MANONMANIAM SUNDARANAR UNIVERSITY

DIRECTORATE OF DISTANCE & CONTINUING EDUCATION

TIRUNELVELI 627012, TAMIL NADU

M.B.A. SYSTEM - II YEAR

DKY24 - SYSTEM ANALYSIS AND DESIGN

(From the academic year 2016-17)

Most **S**tudent friendly **U**niversity - Strive to Study and Learn to Excel

For more information visit: *http://www.msuniv.ac.in*

DKY24 - SYSTEM ANALYSIS AND DESIGN

Unit I

System concept & Information system environment: system concepts – definition – characteristic a system – elements of systems – introduction to systems analysis and design – system analysis – system design – system development life cycle.

Unit II

Introduction – where does the systems analyst come from? What does he do? Preparing for case as a system analyst general business knowledge – Technical skills – Communication skills – Role of system analyst – change agent – investigator and monitor – Psychologist – Sales person – motivator – Politician plan of the system analyst – position in the MIS organisation.

Unit III

Problems with development life cycle approach need for structured approach: Information gathering A problem solving approach – Data flow diagrams: data modeling with logical entity relationship; process modeling with logical data flow diagram; data dictionary; decision TREE – Decision Tables; structures English.

Unit IV

Introduction to the process of design logical and Physical: designing conventional computer files and control modern computer data bases, computer outputs and controls, computer inputs and control. Code design, computer based methods, procedures and controls.

Unit V

System testing conversion – combating Resistance to change: past implementation Review: software maintenance: Hardware / Software Selection: Security disaster / Recovery and Ethics in System Development.

References:

1. System analysis and Design – Elias M. Awad
2. System Analysis and Design Methods – Jeffrey I. Written Lonney D. Benticy & Victor M. Bar
3. System Analysis and Design – A case study approach – Robert J. Thicrant.

DKY24: System Analysis and Design

Contents

Unit I

System concept and Information System Environment

System Concepts

System is defined as a set of elements arranged in an orderly manner to accomplish an objective. System is not a randomly arranged set. It is arranged with some logic governed by rules, regulations, principles and policies. Such an arrangement is also influenced by the objective the system desires to achieve. Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design (SAD) mainly deals with the software development activities.

For example, if a computer system is designed to perform commercial data processing, then the elements will be data entry devices, a CPU, a disk, a memory, application programs and a printer. If a computer is designed to achieve the objective of design, engineering and drawing processing, then the elements will be the graphic processor, and the languages suitable for engineering and design applications, and plotters for drawing the output.

However, a clear statement of objectives brings a precision and an order into the selection of elements and their arrangements in the system. Any disorder would create a disturbance in the system, affecting the accomplishment of the objectives.

If a system in any field is analyzed. It will be observed that it has three basic parts, which are organized in an orderly manner. These three parts can be represented in a model as shown:-

A system may have single input and multiple outputs or may have several inputs and outputs. All the systems operate in an environment. The environment may influence the system in its design and performance. When a system is designed to achieve certain objectives, it automatically sets the boundaries for itself. The understanding of boundaries of the system is essential to bring clarity in explaining the system components and their arrangements.

A collection of component that work together to realize some objectives forms a system. In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth.

The objective of the system demands that some output is produced as a result of processing the suitable inputs. A well designed system also includes an additional element referred to as 'control' that provides a feedback to achieve desired objectives of the system.

The system can be classified in different categories based on the predictability of its output and the degree of information exchange with the environment. A system is called deterministic when the inputs, the process and outputs are known with certainty. In a system, if the output can only be predicted in probabilistic terms, then the accounting system is a probabilistic one. A deterministic system operates in a predictable manner while a probabilistic system behavior is not predictable.

Definition of a System

The term system may be defined as an orderly arrangement of a set of interrelated and interdependent element that operate collectively to accomplish some common purpose or goal.

For example – human body is a system, consisting of various parts such as head, heart, hands, legs and so on. The various body parts are related by mean of connecting networks of blood vessels and nerves and the system has a main goal of "living". Thus, a system can be described by specifying its parts, the way in which they are related, and goals which they are expected to achieve.

A business is also a system is also a system where economic resources such as people, money, material, machines, etc are transformed by various organizational processes (such as production, marketing, finance etc.) into goods and services.

A computer based information system is also a system which is a collection of people, hardware, software, data and procedures that interact to provide timely information to authorized people who need it.

Characteristics of a system

There are five types of characteristics for a system. They are

1. Organization

2. Interaction

3. Interdependence

4. Integration

5. A central objective

ORGANIZATION:

Organization implies structure and order. It can also be defined as the arrangement of components that helps to achieve objectives.

For example in the design of a business system, the hierarchical relationships starting with the president on top and leading towards the workers represents the organization structure. So this gives the authority structure and specifies the formal flow of communication.

Like wise a computer system is designed around an input device, a central processing unit, an output device and one or more storage units.

INTERACTION:

Interaction refers to the manner in which each component functions with other components of the system. ie, there should be an interrelationship between each components of a system.

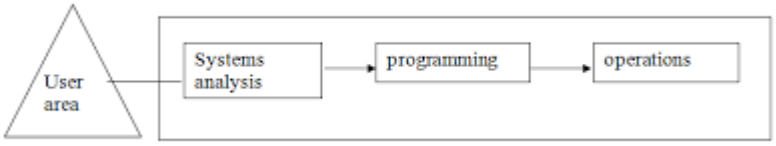
For example, in an organization there should be interaction between purchase department and production department, same way advertising with sales, payroll with personnel.

In 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 arithmetic unit uses for computation.

INTERDEPENDENCE

This is one of the important characteristics of a system.

Interdependence means the parts or the components of an organization or computer system depends on one another. Each component or parts should depend on other components of an organization. One component or subsystem depends on the input of another subsystem for proper functioning, ie, the output of one subsystem is required input for another subsystem. For example: - A decision to computerize an application is initiated by the user, analyzed and designed by the analyst, programmed and tested by the computer operator. In the below figure:- none of these persons can perform properly without the required input from others in the computer center subsystem.



INTEGRATION

Integration is concerned with how system components are connected together. It means that the parts of the system work together within the system even if each part performs a unique function

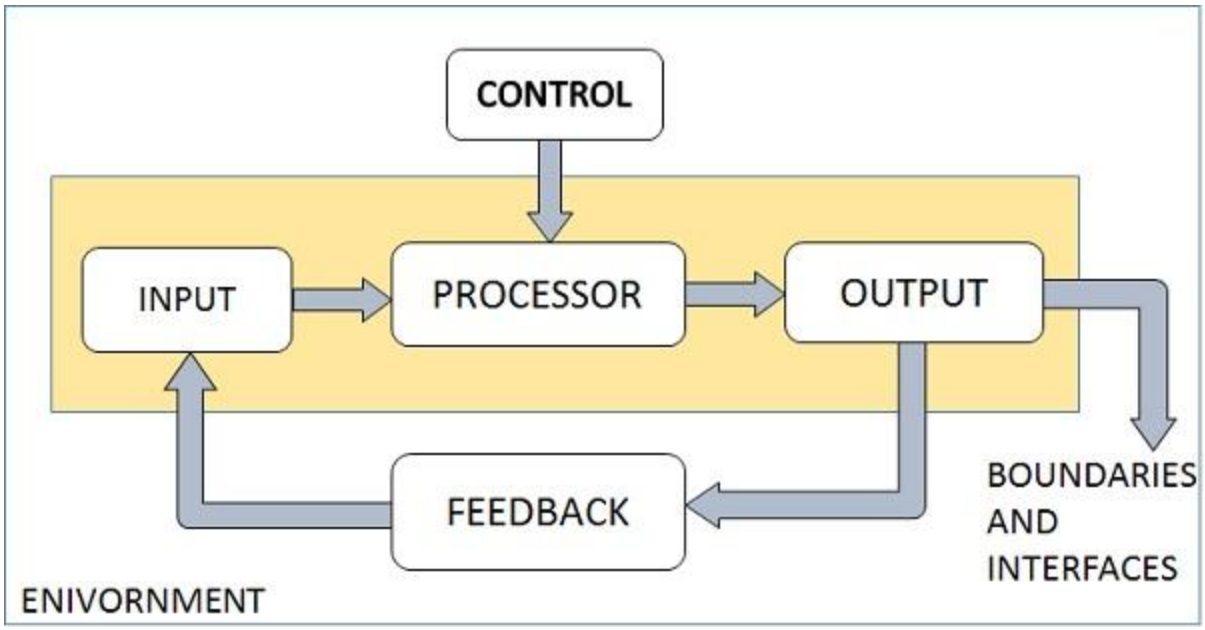
CENTRAL OBJECTIVE

The objective of system must be central. It may be real or stated. It is not uncommon for an organization to state an objective and operate to achieve another.

The users must know the main objective of a computer application early in the analysis for a successful design and conversion.

Elements of the System

A system has three basic elements input, processing and output. The other elements include control, feedback, boundaries, environment and interfaces.



INPUTS AND OUTPUTS

Inputs are the information or elements that we enter the system for processing. Output is the outcome of processing. A major objective of a system is to produce an output that has value to its user. Whatever the nature of the output it must be in line with the expectations of the intended user. A system feeds on input to produce output.

PROCESSOR

The processor is the element of a system that involves the actual transformations of input into output. It is the operational component of a system. Processors may modify the input totally or partially, depending on the specifications of the output.

CONTROL

The control element guides the system. It is the decision-making subsystem that controls the pattern of activities governing input, processing and output. In an organization context, management as a decision making body controls the inflow, handling and outflow of activities that affect the welfare of the business. In a computer system, the operating system and accompanying software influence the behavior of the system. Output specifications determine what and how much input is needed to keep the system in balance.

FEEDBACK

Control in a dynamic system is achieved by feedback. Feedback measures output against a standard procedure that includes communication and control. After the output is compared against performance standards, changes can result in the input or processing and consequently the output.

- Feedback provides the control in a dynamic system.
- Positive feedback is routine in nature that encourages the performance of the system.
- Negative feedback is informational in nature that provides the controller with information for action.

ENVIRONMENT

- The environment is the "super system" within which an organization operates.
- It is the source of external elements that strike on the system.
- It determines how a system must function. For example, vendors and competitors of organization's environment may provide constraints that affect the actual performance of the business.

BOUNDARIES AND INTERFACE

- A system should be defined by its boundaries. Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.
- Each system has boundaries that determine its sphere of influence and control.
- The knowledge of the boundaries of a given system is crucial in determining the nature of its interface with other systems for successful design.

Types of Systems

The systems can be divided into the following types −

Physical or Abstract Systems

- Physical systems are tangible entities. We can touch and feel them.
- Physical System may be static or dynamic in nature. For example, desks and chairs are the physical parts of computer center which are static. A programmed computer is a dynamic system in which programs, data, and applications can change according to the user's needs.
- Abstract systems are non-physical entities or conceptual that may be formulas, representation or model of a real system.

Open or Closed Systems

- An open system must interact with its environment. It receives inputs from and delivers outputs to the outside of the system. For example, an information system which must adapt to the changing environmental conditions.
- A closed system does not interact with its environment. It is isolated from environmental influences. A completely closed system is rare in reality.

Adaptive and Non Adaptive System

- Adaptive System responds to the change in the environment in a way to improve their performance and to survive. For example, human beings, animals.
- Non Adaptive System is the system which does not respond to the environment. For example, machines.

Permanent or Temporary System

- Permanent System persists for long time. For example, business policies.
- Temporary System is made for specified time and after that they are demolished. For example, A Data Processing system is set up for a program and it is dissembled after the program.

Natural and Manufactured System

- Natural systems are created by the nature. For example, solar system, seasonal system.
- Manufactured System is the man-made system. For example, Rockets, dams, trains.

Deterministic or Probabilistic System

- Deterministic system operates in a predictable manner and the interaction between system components is known with certainty. For example, two molecules of hydrogen and one molecule of oxygen make water.
- Probabilistic System shows uncertain behavior. The exact output is not known. For example, Weather forecasting, mail delivery.

Social, Human-Machine, Machine System

- Social System is made up of people. For example, social clubs, societies.
- In Human-Machine System, both human and machines are involved to perform a particular task. For example, Computer programming.
- Machine System is where human interference is neglected. All the tasks are performed by the machine. For example, an autonomous robot.

Man–Made Information Systems

- It is an interconnected set of information resources to manage data for particular organization, under Direct Management Control (DMC).
- This system includes hardware, software, communication, data, and application for producing information according to the need of an organization.

Man-made information systems are divided into three types −

- **Formal Information System** − It is based on the flow of information in the form of memos, instructions, etc., from top level to lower levels of management.
- **Informal Information System** – It is employee based system which solves the day to day work related problems.
- **Computer Based System** − This system is directly dependent on the computer for managing business applications. For example, automatic library system, railway reservation system, banking system, etc.

Systems Models

Schematic Models

- A schematic model is a 2-D chart that shows system elements and their linkages.
- Different arrows are used to show information flow, material flow, and information feedback.

Flow System Models

- A flow system model shows the orderly flow of the material, energy, and information that hold the system together.
- Program Evaluation and Review Technique (PERT), for example, is used to abstract a real world system in model form.

Static System Models

- They represent one pair of relationships such as activity–time or cost–quantity.
- The Gantt chart, for example, gives a static picture of an activity-time relationship.

Dynamic System Models

- Business organizations are dynamic systems. A dynamic model approximates the type of organization or application that analysts deal with.
- It shows an ongoing, constantly changing status of the system. It consists of −
    - Inputs that enter the system
    - The processor through which transformation takes place

- The program(s) required for processing
- The output(s) that result from processing.

Categories of Information

There are three categories of information related to managerial levels and the decision managers make.

Strategic Information

- This information is required by topmost management for long range planning policies for next few years. For example, trends in revenues, financial investment, and human resources, and population growth.
- This type of information is achieved with the aid of Decision Support System (DSS).

Managerial Information

- This type of Information is required by middle management for short and intermediate range planning which is in terms of months. For example, sales analysis, cash flow projection, and annual financial statements.
- It is achieved with the aid of Management Information Systems (MIS).

Operational information

- This type of information is required by low management for daily and short term planning to enforce day-to-day operational activities. For example, keeping employee attendance records, overdue purchase orders, and current stocks available.
- It is achieved with the aid of Data Processing Systems (DPS).

Introduction to System Analysis And Design

Computers have become an integral part of our life. One cannot imagine life without computers in today's world. You go to a bank to find balance in your account; you go to railway station for reservation, you will find computers at all place. Since computers are being used in every possible field today, it is necessary to understand and build these computerized systems in an effective way.

A system may be defined as a set of procedures established or formulated to carry out a specific activity to solve a problem. It is a set of interrelated elements forming a complete structure to perform a specific task. System, analysis and design is a methodology applied in computer system to develop a new system or to enhance a given system which can solve a given problem, for example, developing software to process salary of employees. System analysis refers to understanding the present system, future requirement and defining solution to meet the user's requirement within constraints and timeframe. Designing deals with the

final solution approach to a given problem. It aims at telling how the input will flow to give a specific output. Thus, system analysis and design is a whole process of analyzing a given system, defining problems, defining the system specification, providing best feasible solution to a given problem and maintenance of the proposed system with day to day future developments.

In short, System Analysis and Design (SAD), mainly deals with the software development activities. Success of any system depends on good SAD.

SAD, as performed by the system analysts, seeks to understand what human need to analyze data input or data flow systematically, process information in the context of a particular business. Furthermore, system analysis and design is used to analyze, design and implements in the support of users and the functioning of business that can be accomplished through the use of computerized information system.

Installing a system without proper planning leads to great user dissatisfaction and frequently causes the system to fall into disuse. System analysis and design lends structure to the analysis and design of information systems, a costly endeavor that might otherwise have been done in a haphazard way. It can be thought of as a series of processes systematically undertaken to improve a business through the use of computerized information system. SAD involves working with current and eventual users of information system to support them in working with technologies in an organizational setting.

System Analysis

System analysis may be understood as a process of collecting and interpreting facts, identifying problems and using the information recommend improvements in the system. In other words system analysis means identification, understanding and examine the system for achieving predetermined goals/objectives of the system. System analysis is carried out with the following two objectives:-

1. To know how a system currently operates and
2. To identify the users requirements in the proposed system

Basically, system analysis is a detailed study of all important business aspects under consideration and the existing system, and thus, the study becomes a basis for the proposed system (may be a modified or an altogether new system). System analysis is regarded as a logical process. The emphasis in this phase is an investigation to know how the system is currently operating and to determine what must be done to solve the problem.

The system analysis phase is very important in the total development efforts of a system. The user may be aware of the problem but may not know how to solve it. During system analysis, the developer (system designer) works with the user to develop a logical model of the system. A system analyst, because of his technical background, may move too quickly to program design to make the system prematurely physical, which is not desirable and may affect the ultimate success of the system. In order to avoid this, the system analyst must

involve the user at this stage to get complex information about the system. This can be achieved if a logical model of the system is developed on the basis of detailed study. Such a study (analysis) should be done by using various modern tools and techniques, such as data flow diagrams, data dictionary, and rough description of the relevant algorithm the final requirement of proposed information system.

System analysis is a process of collecting factual data, understanding the process involved, identifying problems and recommending feasible suggestion for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weakness of the system so as to achieve the organizational goals. System analysis also includes subdividing of complex process involving the entire system, identification of data store and manual process.

The major objectives of system analysis are to find answers for each business process: what is being done how is it being done, who is doing it, when is he doing it, why is it being done and how can it be improved? It is more of a thinking process and involves the creative skills of the system analyst. It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is a logical system design. System analysis is an interactive process that continues until a preferred and acceptable solution emerges.

