handle admit and discharge of patient

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 Hospital Management system . Our Data Dictionary record stores the following descriptions:

Patient:- Stores information about the paitent

Field Name	Data type	Size	Constraint	Description
Patient_Id	Integer	5	Primary Key	Patient Identification.
Patient_Name	Varchar	50		Name of the Patient.
Patient_Father_Name	Varchar	50		Name of the Patient father.
Patient_DoB	Date/Time	10		Date of Birth of the Patient.
Patient_Address	Varchar	50		Address of the Patient.
Patient_Contact_No.	Varchar	10		Contact Number of the Patient.
Patient_Blood_Group	Varchar	3		Blood Group of the Patient.
Patient_Gender	Varchar	6		Gender of the Patient.
Patient_Marital_Status	Varchar	10		Marital Status of the Patient.

Master_Doctor:- Stores information about the doctor

Field Name	Data type	Size	Constraint	Description
Doc_Id	Integer	3	Primary Key	Doctor Identification.
Doc_Name	Varchar	50		Name of the Doctor.
Doc_Age	Integer	3		Age of the Doctor.
Doc_Qualification	Varchar	50		Qualification of the Doctor.
Doc_Address	Varchar	50		Address of the Doctor.
Doc_Contact_No	Varchar	10		Contact Number of the Doctor.
Doc_Gender	Varchar	6		Gender of the doctor.
Doc_Designation	Varchar	50		Designation of the Doctor.

Master_Department:- stores information about department

Field Name	Data type	Size	Constraint	Description
Department_Id	Integer	2	Primary Key	Unique department identification.
Department_Name	Varchar	25		Name of the Department.

Master_Bed:- stores bed details

Field Name	Data type	Size	Constraint	Description
Bed_Id	Integer	5	Primary Key	Unique Bed Identification.
Bed_No	Integer	5		Bed Number of the Patient.
ward_Id	Integer	2		Ward Identification.
Status	Varchar	10		It shows the Bed Status.

Account Detail:- stores account detail

Field Name	Data type	Size	Constraint	Description
Accounts_Id	Integer	5	Primary Key	Unique Account Identification Number.
Patient_Reg_Id	Integer	11	Foreign Key	Patient Registration ID.
Total_Days	Integer	4		Total days of Patient.
Bed_Charges	Integer	5		Total no of days X Bed Accommodation Rate.
Test_charges	Integer	5		Total Test Charges.
Nursing Home_Fee	Integer	5		Nursing Home Fee.
Ambulance_Charges	Integer	5		Ambulance Use Charges.
Total_Charges	Integer	6		Sum of Bed, Test, Nursing Home And Ambulance Charges.
Tax	Integer	5		Tax Calculated on Total Charges.
Rebate	Integer	5		Discount allowed by the Nursing Home.
Net_Payable	Integer	6		Net Payable Amounts.

Patient Registration Detail:- stores patient registration detail

Field Name	Data type	Size	Constraint	Description
Patient_Reg_Id	Varchar	11	Primary Key	Unique Patient Registration Identification.
Patient_Id	Integer	5	Foreign key	Patient Identification.
Admission_date	Date/Time	10		Admission Date of Patient.
Release_date	Date/Time	10		Release date of Patient.
Bed_Id	Integer	5		Bed Identification of Registered Patient.
Status	Varchar	8		Status of Registration.

Routine Examination:- stores routine examination details

Field Name	Data type	Size	Constraint	Description
Routine_Id	Integer	5	Primary Key	Unique Routine Identification.
Patient_reg_Id	Varchar	11		Registered Patient Identification.
Doc_Id	Integer	3	Foreign Key	Doctor Identification.
Date	Date/Time	10		Date of Routine Examination.
Time	Date/Time	9		Time of Routine Examination.
BP	Varchar	3		Blood Pressure.
Pulse	Integer	2		Pulse Rate of Patient.
Weight	Integer	3		Weight of the Patient.
Precautions	Varchar2	400		Precaution advised by the Doctor.
Comment	Varchar2	400		Comment given by the Doctor.
Next_Visit_date	Date/Time	10		Next Visit Date of Patient.

Department Doctor Map:- stores doctors department detail

Field Name	Data type	Size	Constraint	Description
Dept_ID	Integer	2		Department Identification.
Doc_ID	Integer	3		Doctor Identification.

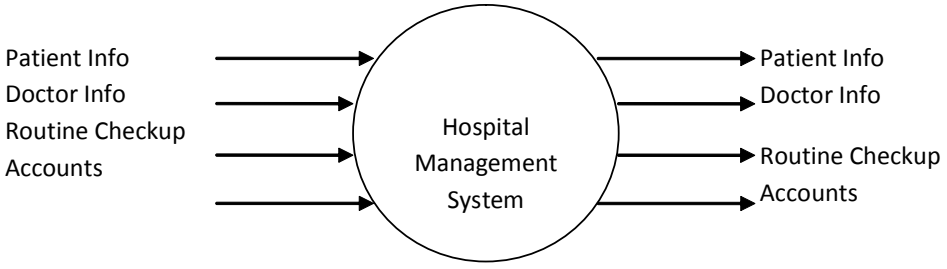

Figure : 3 **Zero level DFD**

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 patient is identified by the unique code, the details of the doctor through a unique doctor 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 canbe kept as a hardcopy in the printed layouts. Our system produces following reports:

1. Patient Detail Report.
2. Accounts Report
3. Doctor Detail Report
4. Routine Checkup Report

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

Patient:– (Patient_Id, Patient_Name, Patient_Father_Name, Patient_DoB, Patient_Address, Patient_Contact_No., Patient_Blood_Group, Patient_Gender, Patient_Marital_Status). **Master_Doctor:**– (Doc_Id, Doc_Name, Doc_Age, Doc_Qualification, Doc_Address, Doc_Contact_No, Doc_Gender, Doc_Designation)

Master_Department: (Department_Id, Department_Name)

Master_Bed: (Bed_Id, Bed_No, ward_Id, Status)

Account Detail: (Accounts_Id, Patient_Reg_Id, Total_Days, Bed_Charges, Test_charges, Nursing Home_Fee, Ambulance_Charges, Ambulance_Charges, Total_Charges)

Patient Registration Detail: (Patient_Reg_Id, Patient_Id, Admission_date, Release_date, Bed_Id, Status)

After these steps, coding in any programming languages can be done and then the system will be tested against the requirements of the user.

2.12 Summary

In this unit, you learned about 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. You also learned Prototyping (Rapid Application Development approach), Joint Application Design and Participatory Design.

We also learned the skills necessary a system analyst. 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

2.13 Questions for Exercise

1. What are the different phases of SDLC? Explain.
2. What do you understand by logical and physical design? Explain.
3. What are the products of SDLC Phases?
4. What is prototyping?
5. what are the roles of a system Analyst? Explain.
6. What are the qualities and qualifications of a system Analyst.
7. Define joint application development.

2.14 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

●●●