System Design

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the development of a system. The logical system design arrived at as a result of system analysis is converted into physical system design. Normally, the design proceeds in two stages:

PRELIMINARY OR GENERAL DESIGN

In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible (possible), we move to the detailed design stage.

STRUCTURED OR DETAILED DESIGN

In the detailed design stage, computer oriented work begins in earnest. At this stage the design of the system becomes more structured. Structure design is a blue print of a computer system solution to a given problem having the same components and inter-relationships among the same components as the original problem. Input, output, databases, forms, codifications schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decide.

The system design involves:-

- Defining precisely the required system output

- Determining the data requirement for producing the output
- Determining the medium and format of files and databases
- Devising processing methods and use of software to produce output
- Determine the methods of data capture data input
- Designing input forms
- Designing codification scheme
- Detailed manual procedures
- Documenting the design

System Development Life Cycle

An effective System Development Life Cycle (SDLC) should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.

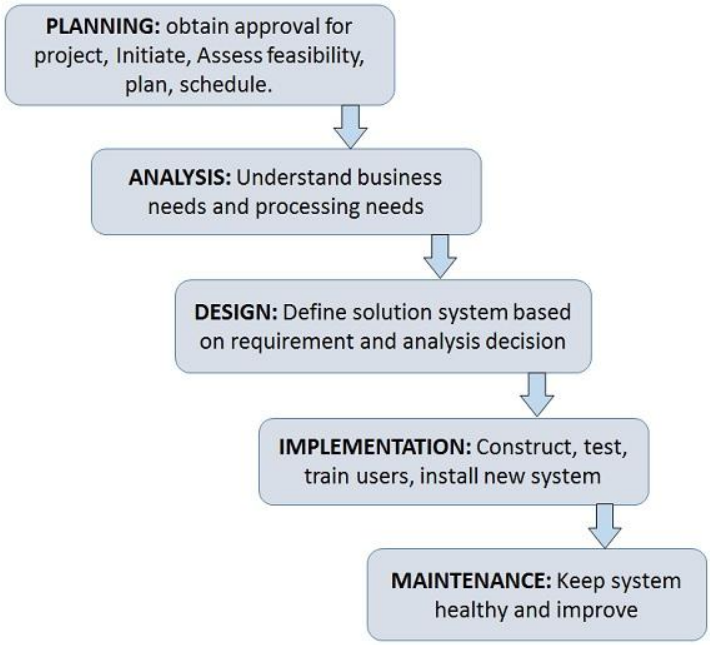
System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

SDLC is used by analysts to develop an information system. SDLC includes the following activities −

- requirements
- design
- implementation
- testing
- deployment
- operations
- maintenance

Phases of SDLC

Systems Development Life Cycle is a systematic approach which explicitly breaks down the work into phases that are required to implement either new or modified Information System.



Feasibility Study or Planning

- Define the problem and scope of existing system.
- Overview the new system and determine its objectives.
- Confirm project feasibility and produce the project Schedule.
- During this phase, threats, constraints, integration and security of system are also considered.
- A feasibility report for the entire project is created at the end of this phase.

Analysis and Specification

- Gather, analyze, and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.
- A Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system, is to be prepared at the end of this phase.

System Design

- Includes the design of application, network, databases, user interfaces, and system interfaces.
- Transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented in a programming language.

- Create a contingency, training, maintenance, and operation plan.
- Review the proposed design. Ensure that the final design must meet the requirements stated in SRS document.
- Finally, prepare a design document which will be used during next phases.

Implementation

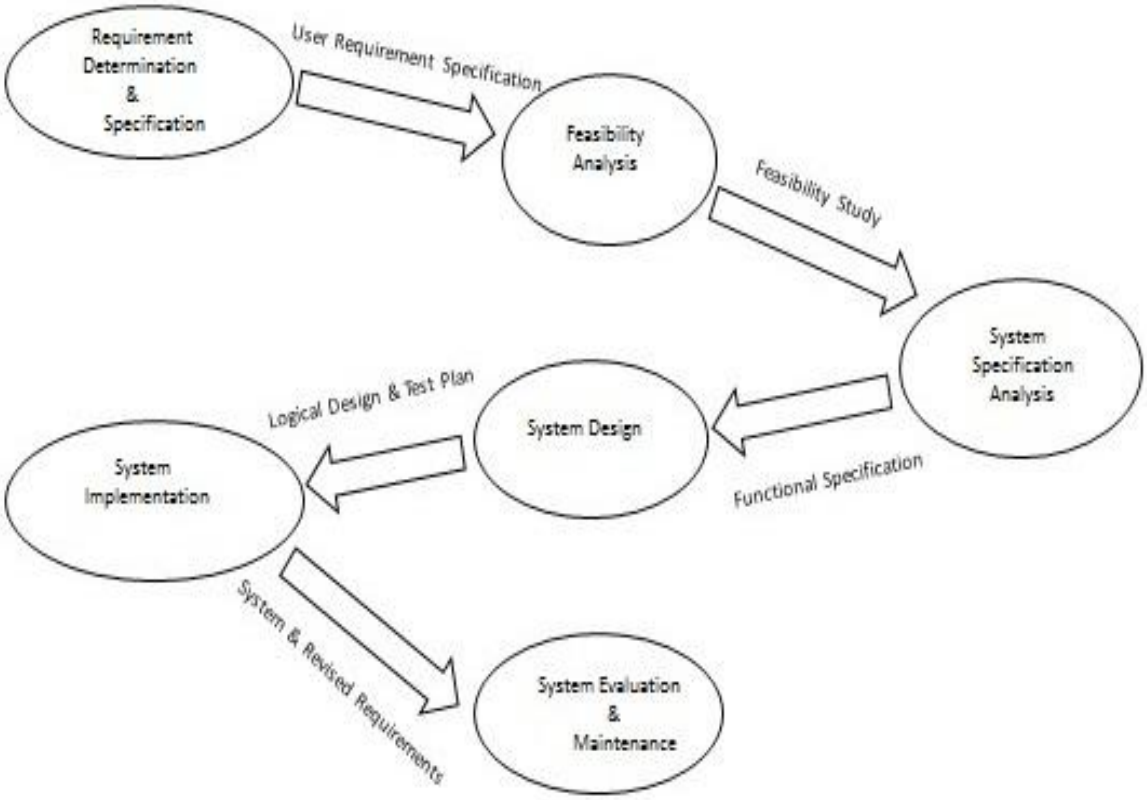
- Implement the design into source code through coding.
- Combine all the modules together into training environment that detects errors and defects.
- A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria, and resource allocation for testing.
- Integrate the information system into its environment and install the new system.

Maintenance/Support

- Include all the activities such as phone support or physical on-site support for users that is required once the system is installing.
- Implement the changes that software might undergo over a period of time, or implement any new requirements after the software is deployed at the customer location.
- It also includes handling the residual errors and resolves any issues that may exist in the system even after the testing phase.
- Maintenance and support may be needed for a longer time for large systems and for a short time for smaller systems.

Life Cycle of System Analysis and Design

The following diagram shows the complete life cycle of the system during analysis and design phase.


Requirement Determination & Specification
User Requirement Specification
Feasibility Analysis
Feasibility Study
System Specification Analysis
Functional Specification
System Design
Logical Design & Test Plan
System Implementation
System & Revised Requirements
System Evaluation & Maintenance

Unit II

Introduction

Where does System Analyst come from?

A **systems analyst** is an information technology (IT) professional who specializes in analyzing, designing and implementing information systems. Systems analysts assess the suitability of information systems in terms of their intended outcomes and interact with end users, software vendors and programmers in order to achieve these outcomes. A systems analyst is a person who uses analysis and design techniques to solve business problems using information technology. Systems analysts may serve as change agents who identify the organizational improvements needed, design systems to implement those changes, and train and motivate others to use the systems

Although they may be familiar with a variety of programming languages, operating systems, and computer hardware platforms, they do not normally involve themselves in the actual hardware or software development. They may be responsible for developing cost analysis, design considerations, staff impact amelioration, and implementation timelines.

A systems analyst is typically confined to an assigned or given system and will often work in conjunction with a business analyst. These roles, although having some overlap, are not the same. A business analyst will evaluate the business need and identify the appropriate solution and, to some degree, design a solution without diving too deep into its technical components, relying instead on a systems analyst to do so. A systems analyst will often evaluate code, review scripting and, possibly, even modify such to some extent.

Some dedicated professionals possess practical knowledge in both areas (business and systems analysis) and manage to successfully combine both of these occupations, effectively blurring the line between business analyst and systems analyst.

What does he do?

A systems analyst may:

- Identify, understand and plan for organisational and human impacts of planned systems, and ensure that new technical requirements are properly integrated with existing processes and skill sets.
- Plan a system flow from the ground up.
- Interact with internal users and customers to learn and document requirements that are then used to produce business requirements documents.
- Write technical requirements from a critical phase.
- Interact with software architect to understand software limitations.
- Help programmers during system development, e.g. provide data flow diagrams.
- Document requirements or contribute to user manuals.

- Whenever a development process is conducted, the system analyst is responsible for designing components and providing that information to the developer.

Preparing for case as a system analyst general business knowledge

Required Skills for an Analyst

An analyst must possess various skills to effectively carry out the job. Mainly the skills can be divided into two types.

1. Communication Skill
2. Technical Skill

Both are required for system development.

Communication Skill

Communication skills deal with relationships and the interface of the analyst with people in business. They are useful in establishing trust, resolving conflict and communicating information. Analyst must have the ability to articulate and speak the language of the user and a knack for working with virtually all managerial levels in the organization. Communication is not just reports, telephone conversations and interviews. It is people talking, listening, feeling and reacting to one another, their experience and reactions. He must be capable of identifying problems and assessing their ramifications having a grasp of company goals and objectives and showing sensitivity to the impact of the system on people at work.

Technical skills

Technical skills focus on procedures and techniques for operations analysis, system analysis and computer science.

Technical skills include:

Creativity-helping users model ideas into concrete plans and developing candidate systems to match user requirements.

Problem solving- reducing problems to their elemental levels for analysis, developing alternative solutions to a given problem and delineating the pros and cons of candidate systems.

Project management- includes scheduling, performing well under time constraints, coordinating team efforts and managing costs and expenditures.

Dynamic interface- blending technical and nontechnical considerations in functional specifications and general design

Questioning attitude and inquiring mind-knowing the what, when, why, where, who, and, how a system works.

Knowledge of the basics of the computer and the business function

System analysts require interpersonal as well as the technical skills, although the necessity for both skills depends on the stages of system development.

During analysis there is greater need for interpersonal skills - working with the user to determine requirements and translate them into design criteria.

During design, the major thrust is to develop a detailed design of the candidate system - highly technical procedures and methodologies. Even then there is some emphasis on the interpersonal factor-the analyst user interface and user participation as a step toward training and implementation.

During program construction, coding and testing it has to be carried out with some user participation.

During system implementation, technical and interpersonal skills converge. The technical aspects focus on "proving" the software and preparing for the final conversion of files and documentation. The interpersonal aspects deal with user training and selling the user on the benefits and potential of the candidate system.

During the maintenance stage the role of the analyst drops off, except when unanticipated problems develop.

Role of the System Analyst

The multifaceted roles of analyst are change agent, monitor, architect, psychologist, salesperson, motivator and politician.

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. It is important, that the user accept change. Analyst can secure user acceptance is through user participation during design and implementation.

In the role of a change agent, the systems analyst may select various styles to introduce change to the user organization. The styles range from that of persuader (the mildest form of intervention) to imposer (the most severe intervention). In between there are the catalyst and the confronter roles. When the user appears to have a tolerance for change the persuader or catalyst style is appropriate. On the other hand, when drastic changes are required, it may be necessary to adopt the confronter or even the imposer style. No matter what style is used, the goal is same: to achieve acceptance of the candidate system with a minimum of resistance.

Investigator and monitor

In defining a problem, the analyst will collect and put together all the information to determine why the present system does not work well and what changes will correct the problem. This work is similar to that of an investigator- extracting the real problems from existing systems and creating information structures that uncover previously unknown trends that may have a direct impact on the organization.

Related to the role of investigator is that of monitor. To undertake and successfully complete a project, the analyst must monitor programs in relation to time, cost, and quantity. Of these resources, time is the most important. If time "gets away", the project suffers from increased costs and wasted human resources. Implementation will also get delayed.

Architect

As architect an analyst must create detailed physical design of candidate system. He aids users in formalizing abstract ideas and provides details to build the end product-the candidate system.

Psychologist

The analyst plays the role of a psychologist in the way he reaches people interprets their thoughts, assesses their behavior, and draws conclusions from these interactions. Understanding inter functional relationships is important. It must be aware of people's feelings and be prepared to get around things in a graceful way. The art of listening is important in evaluating responses and feedback.

Salesperson

Selling change can be crucial as initiating change. Selling the system actually takes place at each step in the system life cycle. Sales skills and persuasiveness are crucial to the success of the system.

Motivator

A candidate system must be well designed and acceptable to the user. The analyst role as a motivator becomes obvious during the first few weeks after implementation and during times when turn over results in new people being trained to work with the candidate system. The amount of dedication it takes to motivate users often taxes the analyst's abilities to maintain the pace.

Politician

In implementing a candidate system, the analyst tries to appease all parties involves. Diplomacy and finesse in dealing with people can improve acceptance of the system. In as much as a politician must have the support of his or her constituency, so is the analyst's goal

to have the support of the users' staff. He or she represents their thinking and tries to achieve their goals through computerization.

In summary these multiple roles require analysts to be orderly, approach a problem in a logical, methodical way and pay attention to details.

MIS and System Analysis

System analysis plays central role in the development of the MIS. Since the MIS is a corporation of the various systems, a systematic approach in its development helps in achieving the objective of the MIS. Each system within the MIS plays a role which contributes to the accomplishment of the MIS objective.

The tools of the system analysis and the method of development enforce a discipline on the designer to follow the steps strictly as stipulated. The possibility of a mistake or an inadvertence is almost ruled out. The system analysis with its structural analysis and design approach ensures an appropriate coverage of the subsystems. The data entities and attributes are considered completely keeping in view the needs of the systems in question and their interface with other systems.

The systems analysis begins with the output design which itself ensures that the information needs are considered and displayed in the appropriate report or screen format; the subsequent design steps are taken to fulfill these needs.

The MIS may call for an open system design. In such a case while making the systems analysis and design, the aspect of open system design is considered and necessary modification are introduced in the designed the information system.

The user's application in the system development ensures the attention to the smaller details in the design. The users actively come out with their requirements automatically ensuring that the users are met more precisely.

The systems analysis and designs, as a tool of the MIS development, helps in streamlining the procedures of the company to the current needs of the business and information objectives. New transactions, new documents, new procedures are brought in to make the system more efficient before it is designed.

The SAD exercise considers testing the feasibility of the system as an important step. This step, many a times, saves the implementation of inefficient systems. Sometimes it forces the management and analysts to look into the requirement and its genuineness. The MIS development process largely relies on the systems analysis and design as a source of the scientific development of the MIS.

The development of the MIS in today advance information technology and internet, web environment is a challenge. The nature of the system analysis has undergone a change, while the core process of the analysis and development has remained the same.

The system analysis is not restricted to the data-process-output. It also covers the technologies which enables the process feasible. The subject now covers the analysis of interfacing and supports the technologies and it's fitting into a chosen hardware-software platforms for a core system development. The MIS largely depends on how these technologies are bladed with the main system. The system architecture of the MIS is now different due to the high tech involvement of the data capture, communication, and processing technologies. The trend is towards more swift data capture and making it available in the fastest possible time leaving its usage to the user.

The development methodology may be the predictable design of data, databases and file approach or object oriented analysis and design approach. The MIS design is same, the difference is in the development cycle time, quality of information efficiency of design and the case of maintenance of the system.

WHAT IS BUSINESS ANALYST

BA refers to any person who is responsible for performing the business analysis functions for IT system development projects such as analysing business needs, facilitating the elicitation of user requirements, documenting and prioritising the business requirements, verifying the major project deliverables, business reengineering opportunities and workflow from business perspective, and facilitating effective communication between business and IT sides.

IMPORTANCE OF AND NEED FOR A Business Analyst ROLE

(a) During IT system development, communication gap often exists between IT staff and business users due to differences in knowledge, skills, background and orientation. Users may not understand the IT terminology and technical solutions while IT staff may not understand the business terminology, functions, processes and environment, leading to difficulties in eliciting real business needs and understanding of requirements as well as affecting the design of the proposed system. The situation becomes even more challenging if the IT project is outsourced, where more communication and collaboration issues may arise especially when the external IT contractor is not familiar with the Government environment and the business processes. Therefore, a BA role is important and needed to be instituted in the IT project organisation to improve the collaboration between users and IT staff throughout the SDLC.

(b) At project initiation stage before the formation of a project team, BA can help explore improvement opportunities of current state by developing sound business cases to justify the investment of IT project and produce a clear project scope and estimation. BA role is especially helpful in scoping and planning of large-scale projects at project initiation stage.

(c) Where the demand and resources justify, a permanent establishment of the BA role is recommended to aid in future system maintenance, support and enhancement.

BENEFITS OF HAVING DEDICATED BA

BA serves as the bridge between the business users and the technical IT people. Its presence will contribute significantly to the success of IT projects. The anticipated benefits of having a dedicated BA include the following:

i) More able to deliver a clear project scope from a business point of view;

ii) More able to develop sound business cases and more realistic estimation of resources and business benefits;

iii) More able to make a better project scoping, planning and management in costs and schedule especially for large-scale IT projects;

iv) More able to produce clear and concise requirements, which in turn provide clearer and more accurate tender requirements if the project is outsourced.

v) More able to elicit the real business needs from users, and effectively manage user expectations and changes;

vi) More able to improve the quality of design for the proposed IT system so that it is able to meet real user needs and achieve the anticipated benefits;

vii) More able to ensure the quality of the system developed before passing on to end-users for review and acceptance; and

viii) More competent to arrange the comprehensive and quality test on the delivered systems or functions and provide feedback to the technical IT people.

REQUIREMENT DETERMINATION

It is also termed as a part of software requirement specification (SRS); it is the starting point of the system development activity. This activity is considered as the most difficult and also the most error prone activity because of the communication gap between the user and the developer. This may be because the user usually does not understand the users problem and application area. The requirement determination is a means of translating the ideas given by the user, into a formal document, and thus to bridge the communication gap. A good SRS provides the following benefits:-

- It bridges the communication gap between the user and the developer by acting as a basis of agreement between the two parties.
- It reduces the development cost by overcoming errors and misunderstandings early in the development.
- It becomes a basis of reference for validation of the final product and thus acts as a benchmark.

Requirement determination consists of three activities namely **requirement anticipation, requirement investigation and requirement specification**. A requirement anticipation activity includes the past experience of the analysis, when influence the study. They may

force the likelihood of certain problems or features and requirements for a new system. Thus, the background of the analysts to know what to ask or which aspects to investigate can be useful in at the system investigation. Requirement investigation is at the centre of system analysis. In this, the existing system is studied and documented for further analysis. Various methods like fact-finding techniques are used for the investigation are analyzed to determine requirement specification, which is the description of the features for a proposed system.

Requirement determination, in fact, is to learn and collect the information about:-

✓ The basic process

✓ The data which is used or produced during the process

✓ The various constraints in terms of time and volume of work and

✓ The performance controls used in the system.

UNDERSTAND THE PROCESS

Process understanding can be acquired, if the information is collected regarding:-

✓ The purpose of the business activity

✓ The steps which and where they are performed

✓ The persons performing them, and

✓ The frequency, time and user of the resulting information

Identify data used and information generated

Next to process understanding, an information analyst should find out what data is used to perform each activity.

Determine frequency, timing and volume.

Information should also be collected to know how often the activity is repeated and volume of items to be handled. Similarly, timing does affect the way analysts evaluate certain steps in carrying out an activity, in other words, timing, frequency and volume of activities are important facts to collect.

Know the performance controls

System controls enable analysts to understand how business functions can be maintained in an acceptable manner.

In order to understand the business operations of the organizations and thus to know the existing system and information requirement for the new system and information analyst collects the information and then makes an analysis of the collected information by using certain analysis tools.

Strategies for Requirement Determination

In order to collect information so as to study the existing system and to determine information requirement, there are different strategies, which could be used for the purpose. These strategies are discussed below:-

Interview: the interview is a face-to-face method used for collecting the required data. In this method, a person (the interviewer) asks question from the other person being interviewed. The interview may be formal or informal and the question asked may be structured or unstructured. The interview is the oldest and the most often used device for gathering information about an existing system.

Because of time required for interviewing and the inability of the users to explain the system in detail, other methods are also used to gather information. Interviewing is regarded as an art and it is important that analysts must be trained in the art of successful interviewing. This is also important because of the fact that the success of an interviewer and on his or her preparation for interview.

Questionnaire: a questionnaire is a term used for almost any tool that has questions to which individual respond. The use of questionnaires allows analysts to collects information about various aspects of a system from a large number of persons. The questionnaire may give more reliable data than other fact-finding techniques. Also the wide distribution ensures greater uncertainty for responses. The questionnaire survey also helps in saving time as compared to interviews. The analysts should know the advantages and disadvantages of structured as well as unstructured questionnaires must be tested and modified as per the background and experience of the respondents.

Record review: record review is also known as review of documentation. Its main purpose is to establish quantitative information regarding volumes, frequencies, trends, ratios, etc. in record review; analysts examine information that has been recorded about the system and its users. Records/documents may include written policy manuals, regulations and standard operating procedures used by the organization as a guide for managers and other employees. Procedures, manuals and forms are useful sources for the analysts to study the existing system.

Observation: another information gathering tool used in system studies is observation. It is the process of recognizing and noticing people, objects and occurrences to obtain information. Observation allows analysis to get information. This is difficult to obtain by any other fact finding method. This approach is most useful when analysts need to actually observe the way documents are handled, Processes are carried out and weather specific steps are actually followed. As an observer, the analyst follows a set of rules. While making observations, he/she is more likely to listen than talk.

The analysis usually use a combination of all these approached to study an existing system as any one approach may not be sufficient for electing information requirement of the system.

SUMMARY

System analysis is a detailed study of all important business aspects of a future system, as well as existing system. Thus, the study becomes a basis for a proposed system. In this process of system analysis, emphasis is placed on 'what must be done to solve the problem'. The final product of system analysis is a set of system requirement of a proposed information system. Requirement determination, which is an important activity of system analysis, is a means of translating the ideas given by the users into a formal document. System analysis ensures that the system analyst understands the user's requirements in a clear way and thus reduces the communication gap between the user and the developer. It reduces the development cost by overcoming errors and misunderstandings early in the development and becomes a basis for reference for validation of the final product.

In order to study the existing system and to determine information requirements, there are several strategies which could be used for the purpose. These strategies may include interviews, questionnaires, record reviews and observation. As any one may not be sufficient for electing information requirements of the system, the analysis usually use a combination of all these strategies.

System analysis is carried out with the help of certain tools. The main tools, which are used for analyzing and documenting the system specification, are data flow diagram, data dictionary, structured English, decision trees and decisions tables.

The main objectives of the system design are to produce system specifications which can then be converted into an information system for use in the organization. However, the system design is a creative activity and is considered to evolve through two different levels of design, i.e. conceptual and detailed design. The conceptual design which is also called feasibility design sets the direction for the MIS project and provides performance requirements. The output of the conceptual design i.e. performance specifications are taken as an input to the detailed design to produce system specifications. The system specifications thus generated are handled over to the computer programmer for translating into a physical information system.

The system specifications, called detailed system design or logical system design provide all details of inputs, outputs, file, database controls and procedures. For ensuring an effective, efficient and successful MIS, the system analysts must not rush through this phase, rather each and every step must be undertaken very carefully to prepare a detailed system design.

Unit 3

Problems with Development Life Cycle Approach: Need for Structured Approach

The system development life cycle (SDLC) is the traditional system development method that organizations use for large-scale IT Projects. The SDLC is a structured framework that consists of sequential processes by which an information system is developed.

1. System Investigation
2. System Analysis
3. System Design
4. Programming
5. Testing
6. Implementation
7. Operation and Maintenance

Once a development project has the necessary approvals from all participants, the systems analysis stage begins. System analysis is the examination of the business problem that organizations plan to solve with an information system. The main purpose of the systems analysis stage is to gather information about the existing system in order to determine the requirements for an enhanced system or a new system. The end product of this stage, known as the deliverable, is a set of system requirements.

Perhaps the most difficult task in system analysis is identifying the specific requirements that the system must satisfy. These requirements often are called user requirements because users provide them. When the system developers have accumulated the user requirements for the new system, they proceed to the system design stage.

Considerations for Candidate System

In today's business there is more demand for computer services than there are resources available to meet the demand. The demand is made up of the following.

1. Operations of existing system.
2. Maintenance that focuses on "patching "programs
3. Enhancement that needs major modifications in program structure or equipments.
4. Requests for candidate system

All these demands require resources-human, financial, and technological. On the human side the computer department has to provide the following.

1. Computer operators to run equipment.
2. Data entry personnel.
3. System Analyst to define and design specifications.
4. Application programmers to convert System specifications to Computer programs.

5. Maintenance Programmers to debug errors.
6. Supervisors, project leaders and Managers to Co-ordinate the jobs with the users.

Thus, the basic problem is to match the demands for services with the available resources. How much one project is favored over another depends on technical, behavioral and Economic Factors.

The technical factor involves the system departments' ability to handle a project. It mostly depends on the availability of the qualified analyst, designers and software specialist to do the work.

The behavioral factors involves the users past experience with the existing system, the success record of the analyst and the influence the user can exert on upper management to finance a candidate system.

The most important criteria in selecting a project is the economic factors. It focuses on the systems potential return on investment.

Political considerations

In conjunction with the preceding considerations is the political factor, which is partly behavioral. For eg-managers in a production firm are considering two office automation proposals: proposal A-a teleconferencing system designed to reduce travel cost, and proposal B- a sales support system. Proposal –A is justified by hard figures, but it was turned down. Instead proposal-B was sponsored by an influential executive and had the support of the committee. It passed because the right people were convinced it should.

Planning and control for system success

For the success of a system the analyst role is very important. First, a plan must be devised, detailing the procedure, some methodology, activities, resources, costs and timetable for completing the system.

Second, in larger projects, a project team must be formed of analysts, programmers, system consultant and user representatives. Shared knowledge, interaction and the coordination realized through team effort can be extremely effective in contrast with individual analysts doing the same work.

Finally the project should be divided into manageable modules to reflect the phases of system development-analysis, design and implementation

Prototyping

The two major problems with building information systems are

1) The system development life cycle is too long and

2) The right system rarely developed at the first time.

Lengthy development frustrates the user. Analysts seem to get bogged down with tedious methodologies for developing systems. An alternative to this "paralysis by analysis" is an advanced technique called prototyping. The basic steps are:

1. Identify user information and operating requirements.
2. Develop a working prototype that focuses on only the most important function, using a basic database.
3. Allow the user to use the prototype, discuss requested changes and implement the most important changes.

4. Repeat the next version of prototype with further changes incorporated until the system fully meets user requirements.

What is Requirements Determination?

A requirement is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management.

Requirements determination involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made.

Major Activities in Requirement Determination

Requirements Anticipation

- It predicts the characteristics of system based on previous experience which include certain problems or features and requirements for a new system.
- It can lead to analysis of areas that would otherwise go unnoticed by inexperienced analyst. But if shortcuts are taken and bias is introduced in conducting the investigation, then requirement Anticipation can be half-baked.

Requirements Investigation

- It is studying the current system and documenting its features for further analysis.
- It is at the heart of system analysis where analyst documenting and describing system features using fact-finding techniques, prototyping, and computer assisted tools.

Requirements Specifications

- It includes the analysis of data which determine the requirement specification, description of features for new system, and specifying what information requirements will be provided.

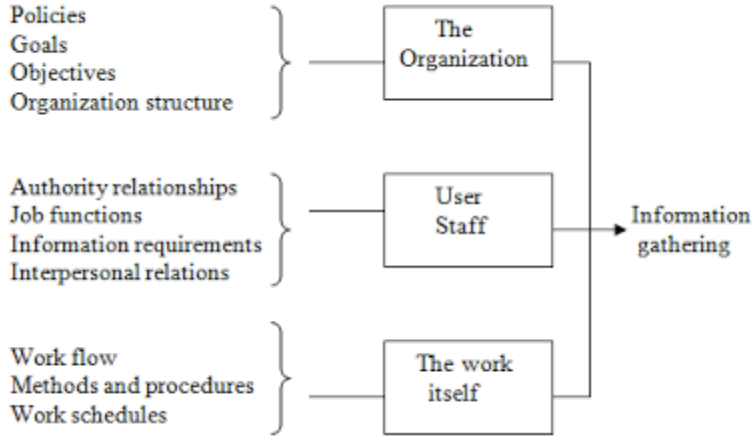
- It includes analysis of factual data, identification of essential requirements, and selection of Requirement-fulfillment strategies.

Information Gathering: A problem Solving Approach

Information gathering is an art and a science. The approach and manner, in which information is gathered, require persons with sensitivity, common sense and knowledge of what and when to gather and the channels used to secure information.

KINDS OF INFORMATION REQUIRED

Before one determines where to go for information or what tools to use, the first requirement is to figure out what information to gather. Much of the information we need to analyze relates to the organization in general, the user staff, and the workflow.



Information about the Organization

Information about the organization's policies, goals, objectives, and structure explains the kind of environment that promotes the introduction of computer-based systems. Company policies are guidelines that determine the conduct of business. Policies are translated into rules and procedures for achieving goals. A statement of goals describes management's commitment to objectives and the direction system development will follow. Objectives are milestones of accomplishments toward achieving goals. Information from manuals, pamphlets, annual reports etc help the analyst to get an idea of the goals of the Organization.

After policies and goals are set, a firm is organized to meet these goals. The organization structure indicates management directions and orientation. The organization chart represents

an achievement-oriented structure. It helps us understand the general climate in which candidate systems will be considered. In gathering information about the firm, the analyst should watch for the correspondence between what the organization claims to achieve goals and actual operations. Policies, goals, objectives and structure are important elements for analysis.

Information about the User Staff

Another kind of information for analysis is knowledge about the people who run the present system, their job functions and information requirements, the relationships of their jobs to the existing system and the interpersonal network that holds the user group together. The main focus is on the roles of the people, authority relationships and inters personnel relations. Information of this kind highlights the organization chart and establishes a basis for determining the importance of the existing system for the organization. Thus the major focus is to find out the expectations of the people before going in for the design of the candidate system.

Information about the Work Flow

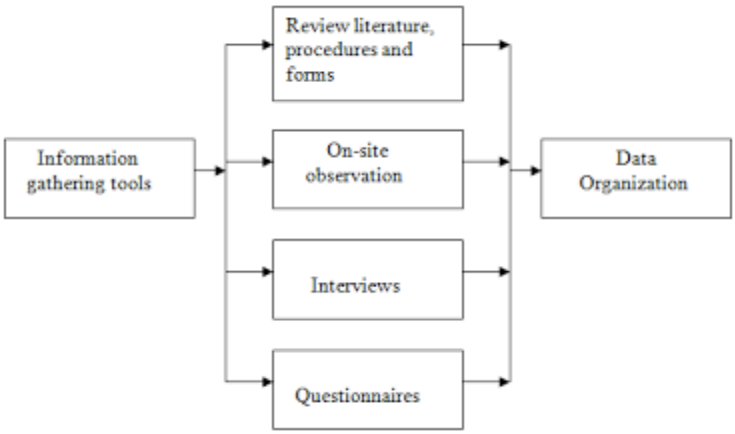
The workflow focuses on what happens to the data through various points in a system. This can be shown by a data flow diagram or a system flow chart.

A data flow diagram represents the information generated at each processing point in the system and the direction it takes from source to destination.

A system flowchart describes the physical system. The information available from such charts explains the procedures used for performing tasks and work schedules.

Information Gathering Techniques

No two projects are ever the same. This means that the analyst must decide on the information-gathering tool and how it must be used. Although there are no standard rules for specifying their use, an important rule is that information must be acquired accurately, methodically, under the right conditions, and with minimum interruption to user personnel. There are various information-gathering tools. Each tool has a special function depending on the information needed.



Review of Literature, Procedures and Forms

Review of existing records, procedures, and forms helps to seek insight into a system which describes the current system capabilities, its operations, or activities.

Advantages

- It helps user to gain some knowledge about the organization or operations by themselves before they impose upon others.
- It helps in documenting current operations within short span of time as the procedure manuals and forms describe the format and functions of present system.
- It can provide a clear understanding about the transactions that are handled in the organization, identifying input for processing, and evaluating performance.
- It can help an analyst to understand the system in terms of the operations that must be supported.
- It describes the problem, its affected parts, and the proposed solution.

Disadvantages:

- The primary drawback of this search is time.
- Sometimes it will be difficult to get certain reports.
- Publications may be expensive and the information may be out dated due to a time lag in publication.

On-site Observation

A fact-finding method used by the systems analyst is on-site or direct observation. It is the process of recognizing and noting people, objects and occurrences to obtain information. The major objective of on-site observation is to get as close as possible to the "real" system

being studied. For this reason it is important that the analyst is knowledgeable about the general makeup and activities of the system. The analyst's role is that of an information seeker. As an observer, the analyst follows a set of rules.

- While making observations he/she is more likely to listen than talk and has to listen with interest when information is passed on.
- The analyst has to observe the physical layout of the current system, the location and movement of the people and the workflow.
- The analyst has to be alert to the behavior of the user staff and the people to whom they come into contact. A change in behavior provides a clue to an experienced analyst. The clue can be used to identify the problem.

The following questions can serve as a guide for on-site observations:

1. What kind of system is it? What does it do?
2. Who runs the system? Who are the important people in it?
3. What is the history of the system? How did it get to its present stage of development?
4. Apart from its formal function, what kind of system is it in comparison with other systems in the organization?

Advantages

- It is a direct method for gathering information.
- It is useful in situation where authenticity of data collected is in question or when complexity of certain aspects of system prevents clear explanation by end-users.
- It produces more accurate and reliable data.
- It produces all the aspect of documentation that is incomplete and outdated.

Difficulties in on-site observations:

- On-site observation is the most difficult fact-finding technique. It requires intrusion into the user's area and can cause adverse reaction by the user's staff if not handled properly.
- If on-site observation is to be done properly in a complex situation, it will be time-consuming.
- Proper sampling procedures must be used to identify the stability of the behavior being observed. Otherwise inferences drawn from these samples will be inaccurate and unreliable.
- Attitudes and motivations of subjects cannot be easily observed.

- As an observer, the analyst follows a set of rules. While making observations, he /she should be more likely to listen than talk and to listen with a sympathetic and genuine interest when information is conveyed.

Four alternative observation methods are considered

1) Natural or contrived: A natural observation occurs in a setting such as the employee's place of work, whereas the observer in a place like a laboratory sets up is contrived observation.

2) Obtrusive or unobtrusive: An obtrusive observation takes place when the respondent knows he/she is being observed; an unobtrusive observation takes place in a contrived way such as behind a one-way mirror.

3) Direct or indirect: A direct observation takes place when the analyst actually observes the subject or the system at work. In an indirect observation, the analyst uses mechanical devices such as cameras and videotapes to capture information.

4) Structured or unstructured: In a structured observation, the observer looks for and records a specific action. Unstructured methods place the observer in a situation to observe whatever might be applicable at the time.

Any of these methods may be used in information gathering. Natural, direct, obtrusive and unstructured observations are frequently used to get an overview of an operation. Electronic observation and monitoring methods are becoming widely used tools for information gathering.

Interviews

On-site observation is less effective for learning about people's perceptions, feelings and motivations. The alternative is the personal interview and the questionnaire. In both the methods heavy reliance is placed on the interviewees report for information about the job, the present system, or experience. The quality of the response is judged in terms of its reliability and validity.

Reliability means that the information gathered is trustworthy enough to be used for making decisions about the system being studied. Validity means that the questions to be asked are worded in such a way as to elicit (obtain) the intended information. So the reliability and validity of the data collected depends on the design of the interview or questionnaire and the manner in which each instrument is administered.

The interview is a face-to-face interpersonal role situation in which a person called the interviewer asks a person being interviewed questions designed to gather information about a problem area. The interview is the oldest and most often used device for gathering information in system work. It can be used for two main purposes

- As an exploratory device to identify relations or verify information
- To capture information, as it exists.

Systems analyst collects information from individuals or groups by interviewing. The analyst can be formal, legalistic, play politics, or be informal; as the success of an interview depends on the skill of analyst as interviewer.

Advantages of Interviewing

- This method is frequently the best source of gathering qualitative information.
- It is useful for them, who do not communicate effectively in writing or who may not have the time to complete questionnaire.
- Information can easily be validated and cross checked immediately.
- It can handle the complex subjects.
- It is easy to discover key problem by seeking opinions.
- It bridges the gaps in the areas of misunderstandings and minimizes future problems.

The disadvantages of an interview are

- The major drawback of the interview is the long preparation time.
- Interview also takes a lot of time to conduct, which means time and money.

In an interview, since the analyst and the person interviewed meet face to face, there is an opportunity for greater flexibility in eliciting information. The interviewer is also in a natural position to observe the subjects and the situation to which they are responding. In contrast the information obtained through a questionnaire is limited to the written responses of the subjects to predefined questions.

The art of interviewing:

Interviewing is an art. The analyst learns the art by experience. The interviewer's art consists of creating a permissive situation in which the answers offered are reliable. Respondent's opinions are offered with no fear of being criticized by others. Primary requirements for a successful interview are to create a friendly atmosphere and to put the respondent at ease. Then the interview proceeds with asking questions properly, obtaining reliable responses and recording them accurately and completely.

Arranging the interview:

The interview should be arranged so that the physical location, time of the interview and order of interviewing assure privacy and minimal interruption. A common area that is non-threatening to the respondent is chosen. Appointments should be made well in advance and a fixed time period adhered to as closely as possible. Interview schedules generally begin at the top of the organization structure and work down so as not to offend anyone.

Guides to a successful interview:

In an interview the following steps should be taken.

1. Set the stage for the interview.
2. Establish rapport: put the interviewee at ease.
3. Phrase questions clearly and briefly
4. Be a good listener; avoid arguments.
5. Evaluate the outcome of the interview.

1. Stage setting: This is a relaxed, informal phase where the analyst opens the interview by focusing on

- The purpose of the interview
- Why the subject was selected
- The confidential nature of the interview.

After a favorable introduction, the analyst asks the first question and the respondent answers it and goes right through the interview. The job of the analyst should be that of a reporter rather than a debater. Discouraging distracting conversation controls the direction of the interview.

2. **Establishing rapport:**

Some of the pitfalls to be avoided are

1. Do not deliberately mislead the user staff about the purpose of the study. A careful briefing is required. Too much of technical details will confuse the user and hence only information that is necessary has to be given to the participants.
2. Assure interviewees confidentiality that no information they offer will be released to unauthorized personnel. The promise of anonymity is very important.
3. Avoid personal involvement in the affairs of the users department or identification with one section at the cost of another.
4. Avoid showing off your knowledge or sharing information received from other sources.

5. Avoid acting like an expert consultant and confidant. This can reduce the objectivity of the approach and discourage people from freely giving information
6. Respect the time schedules and preoccupations of your subjects. Do not make an extended social event out of the meeting.
7. Do not promise anything you cannot or should not deliver, such as advice or feedback.
8. Dress and behave appropriately for the setting and the circumstances of the user contact.
9. Do not interrupt the interviewee. Let him/her finish talking.

3. **Asking the questions:** Except in unstructured interviews, it is important that each question is asked exactly as it is worded. Rewording may provoke a different answer .The question must also be asked in the same order as they appear on the interview schedule. Reversing the sequence destroys the comparability of the interviews. Finally each question must be asked unless the respondent, in answering the previous question, has already answered the next one.

4. **Obtaining and recording the response:** Interviews must be prepared well in order to collect further information when necessary. The information received during the interview must be recorded for later analysis.

5. **Data recording and the notebook:** Many system studies fail because of poor data recording. Care must be taken to record the data, their source and the time of collection. If there is no record of a conversation, the analyst won't be remembering enough details, attributing to the wrong source or distorting the data. The form of the notebook varies according to the type of study, for the amount of data, the number of analysts, and their individual preferences. The "notebook" may be a card file, a set of carefully coded file folders. It should be bound and the pages numbered.

Questionnaire

This method is used by analyst to gather information about various issues of system from large number of persons. This tool has collection of questions to which individuals respond.

The advantages of questionnaire are

1. It is economical and requires less skill to administer than the interview.
2. Unlike the interview, which generally questions one subject at a time, a questionnaire can be administered to large numbers of individuals simultaneously.
3. The standardized wording and order of the questions and the standardized instructions for reporting responses ensure uniformity of questions.
4. The respondents feel greater confidence in the anonymity of a questionnaire than in that of an interview. In an interview, the analyst usually knows the user staff by name,

job function or other identification. With a questionnaire, respondents give opinions without fear that the answer will be connected to their names.

5. The questionnaire places less pressure on subjects for immediate responses. Respondents have time to think the questions over and do calculations to provide more accurate data.

Types of interviews and questionnaires

Interviews and Questionnaires vary widely in form and structure. Interviews range from highly unstructured to the highly structured alternative in which the questions and responses are fixed.

The unstructured Interview:

The Unstructured interview is non-directive information gathering technique. It allows respondents to answer questions freely in their own words. The responses in this case are spontaneous and self-revealing. The role of the analyst as an interviewer is to encourage the respondent to talk freely and serve as a catalyst to the expression of feelings and opinions. This method works well in a permissive atmosphere in which subjects have no feeling of disapproval.

The structured Interview:

In this alternative the questions are presented with exactly the same wordings and in the same order to all subjects. Standardized questions improve the reliability of the responses by ensuring that all subjects are responding to the same questions.

Structured interviews and questionnaires may differ in the amount of structuring of the questions.

Questions may be either

- Open-ended questions
- Close-ended questions

An **open-ended** question requires no response direction or specific response. Questionnaire is written with space provided for the response. Such questions are more often used in interviews than in questionnaires because scoring takes time.

Close-ended questions are those, in which the responses are presented as a set of alternatives.

There are five major types of closed questions.

1. **Fill in the blanks**: in which questions request specific information. These responses can be statically analyzed.

2. **Dichotomous (Yes/No type):** in which questions will offer two answers. This has advantages similar to those of the multiple-choice type. Here the question sequence and content are also important
3. **Ranking scales questions:** Ask the respondent to rank a list of items in order of importance or preference
4. **Multiple-choice questions:** Offer respondents specific answer choices. This offers the advantage of faster tabulation and less analyst bias due to the order in which the questions are given. Respondents have a favorable bias toward the first alternative item. Alternating the order in which answer choices are listed may reduce bias but at the expense of additional time to respond to the questionnaire.
5. **Rating scales** – These types of questions are an extension of the multiple-choice design. The respondent is offered a range of responses along a single dimension

Open-ended questions are ideal in exploratory situations where new ideas and relationships are sought.

Disadvantages of open-ended questions:

- The main drawback is the difficulty of interpreting the subjective answers and the tedious responses to open ended questions.
- Other drawbacks are potential analyst, bias in interpreting the data and time-consuming tabulation

Closed questions are quick to analyze.

Disadvantages of close-ended questions:

- They are costly to prepare.
- They have the additional advantage of ensuring that answers are given in a frame of reference consistent with the line of inquiry.

Procedure for questionnaire construction:

There are six steps for constructing a questionnaire

1. Decide on the data should be collected, that is used to define the problem to be investigated.
2. Decide the type of questionnaire should be used (closed or open ended).
3. Outline the topics for the questionnaire and then write the questions.
4. Edit the questionnaire for technical defect that reflect personal values.
5. Pretest the questionnaire to see how well it works
6. Do a final editing to ensure that questionnaire is ready for administration. This includes a close look at the content, form and sequence of questions as well as the appearance and clarity of the procedure for using the questionnaire.

Important thing in questionnaire construction is the formulation of reliable and valid questions. To do a satisfactory job, the analyst must focus on question content, wording and format. The following criteria has to be considered for constructing a questionnaire

- Question content (Is the question necessary?, does it cover the area intended?, does the participants have proper information to answer the questions?, is the question biased?)
- Question wording (Is the question worked for the participant's background and experience?, Can the question be misinterpreted?, is the question clear and direct?)
- Question format (How can the question be asked?, Is the response form easy or adequate for the job?, Is there any contamination effect?)

Reliability of data from respondents:

The data collected from the user staff is assumed to correspond to the way in which events occur. If such reports are used, then there will be several errors like

- Reports of a given event from several staff members who has little training in observation will not be accurate.
- Respondent's ability to forget things.
- Reluctance of the person being interviewed.
- Inability of the participants to communicate their ideas or the analyst to get required information from the participants.

The reliability-validity issue:

Information – gathering instrument faces 2 major tests – reliability and validity. The term reliability is synonymous with dependability, consistency and accuracy. Concern for reliability comes from the necessity for dependability in measurement. Reliability may be approached in 3 ways

1. If we administer the same questionnaire to the same subject will we get the similar or same results? This question implies a definition of reliability as stability, dependability and predictability.
2. Does the questionnaire measure the true variables? This question focuses on the accuracy aspect of reliability.
3. How much error of measurement is there in the proposed questionnaire? Errors of measurement are random errors.

The most common question that defines validity is: Does the instrument measure what we think it is measuring? It refers to the notion that the questions asked are worded to produce the information sought. In contrast reliability means that information gathered is dependable enough to be used for decision making. In validity, the emphasis is on what is being measured?. Thus the adequacy of an information-gathering tool is judged by the criteria of validity and reliability. Both depend on the design of the instrument as well as the way it is administered.

The main aim of fact finding techniques is to determine the information requirements of an organization used by analysts to prepare a precise System Requirement specification (SRS) understood by user.

Ideal SRS Document should

- be Complete, Unambiguous, and Jargon-free.
- specify operational, tactical, and strategic information requirements.
- solve possible disputes between users and analyst.
- use graphical aids which simplify understanding and design.

Feasibility Study

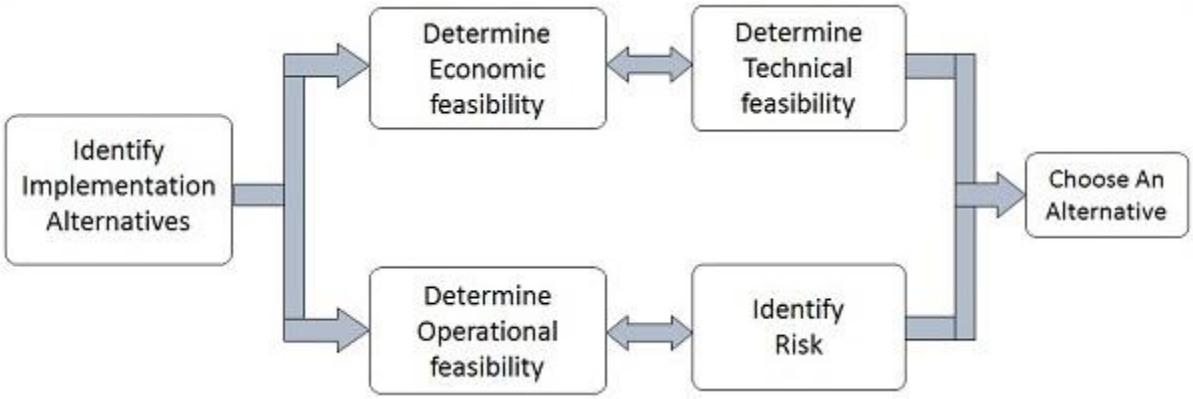
Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

- It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
- It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
- The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
- The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

Steps Involved in Feasibility Analysis

The following steps are to be followed while performing feasibility analysis −

- Form a project team and appoint a project leader.
- Develop system flowcharts.
- Identify the deficiencies of current system and set goals.
- Enumerate the alternative solution or potential candidate system to meet goals.
- Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
- Weight the performance and cost effectiveness of each candidate system.
- Rank the other alternatives and select the best candidate system.
- Prepare a system proposal of final project directive to management for approval.



Types of Feasibilities

Economic Feasibility

- It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
- It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
- The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
- It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

Technical Feasibility

- It investigates the technical feasibility of each implementation alternative.
- It analyzes and determines whether the solution can be supported by existing technology or not.
- The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
- It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

Operational Feasibility

- It determines whether the system is operating effectively once it is developed and implemented.

- It ensures that the management should support the proposed system and its working feasible in the current organizational environment.
- It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits.
- It also ensures that the computer resources and network architecture of candidate system are workable.

Behavioral Feasibility

- It evaluates and estimates the user attitude or behavior towards the development of new system.
- It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Schedule Feasibility

- It ensures that the project should be completed within given time constraint or schedule.
- It also verifies and validates whether the deadlines of project are reasonable or not.

Structured Analysis

Analysts use various tools to understand and describe the information system. One of the ways is using structured analysis.

What is Structured Analysis?

Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user.

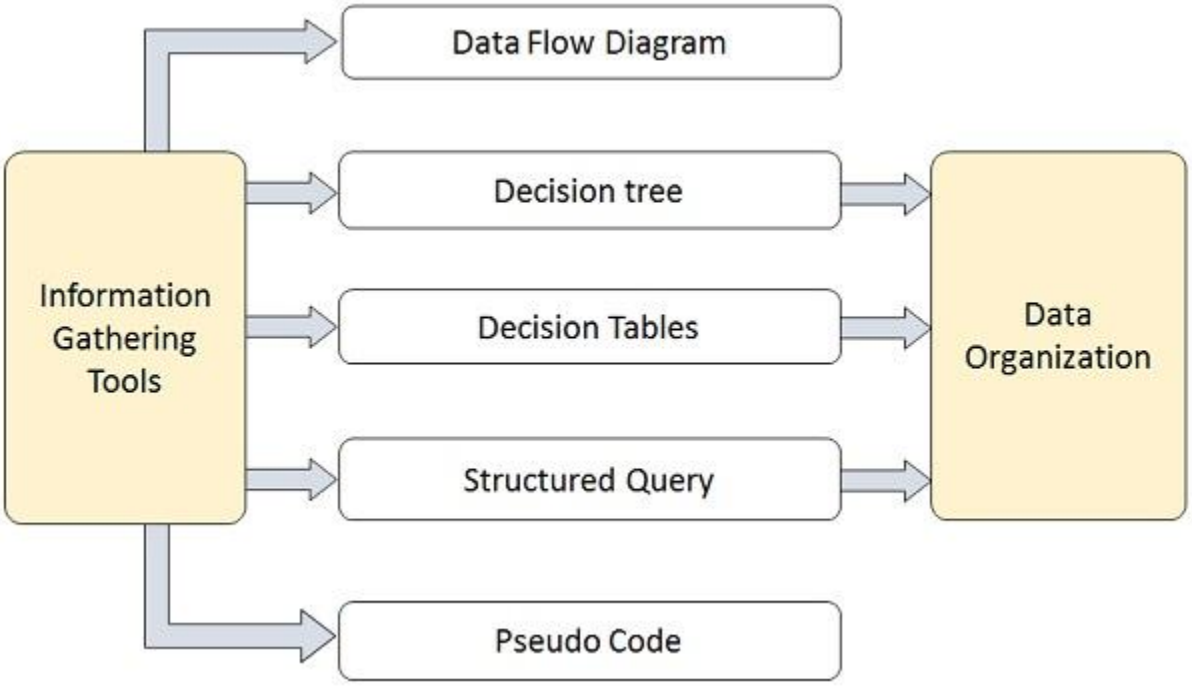
It has following attributes −

- It is graphic which specifies the presentation of application.
- It divides the processes so that it gives a clear picture of system flow.
- It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
- It is an approach that works from high-level overviews to lower-level details.

Structured Analysis Tools

During Structured Analysis, various tools and techniques are used for system development. They are −

- Data Flow Diagrams
- Data Dictionary
- Decision Trees
- Decision Tables
- Structured English
- Pseudocode



Data Flow Diagrams (DFD) or Bubble Chart

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

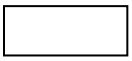
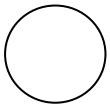
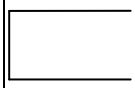
- It shows the flow of data between various functions of system and specifies how the current system is implemented.
- It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
- Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
- It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.
- A DFD consist of a series of bubbles joined by lines. The bubbles represent data transformations and the line represents data flow in the system.

- A DFD describes what data flow rather than how they are processed, so it does not depend on hardware, software, data structure or file organization.

Basic Elements of DFD

DFD is easy to understand and quite effective when the required design is not clear and the user want a notational language for communication. However, it requires a large number of iterations for obtaining the most accurate and complete solution.

The following table shows the symbols used in designing a DFD and their significance −

Symbol Name	Symbol	Meaning
Square		**Source or Destination** of System Data
Arrow		**Data flow**, i.e.: data in motion. It is a pipeline through which information flows.
Circle		Represents **Process** that transforms incoming data flows into outgoing data.
Open Rectangle		**Data Store** ie: data at rest or a temporary repository of data.

Types of DFD

DFDs are of two types: Physical DFD and Logical DFD. The following table lists the points that differentiate a physical DFD from a logical DFD.

Physical DFD	Logical DFD
It is implementation dependent. It shows which functions are performed.	It is implementation independent. It focuses only on the flow of data between processes.

It provides low level details of hardware, software, files, and people.	It explains events of systems and data required by each event.
It depicts how the current system operates and how a system will be implemented.	It shows how business operates; not how the system can be implemented.

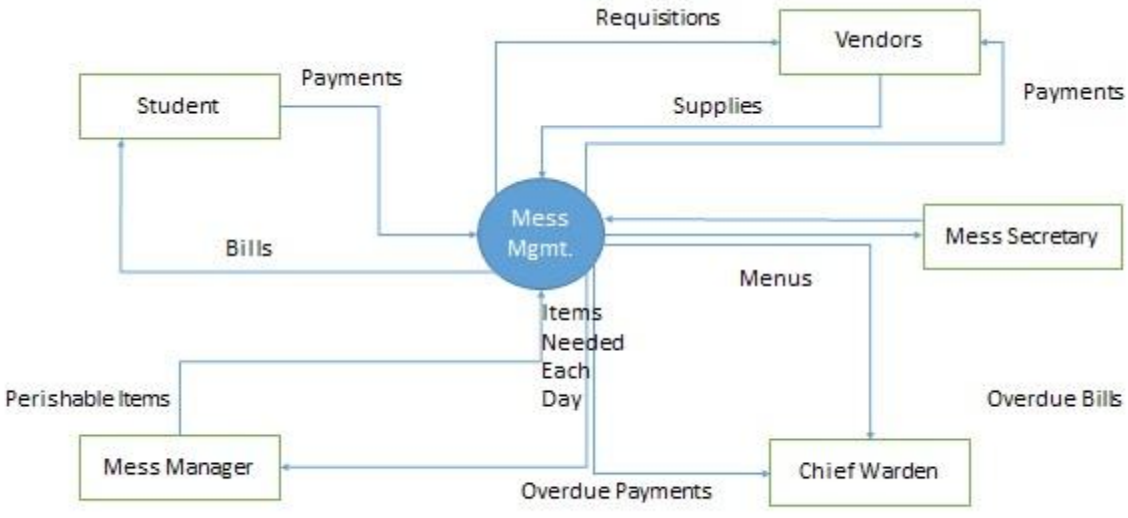
Rules for constructing DFDs are

1. Processes should be named and numbered for easy reference.
2. The direction of flow is from top to bottom and from left to right. Data flows from source to the destination.
3. Process should be numbered, if they are exploded into lower level details.
4. The names of data stores, sources and destination should be in capital letters.
5. Process and data flow names first letter should be in capital letter.

Context Diagram

A context diagram helps in understanding the entire system by one DFD which gives the overview of a system. It starts with mentioning major processes with little details and then goes onto giving more details of the processes with the top-down approach.

The context diagram of mess management is shown below.



Process Modeling with Logical Data Flow Diagram

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

- DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.
- DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub processes.
- DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system's functioning.
- Progression to Levels 3, 4 and beyond is possible, but going beyond Level 3 is uncommon. Doing so can create complexity that makes it difficult to communicate, compare or model effectively.
- Using DFD layers, the cascading levels can be nested directly in the diagram, providing a cleaner look with easy access to the deeper dive.
- By becoming sufficiently detailed in the DFD, developers and designers can use it to write pseudocode, which is a combination of English and the coding language. Pseudocode facilitates the development of the actual code.

Steps in DFD

1. Create a list of activities
2. Construct Context Level DFD (Identify Source and Sink)
3. Construct Level 0 DFD (Identify Manageable sub process)
4. Construct Level 1 – N DFD (Identifies actual data flows and data stores)
5. Check against rules of DFD

Example

The operations of a simple lemonade stand will be used to demonstrate the creation of dataflow diagram.

Think through the activities that take place at a lemonade stand

1. First the customer will place the order
2. Then the server will check for the availability of that product and serve the product
3. Customer will pay for the product

Step 1: Create a list of activities

- ✓ Customer Order
- ✓ Serve Product
- ✓ Collect Payment
- ✓ Produce Product
- ✓ Store Product

Also think of the additional activities needed to support the basic activities

- ✓ Customer Order
- ✓ Serve Product
- ✓ Collect Payment
- ✓ Produce Product
- ✓ Store Product
- ✓ Order Raw Materials
- ✓ Pay for Raw Materials
- ✓ Pay for Labor

Group these activities in some logical fashion, possibly functional areas.

Function I

- ✓ Customer Order
- ✓ Serve Product
- ✓ Collect Payment

Function II

- ✓ Produce Product
- ✓ Store Product

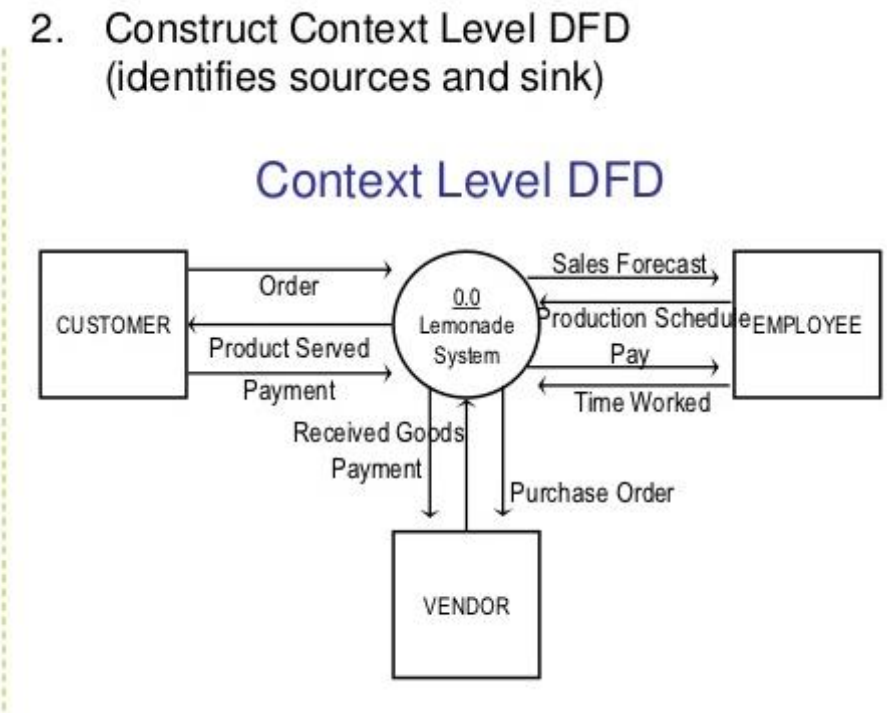
Function III

- ✓ Order Raw Materials
- ✓ Pay for Raw Materials

Function IV

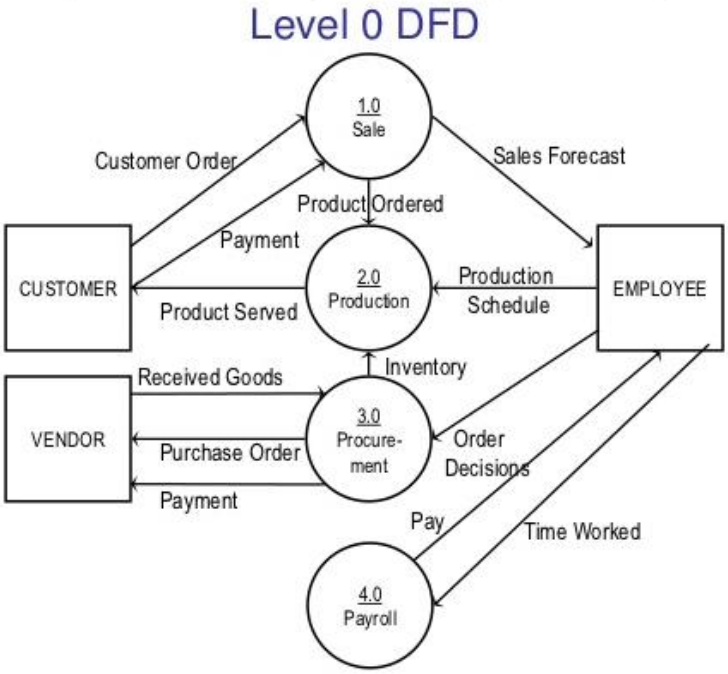
- ✓ Pay for Labor

Step 2: Create a context level diagram identifying the sources and sinks (users)



Step 3: Create a level 0 diagram identifying the logical subsystems that may exist.

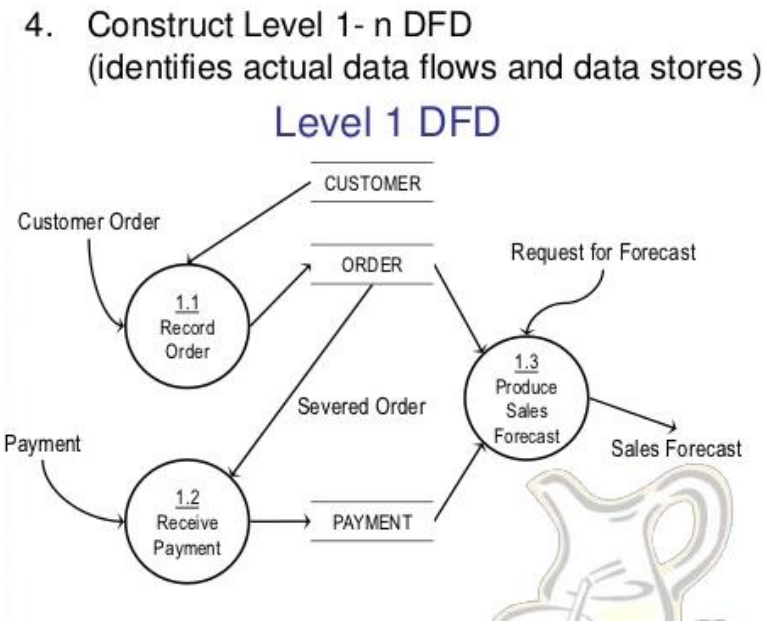
3. Construct Level 0 DFD
(identifies manageable sub processes)



Step 4: Create a level 1 decomposing the processes in level 0 and identifying data stores

Level 1 DFD for

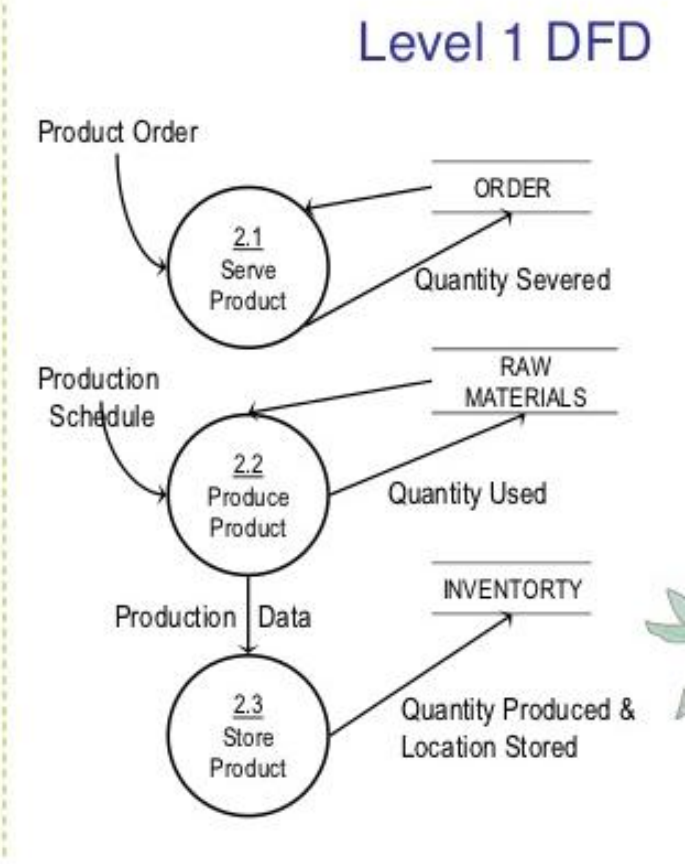
- ✓ **Customer Order**
- ✓ **Collect Payment**



Level 1 DFD for

- ✓ Serve Product
- ✓ Produce Product
- ✓ Store product

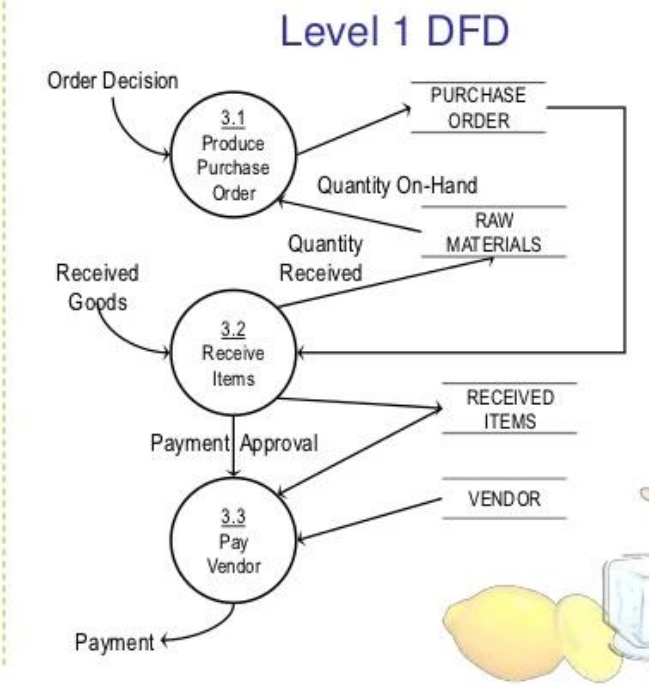
4. Construct Level 1 (continued)



Level 1 DFD for

- ✓ Order Raw Materials
- ✓ Pay for Raw Materials

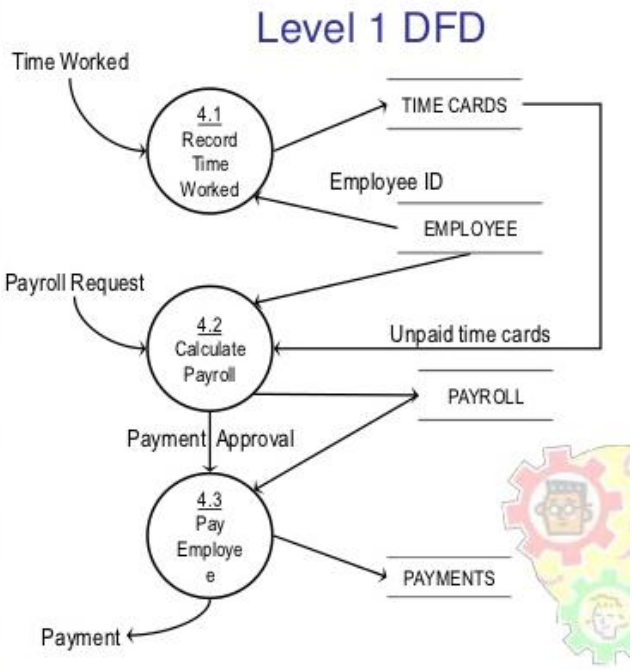
4. Construct Level 1 (continued)

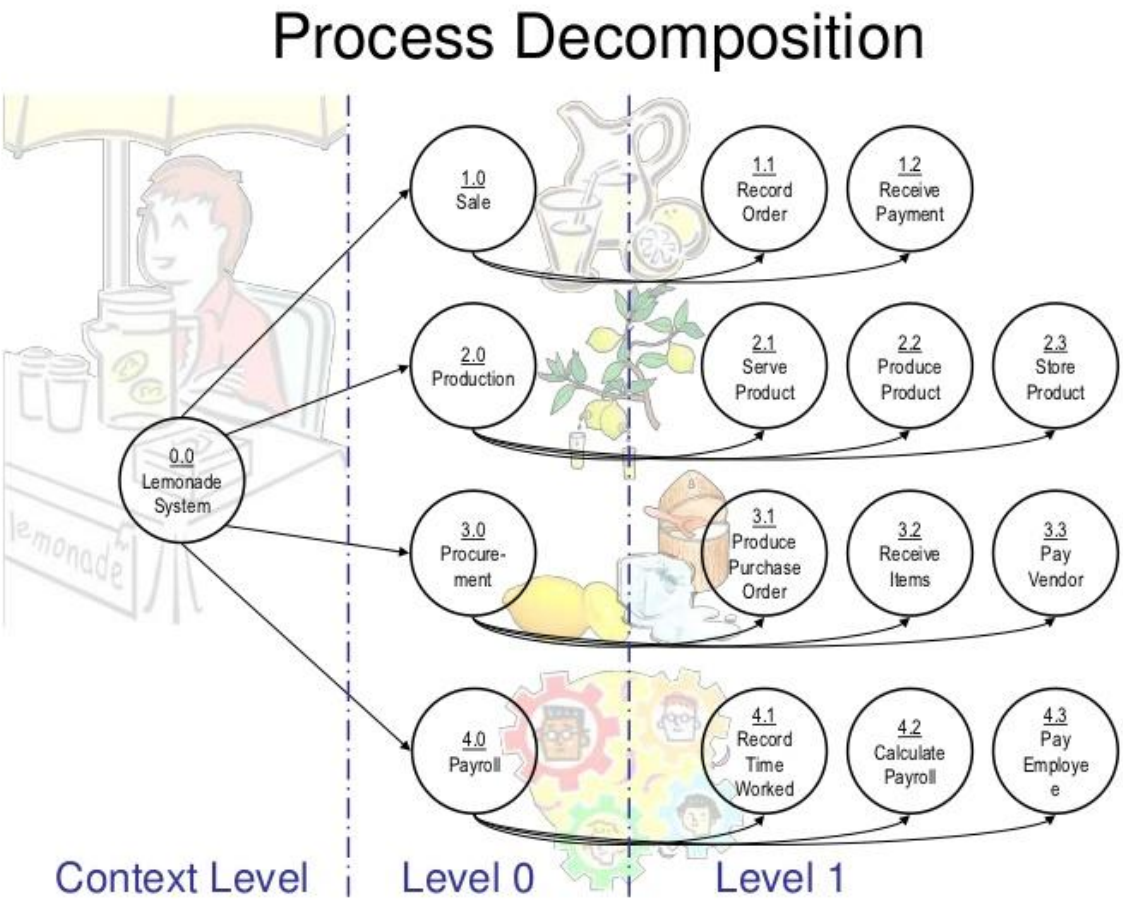


Level 1 DFD for

- ✓ Pay for Labor

4. Construct Level 1 (continued)





Data Dictionary

A data dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements i.e details of data flows, data stored in data stores, and the processes.

A data dictionary improves the communication between the analyst and the user. It plays an important role in building a database. Most DBMSs have a data dictionary as a standard feature. For example, refer the following table −

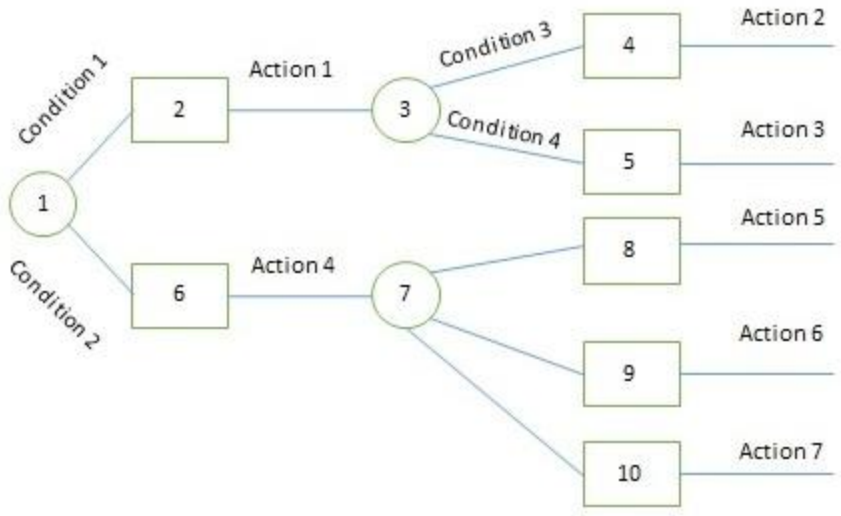
S.No.	Data Name	Description	No. of Characters
1	ISBN	ISBN Number	10
2	TITLE	title	60
3	SUB	Subjects Book	80

4	ANAME	Author Name	15

Decision Trees

Decision trees are a method for defining complex relationships by describing decisions and avoiding the problems in communication. A decision tree is a diagram that shows alternative actions and conditions within horizontal tree framework. Thus, it depicts which conditions to consider first, second, and so on.

Decision trees depict the relationship of each condition and their permissible actions. A square node indicates an action and a circle indicates a condition. It forces analysts to consider the sequence of decisions and identifies the actual decision that must be made.



The major limitation of a decision tree is that it lacks information in its format to describe what other combinations of conditions you can take for testing. It is a single representation of the relationships between conditions and actions.

For example, refer the following decision tree −

Decision Tables

Decision tables are a method of describing the complex logical relationship in a precise manner which is easily understandable.

- It is useful in situations where the resulting actions depend on the occurrence of one or several combinations of independent conditions.
- It is a matrix containing row or columns for defining a problem and the actions.
- It is used to define a problem and the actions to be taken.
- It is a single representation of the relationships between conditions and actions.

Components of a Decision Table

- **Condition Stub** − It is in the upper left quadrant which lists all the condition to be checked.
- **Action Stub** − It is in the lower left quadrant which outlines all the action to be carried out to meet such condition.
- **Condition Entry** − It is in upper right quadrant which provides answers to questions asked in condition stub quadrant.
- **Action Entry** − It is in lower right quadrant which indicates the appropriate action resulting from the answers to the conditions in the condition entry quadrant.

The entries in decision table are given by Decision Rules which define the relationships between combinations of conditions and courses of action. In rules section,

- Y shows the existence of a condition.
- N represents the condition, which is not satisfied.
- A blank - against action states it is to be ignored.

X (or a check mark will do) against action states it is to be carried out.

For example, refer the following table −

CONDITIONS	Rule 1	Rule 2	Rule 3	Rule 4
Advance payment made	Y	N	N	N
Purchase amount = Rs 10,000/-	-	Y	Y	N
Regular Customer	-	Y	N	-
ACTIONS				
Give 5% discount	X	X	-	-
Give no discount	-	-	X	X

Structured English

Structure English is derived from structured programming language which gives more understandable and precise description of process. It is based on procedural logic that uses construction and imperative sentences designed to perform operation for action.

- It is best used when sequences and loops in a program must be considered and the problem needs sequences of actions with decisions.
- It does not have strict syntax rule. It expresses all logic in terms of sequential decision structures and iterations.

For example, see the following sequence of actions −

```
if customer pays advance
   then
      Give 5% Discount
   else
      if purchase amount >=10,000
         then
            if  the customer is a regular customer
               then Give 5% Discount
            else  No Discount
         end if
      else No Discount
```

```
   end if
end if
```

Pseudo code

A pseudo code does not conform to any programming language and expresses logic in plain English.

- It may specify the physical programming logic without actual coding during and after the physical design.
- It is used in conjunction with structured programming.
- It replaces the flowcharts of a program.

Guidelines for Selecting Appropriate Tools

Use the following guidelines for selecting the most appropriate tool that would suit your requirements −

- Use DFD at high or low level analysis for providing good system documentations.
- Use data dictionary to simplify the structure for meeting the data requirement of the system.
- Use structured English if there are many loops and actions are complex.
- Use decision tables when there are a large number of conditions to check and logic is complex.
- Use decision trees when sequencing of conditions is important and if there are few conditions to be tested.

Conceptual Data Modelling

It is representation of organizational data which includes all the major entities and relationship. System analysts develop a conceptual data model for the current system that supports the scope and requirement for the proposed system.

The main aim of conceptual data modeling is to capture as much meaning of data as possible. Most organization today uses conceptual data modelling using E-R model which uses special notation to represent as much meaning about data as possible.

Data Modelling with Logical Entity Relationship

Entity Relationship Model

ER-Diagram is a visual representation of data that describes how data is related to each other. It is a technique used in database design that helps describe the relationship between various entities of an organization.

Terms used in E-R model

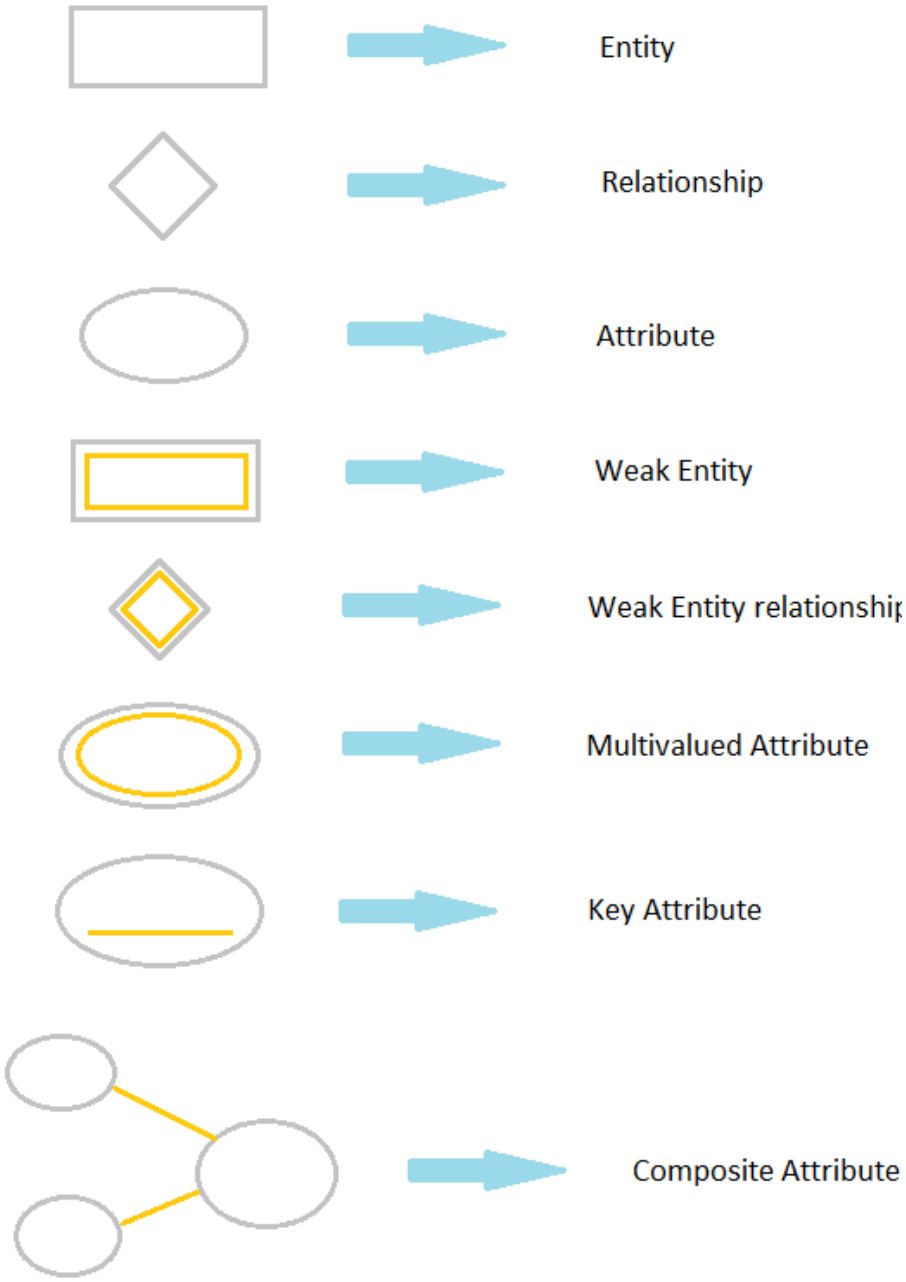
- **ENTITY** − It specifies distinct real world items in an application. For example: vendor, item, student, course, teachers, etc.

- **RELATIONSHIP** − are the meaningful dependencies between entities. For example, vendor supplies items, teacher teaches courses. Here supplies and course are relationship.

- **ATTRIBUTES** − It specifies the properties of relationships. For example, vendor code, student name.

Symbols used in E-R model and their respective meanings −

Symbols and Notations



Components of E-R Diagram

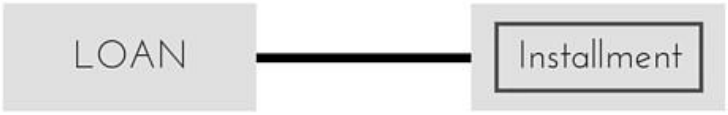
The E-R diagram has three main components.

1) Entity

An **Entity** can be any object, place, person or class. In E-R Diagram, an **entity** is represented using rectangles. Consider an example of an Organisation. Employee, Manager, Department, Product and many more can be taken as entities from an Organisation.

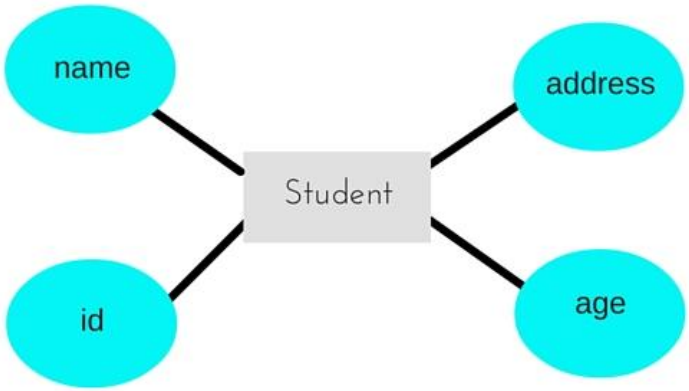
Weak Entity

Weak entity is an entity that depends on another entity. Weak entity doesn't have key attribute of its own. Double rectangle represents weak entity. In the below example, Instalment is weak entity.



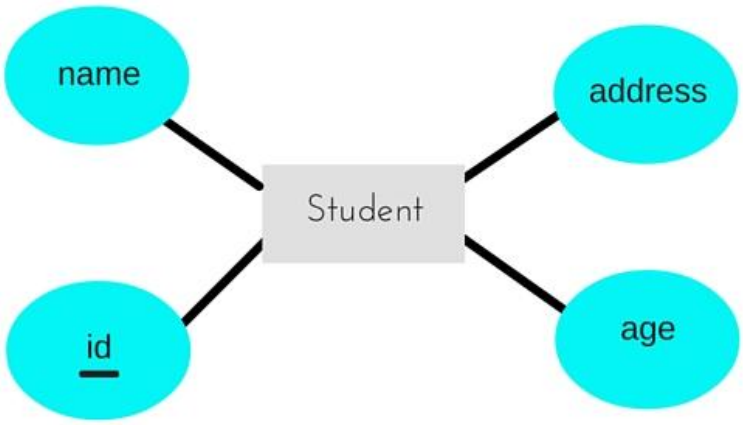
2) Attribute

An **Attribute** describes a property or characteristics of an entity. For example, Name, Age, Address etc can be attributes of a Student. An attribute is represented using ellipse.



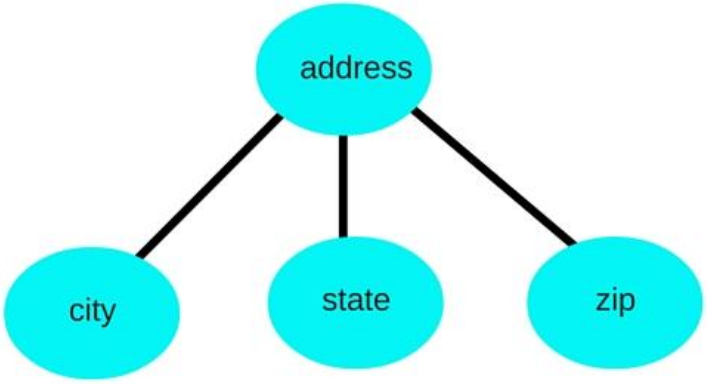
Key Attribute

Key attribute represents the unique characteristic of an Entity. It is used to represent Primary key. Ellipse with underlying line represents Key Attribute.



Composite Attribute

An attribute can also have its own attributes. These attributes are known as **Composite** attribute.



3) Relationship

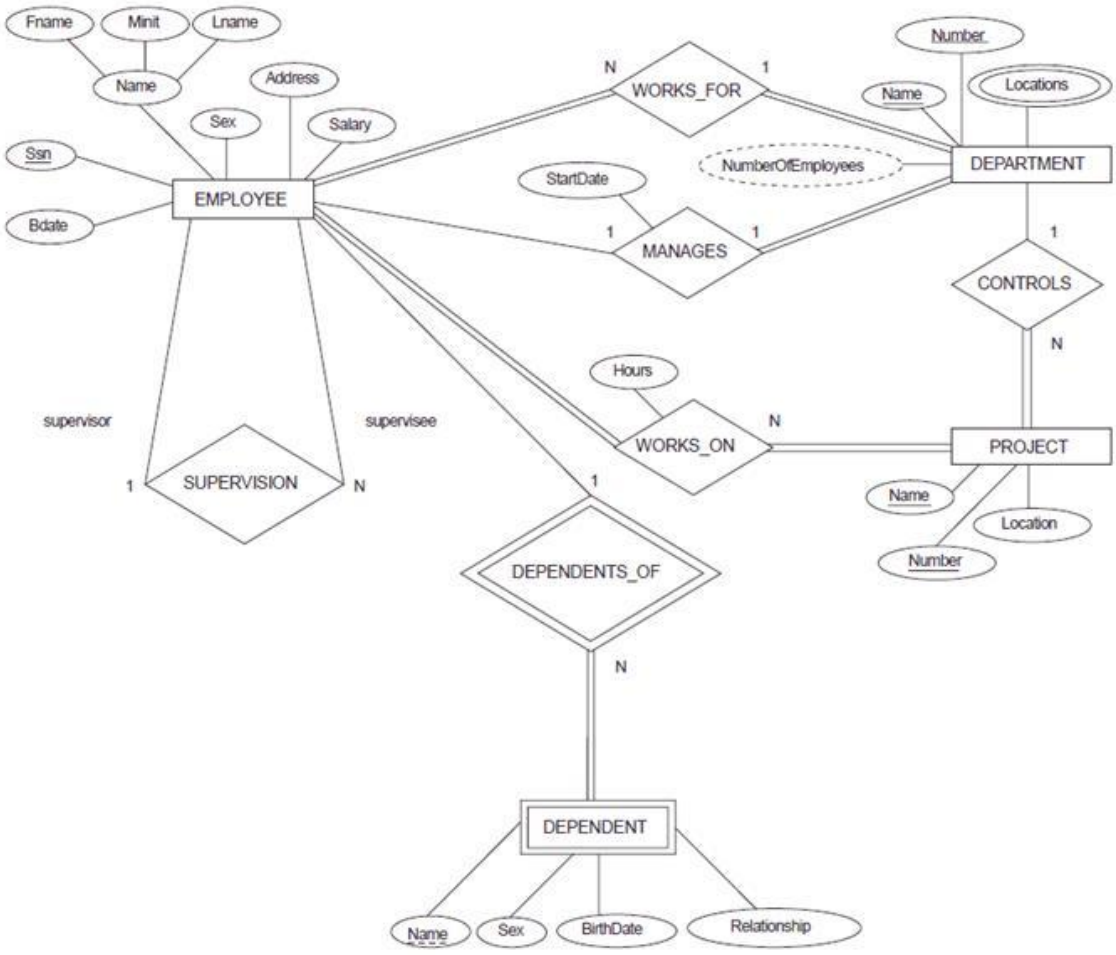
A Relationship describes relations between **entities**. Relationship is represented using diamonds.

ER diagram example

Construct an ER Diagram for Company having following details:

Company is organized into DEPARTMENT. Each department has unique name and a particular employee who manages the department. Start date for the manager is recorded. Department may have several locations.

- A department controls a number of PROJECTS. Projects have a unique name, number and a single location.
- Company's Employee Name, SSNO, address, salary, sex and birth date is recorded. An employee is assigned to one department, but may work for several projects (not necessarily controlled by her dept). Number of hours/week an employee works on each project is recorded; Supervisor for the employee should be recorded.
- Employee's DEPENDENT is tracked for health insurance purposes (dependent name, birth date, relationship to employee).



A university registrar office maintains data about the following entities:

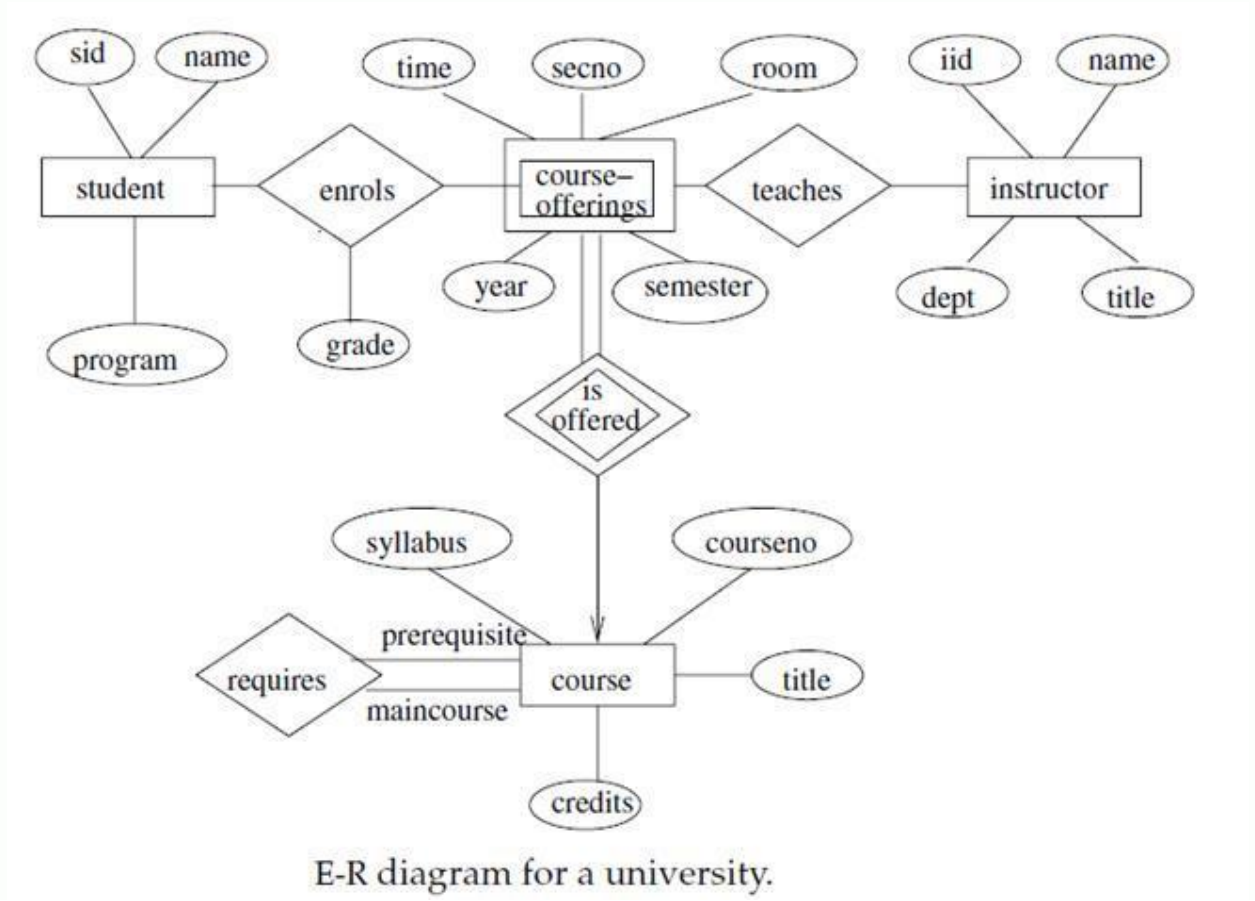
Courses: number, title, credits, syllabus, and prerequisites;

Course offereings: course number, year, semester, section number, instructor(s), timings, and classroom;

Students: student-id, name, and program;

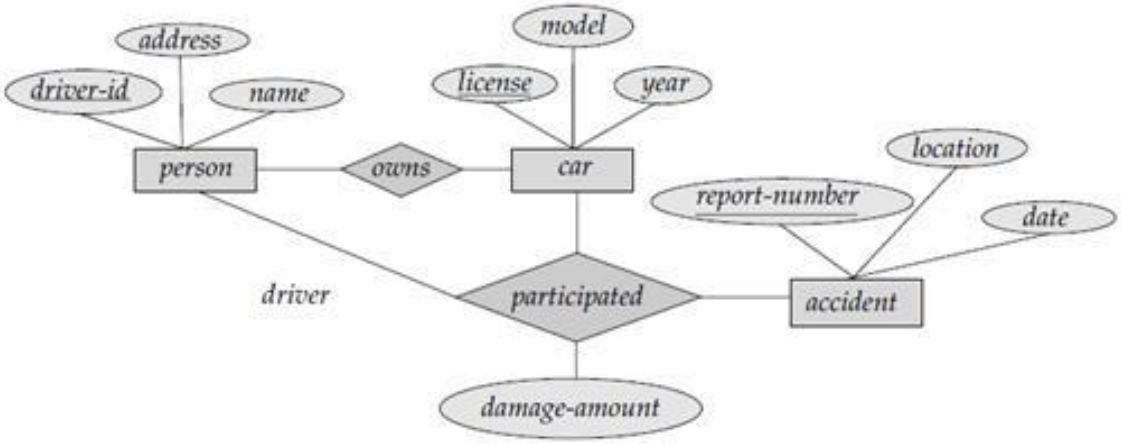
Instructors: identification number, name, department, and title.

Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.



E-R diagram for a university.

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each.



E-R diagram for a Car-insurance company.

Construct appropriate tables for the above ER Diagram

Car insurance tables:

Person (driver-id, name, address)

Car (license, year,model)

Accident (report-number, date, location)

Participated (driver-id, license, report-number, damage-amount)

Unit 4
System Design

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. *"how to implement?"* It tells the approach to the creation of a new system. It provides the methods to implement the system given in the feasibility study. Emphasis is placed on transforming the performance requirements to design specifications. Design is done both logically and physically.

Systems are created in order to meet the needs of the users. They are not only intended to solve the existing problems, but they also come up with acceptable solutions to the problems that may arise in the future. The whole process of system development, from blueprint to the actual product, involves considering all the relevant factors and taking the required specifications and creating a useful system based on strong technical, analytical and development skills of the professionals.

Let's get back to our discussion about what the system design phase is and the importance of system design in the process of system development.

Being another important step in the system development process, system designing phase commences after the system analysis phase is completed. It's appropriate to mention that the output or the specifications taken through the phase of system analysis become an input in the system design phase which in turn leads to workout based on the user defined estimations.

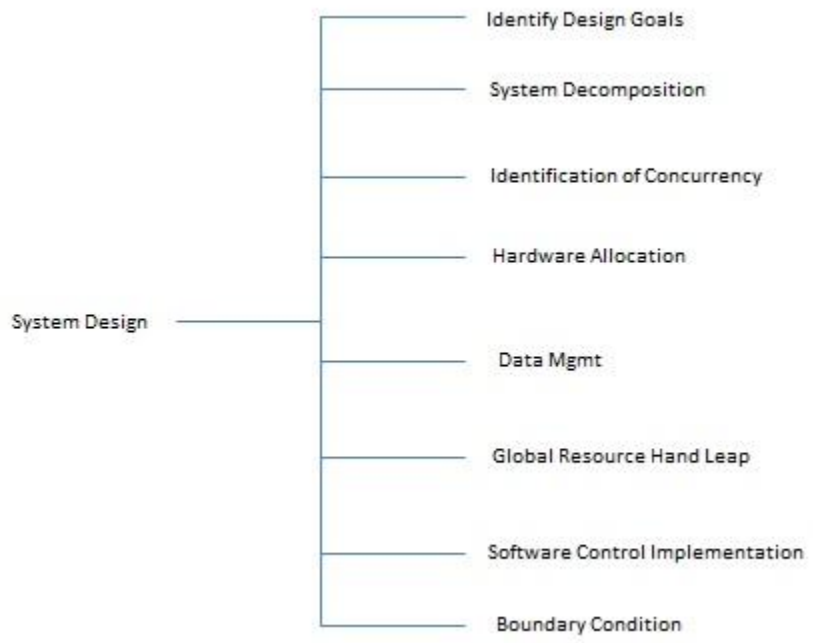
The importance of this phase may be understood by reason of the fact that it involves identifying data sources, the nature and type of data that is available. For example, in order to design a salary system, there is a need for using inputs, such as, attendance, leave details, additions or deductions etc. This facilitates understanding what kind of data is available and by whom it is supplied to the system so that the system may be designed considering all the relevant factors. In addition, system designing leads to ensure that the system is created in such a way that it fulfils the need of the users and keep them at ease being user-oriented.

In terms of the flexibility, one of the main objectives of this phase is that it is intended to design such a system which can be dynamic in nature and responsive to the changes if required.

Another important objective is that the phase of system designing is concerned with creating the system which can work efficiently providing the required output and being responsive to the time within a given time limit.

The aspect of reliability and physical security of data cannot be ignored. With this respect, the system designing phase ensures security measures of the system effectively and efficiently.

So, Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.



Inputs to System Design

System design takes the following inputs −

- Statement of work
- Requirement determination plan
- Current situation analysis
- Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

Outputs for System Design

System design gives the following outputs −

- Infrastructure and organizational changes for the proposed system.
- A data schema, often a relational schema.
- Metadata to define the tables/files and columns/data-items.
- A function hierarchy diagram or web page map that graphically describes the program structure.

- Actual or pseudocode for each module in the program.
- A prototype for the proposed system.

Process of System Design

The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from a user-oriented document to a document, oriented to the programmers or data base personnel.

System design goes through two phases of development

- Logical design
- Physical design

Logical design

The logical system design specifies the 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. Logical design includes entity-relationship diagrams (ER diagrams).

The design covers the following:

1. Reviews the current physical system -its data flow, file content, volumes, frequencies, etc.
2. Prepares output specifications - determines the format, content, and frequency of reports, including terminal specifications and locations.
3. Prepares input specifications - format, content, and most of the input functions. This includes determining the flow of the document from the input data source to the actual input location.
4. Prepares edit, security and control specifications.
5. Specifies the implementation plan.
6. Prepares a logical design of the information flow, output, input, controls and implementation plan.
7. Reviews benefits, costs, target dates and system constraints.

Physical design

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

- Input requirement
- Output requirements
- Storage requirements

- Processing requirements
- System control and backup or recovery

Put another way, the physical portion of system 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.
- **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 system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor.

This produces the working system by defining the design specifications that tell programmers exactly what the candidate system must do. In turn the programmers write the necessary programs or modify the existing software package that gets the input and performs the necessary actions on the existing file or database, and produces the report in hardcopy or displays it on a screen, and update the data base.

Physical system design consists of the following steps.

1. Design the physical system.
    a. Specify input/output media.
    b. Design the database and specify backup procedures.
    c. Design physical information flow through the system and a physical design walkthrough.
2. Plan system implementation
    a. Prepare conversion schedule and a target date.
    b. Determine training procedure, courses, and timetable.
3. Devise a test and implementation plan and specify any new hardware /software

4. Update benefits, costs, conversion date, and system constraints.

STRUCTURED DESIGN METHODOLOGY:

Design methodologies aim at the following

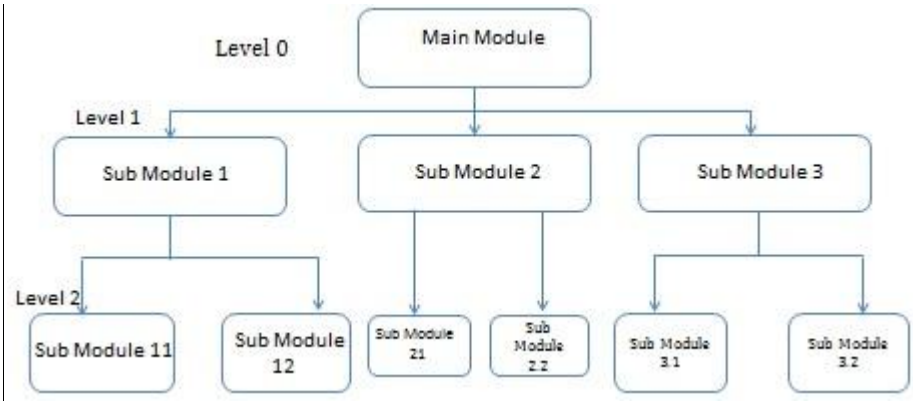
- Improve productivity of the analysts and programmers.
- Improve documentation.
- Cut down costs.
- Improve communication among users, analysts, designers and the programmers.
- Simplify the design by segmentation.

Design Strategies

Top-Down Strategy

The top-down strategy uses the modular approach to develop the design of a system. It is called so because it starts from the top or the highest-level module and moves towards the lowest level modules.

In this technique, the highest-level module or main module for developing the software is identified. The main module is divided into several smaller and simpler sub-modules or segments based on the task performed by each module. Then, each sub-module is further subdivided into several sub-modules of next lower level. This process of dividing each module into several sub-modules continues until the lowest level modules, which cannot be further subdivided, are not identified.



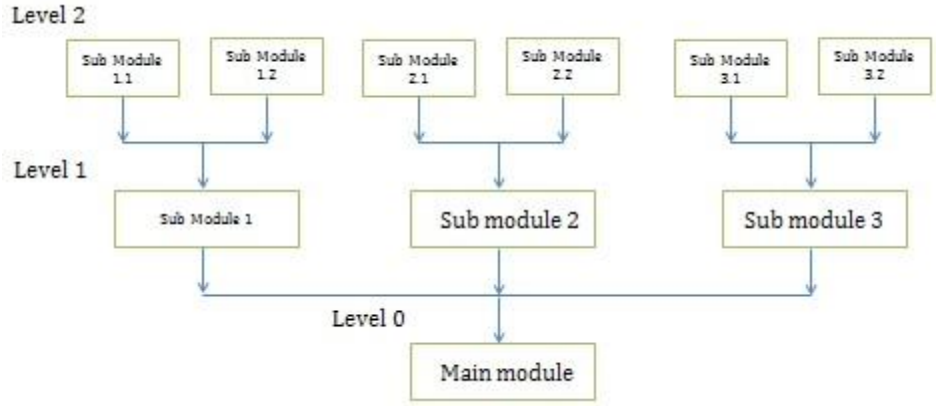
Bottom-Up Strategy

Bottom-Up Strategy follows the modular approach to develop the design of the system. It is called so because it starts from the bottom or the most basic level modules and moves towards the highest level modules.

In this technique,

- The modules at the most basic or the lowest level are identified.
- These modules are then grouped together based on the function performed by each module to form the next higher-level modules.

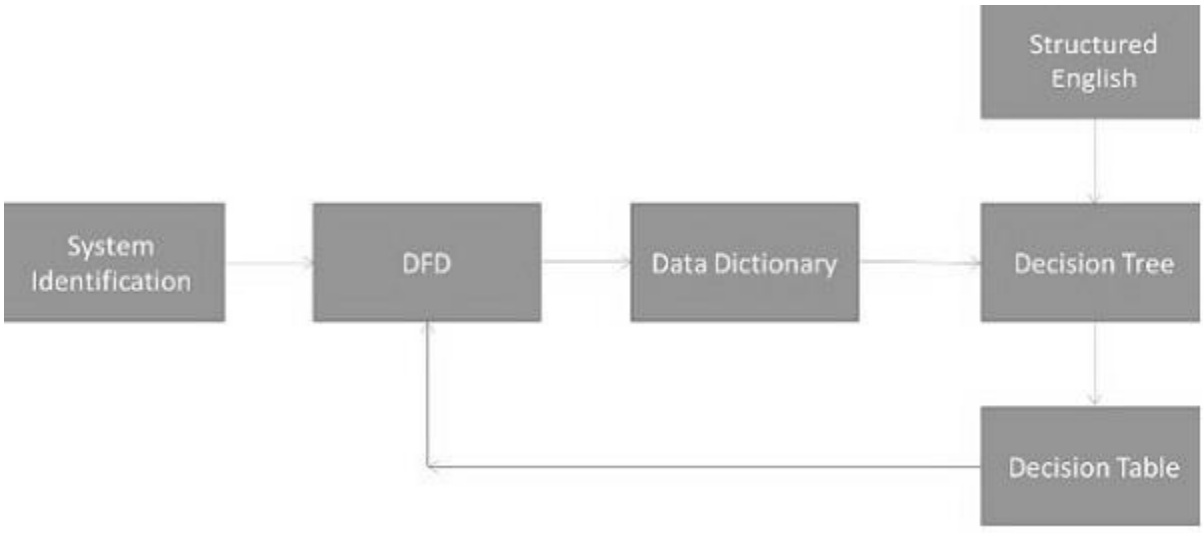
- Then, these modules are further combined to form the next higher-level modules.
- This process of grouping several simpler modules to form higher level modules continues until the main module of system development process is achieved.



Structured Design

Structured design is a data-flow based methodology that helps in identifying the input and output of the developing system. The main objective of structured design is to minimize the complexity and increase the modularity of a program. Structured design also helps in describing the functional aspects of the system.

In structured designing, the system specifications act as a basis for graphically representing the flow of data and sequence of processes involved in a software development with the help of DFDs. After developing the DFDs for the software system, the next step is to develop the structure chart.



Modularization

Structured design partitions the program into small and independent modules. These are organized in top down manner with the details shown in bottom.

Thus, structured design uses an approach called Modularization or decomposition to minimize the complexity and to manage the problem by subdividing it into smaller segments.

Advantages

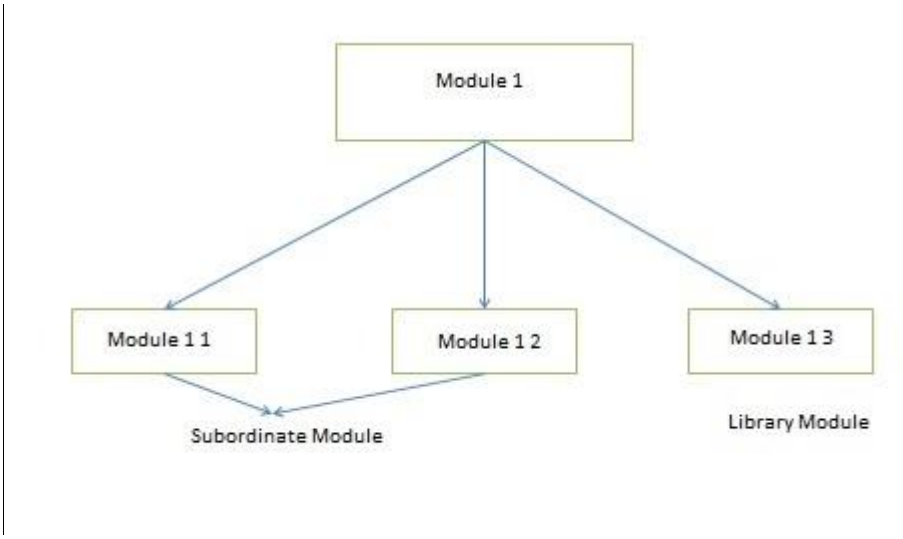
- Critical interfaces are tested first.
- It provides abstraction.
- It allows multiple programmers to work simultaneously.
- It allows code reuse.
- It provides control and improves morale.
- It makes identifying structure easier.

Structured Charts

Structured charts are a recommended tool for designing a modular, top down systems which define the various modules of system development and the relationship between each module. It shows the system module and their relationship between them.

It consists of diagram consisting of rectangular boxes that represent the modules, connecting arrows, or lines.

- **Control Module** − It is a higher-level module that directs lower-level modules, called subordinate modules.
- **Library Module** − It is a reusable module and can be invoked from more than one point in the chart.



We have two different approaches to design a structured chart −

- **Transform-Centered Structured Charts** - are used when all the transactions follow same path.
- **Transaction–Centered Structured Charts** − are used when all the transactions do not follow the same path.

Objectives of Using Structure Flowcharts

- To encourage a top-down design.
- To support the concept of modules and identify the appropriate modules.
- To show the size and complexity of the system.
- To identify the number of readily identifiable functions and modules within each function.
- To depict whether each identifiable function is a manageable entity or should be broken down into smaller components.

Factors Affecting System Complexity

To develop good quality of system software, it is necessary to develop a good design. Therefore, the main focus on while developing the design of the system is the quality of the software design. A good quality software design is the one, which minimizes the complexity and cost expenditure in software development.

The two important concepts related to the system development that help in determining the complexity of a system are **coupling** and **cohesion**.

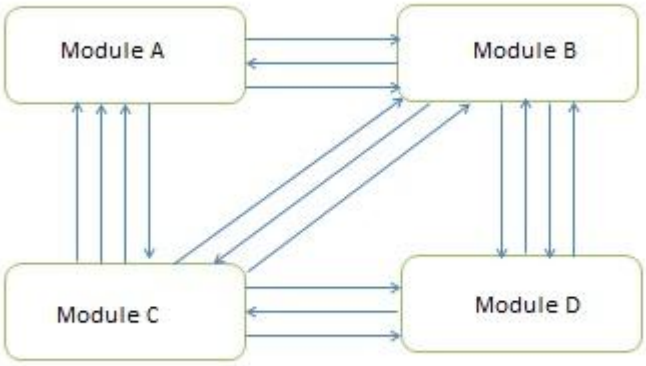
Coupling

Coupling is the measure of the independence of components. It defines the degree of dependency of each module of system development on the other. In practice, this means the stronger the coupling between the modules in a system, the more difficult it is to implement and maintain the system.

Each module should have simple, clean interface with other modules, and that the minimum number of data elements should be shared between modules.

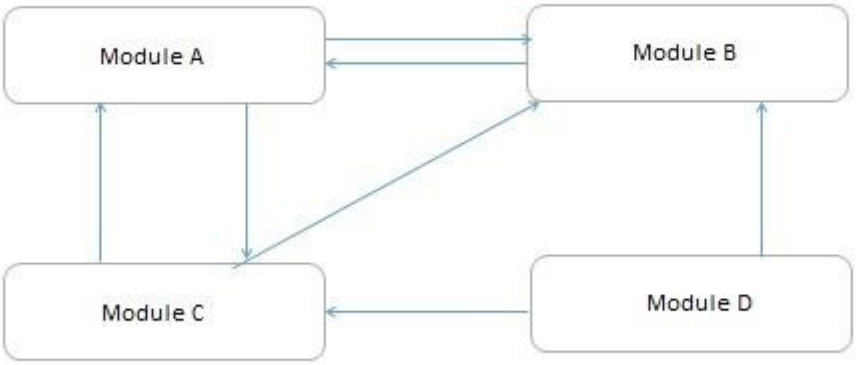
High Coupling

If the program units are highly dependent on each other and changes to one subsystem leads to high impact on the other subsystem.



Low Coupling

When systems are made up of components which are independent or almost independent, a change in one subsystem does not affect any other subsystem.



Coupling Measures

- **Content Coupling** − When one component actually modifies another, then the modified component is completely dependent on modifying one.
- **Common Coupling** − When amount of coupling is reduced somewhat by organizing system design so that data are accessible from a common data store.
- **Control Coupling** − When one component passes parameters to control the activity of another component.
- **Stamp Coupling** − When data structures is used to pass information from one component to another.

- **Data Coupling** − When only data is passed then components are connected by this coupling.

Cohesion

Cohesion is the measure of closeness of the relationship between its components. It defines the amount of dependency of the components of a module on one another. In practice, this means the systems designer must ensure that −

- They do not split essential processes into fragmented modules.
- They do not gather together unrelated processes represented as processes on the DFD into meaningless modules.

The best modules are those that are functionally cohesive. The worst modules are those that are coincidentally cohesive.

The worst degree of cohesion

Coincidental cohesion is found in a component whose parts are unrelated to another.

- **Logical Cohesion** − It is where several logically related functions or data elements are placed in same component.
- **Temporal Cohesion** − It is when a component that is used to initialize a system or set variables performs several functions in sequence, but the functions are related by timing involved.
- **Procedurally Cohesion** − It is when functions are grouped together in a component just to ensure this order.
- **Sequential Cohesion** − It is when the output from one part of a component is the input to the next part of it.

STRUCTURED WALKTHROUGH:

An activity of all the phases of a structured project is a walkthrough. It is an interchange of ideas to agree on the validity of a proposed solution to a problem. The purpose of the design walkthrough is to anticipate as many problems as possible because it is cheaper to make changes in the design phase rather than in during conversion. The objective is to give a maintainable design that is flexible and adaptable.

User Involvement: Walkthroughs are held to review the system test plan, program design and production acceptance. In each case user has to be involved. The amount of success depends on the amount of users' involvement.

- ✓ User involvement gives the designer feedback.

- ✓ It provides the basic understanding of what a candidate system will do and what it will not.
- ✓ It paves a way for user acceptance.
- ✓ It bridges the gap between the designer and the user

Activities during Structured Design

Several development activities are carried out during structured design. They are

- Database design
- Implementation planning
- System test preparation
- System interface specification
- User documentation

1. Database design: This activity deals with the design of the physical database. A key is to determine how the access paths are to be implemented. A physical path is derived from a logical path. It may be implemented by pointers, chains or other mechanisms.

2. Implementation planning: In conjunction with database, decision on the programming language, flowchart, coding and debugging procedure prior to conversion should be made. The operating system limits the programming languages that will run on the system. When the system design is done, the plans and test cases for implementation are required. So there must be detailed schedules for system testing and user training.

3. System test preparation: Each aspect of the system has a separate test requirement. System testing is done after all programming and testing is completed. Acceptance testing is another testing that convinces the user that the candidate system will meet the stated requirements. It is conducted in the presence of users, audit representatives or the entire staff.

4. System interface specification: This phase specifies the user the way in which information should be entered and leave the system. The designer offers the user various options.

5. User Documentation: Before the system is ready for implementation, user documentation in the form of a user or operator's manual must be prepared. The manual provides instructions on how to access, update, or retrieve information, how to display or print output, in what format and so on.

Personnel Allocation: The structured approach is useful in planning process. A completed structure chart gives an idea of the work to be done. Programmers are the assigned appropriately. Programmers are assigned subsystems that are strongly cohesive and loosely coupled. Once modules are allocated, roles are allocated within each team and the designer oversees all the work. Assigning modules are very important. Modules

at the bottom are important because they represent the user interface. So a team with specialized skills should be assigned to such a module.

A well-designed system should have controls to ensure proper operation of the system. A candidate system fails because of the lack of data control. So accuracy, consistency and maintainability is required to eliminate errors. A system design introduces new controls to detect and check for errors. In designing a new system, the designer should specify the location of error-control points and evaluate them according to the basis of error frequency, cost and timing of error detection. With the help of this information they can create procedures for handling the errors.

Designing Conventional Computer Files and Control Modern Computer Databases

File Structure

A file is a collection of data, usually stored on disk. As a logical entity, a file enables you to divide your data into meaningful groups, for example, you can use one file to hold all of a company's product information and another to hold all of its personnel information. As a physical entity, a file should be considered in terms of its organization.

Some of the basic terms used to describe the file hierarchy are as follows

Byte: A byte is an arbitrary set of eight bits that represent a character .It is the smallest addressable unit in today's computers.

Data item (element): One or more bytes are combined into a data item to describe an attribute of an object.

Record: The data items related to an object are combined into a record. There are two types of records

- Logical record maintains logical relationship among all items in the record.
- Physical record is the way in which records are stored in a storage medium.

A collection of related records makes up a file. The size of a file is limited by the size of memory or the storage medium.

File Organisation

File organization is a way of organizing the data or records in a file. It does not refer to how files are organized in folders, but how the contents of a file are added and accessed. There are several types of file organization, the most common of them are

1. Sequential
2. Relative

3. Indexed.

They differ in how easily records can be accessed and the complexity in which records can be organized.

1. Sequential organization

In a sequential file organization, records are organized in the sequence by which they were added. You cannot insert a new record between existing records, but only at the end of the last record. It is a simple file organization that allows you to process batches of records in the file without adding or deleting anything. However, to access a particular record, processing must run through all the other records above it because it does not generate any random key to identify the location of the record. Searching for a record, especially when there are thousands of entries, may be time consuming. Also, inserting or deleting records would mean rearranging the entire sequence.

Data stored on a tape (sequential access) can be accessed only sequentially.

2. Relative or Direct Organization

Another type of organizing files would be relative to the location where the file begins. A relative key is assigned to determine the order of files. The first record would have a relative number of 1, the second record would have a relative number of 2 and so on. It is also called relative, because the sizes of each record may vary, unlike in a sequential organization where the record sizes must be fixed to arrange sequentially. Records may be 128-bytes or 256-bytes in size and they can be arranged relative to each other, with any of the free bytes between them marked as unused. This makes it possible to insert records in those unused areas. Also, with the relative key, you can randomly access any record without starting from the top record. The disadvantage is its dependence on relative keys. If you do not know the relative key of a particular record, you won't be able to randomly access the file.

Data stored on a CD device (direct-access) can be accessed either sequentially or randomly.

3. Indexed

An indexed file organization contains reference numbers, like employee numbers, that identify a record in relation to other records. These references are called the primary keys that are unique to a particular record. Alternate keys can also be defined to allow alternate methods of accessing the record. For example, instead of accessing an employee's record using employee numbers, you can use an alternate key that reference employees by

departments. This allows greater flexibility for users to randomly search through thousands of records in a file. However, it employs complex programming in order to be implemented.

Types of Files used in System

Following are the types of files used in an organization system −

Master file − It contains the current information for a system. For example, customer file, student file, telephone directory.

Table file − It is a type of master file that changes infrequently and stored in a tabular format.

Transaction file − It contains the day-to-day information generated from business activities. It is used to update or process the master file. For example, Addresses of the employees.

Temporary file − It is created and used whenever needed by a system.

Mirror file − are the exact duplicates of other files. Help minimize the risk of downtime in cases when the original becomes unusable. They must be modified each time the original file is changed.

Log files − They contain copies of master and transaction records in order to chronicle any changes that are made to the master file. It facilitates auditing and provides mechanism for recovery in case of system failure.

Archive files − Backup files that contain historical versions of other files.

Database Design

Before the database concept became operational, users had programs that handled their own data independent of other users. It was a conventional file environment with no data integration or sharing of common data across applications. In a data base environment, common data are available and used by several users. Instead of each program managing its own data, data across applications are shared by authorized users with the data base software managing the data as an entity. A program now requests data through the data base management system (DBMS), which determines data sharing.

Objectives of data base

The general theme behind a data base is to handle information as an integrated whole. There is none of the artificiality that is normally embedded in separate files or applications. A data base is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user.

In data base design several specific objectives are considered:

1. Controlled redundancy: Redundant data occupies space and therefore is wasteful. If versions of the same data are in different phases of updating, the system often gives conflicting information. A unique aspect of database design is storing data only once -which controls redundancy and improves system performance

2. Ease of learning and use: A major feature of a user-friendly data base package is how easy it is to learn and use. Related to this point is that a database can be modified without interfering with established ways of using the data.

3. Data independence: An important data base objective is changing hardware and storage procedure or adding new data without having to rewrite application programs. The database should be tunable to improve performance without rewriting programs.

4. More information at low cost: Using, storing, and modifying data at low cost are important. Although hardware prices are falling, software and programming costs are on the rise. This means that programming and software enhancements should be kept simple and easy to update.

5. Accuracy and integrity: The accuracy of a database ensures that data quality and content remain constant. Integrity controls detect data inaccuracies where they occur.

6. Recovery from failure: With multiple users accessing a database, the system must recover quickly after it is down with no loss of transactions. This objective also helps maintain data accuracy and integrity.

7. Privacy and security: For data to remain private, security measures must be taken to prevent unauthorized access. Data base security means that data are protected from various forms of destruction; users must be positively identified and their actions monitored.

8. Performance: This objective emphasizes response time to inquiries suitable to the use of the data. How satisfactory the response time is depends on the nature of the user- database dialogue.

The terms that are normally used in DBMS are

1. **User's view** is a profile that the user expects to see on the report.
2. **Processing** refers to the changes made.
3. **Data model** is a framework of the user's view.
4. **Data file** is the area where the file is stored.

In a data base environment, the DBMS is the software that provides the interface between the data file on disk and the program that requests processing. DBMS stores and manages data. The procedure is as follows:

DDL

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

- CREATE – to create database and its objects like (table, index, views, store procedure, function and triggers)
- ALTER – alters the structure of the existing database
- DROP – delete objects from the database
- TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed
- COMMENT – add comments to the data dictionary
- RENAME – rename an object

DML

DML is short name of Data Manipulation Language which deals with data manipulation, and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in database.

- SELECT – retrieve data from the a database
- INSERT – insert data into a table
- UPDATE – updates existing data within a table
- DELETE – Delete all records from a database table
- MERGE – UPSERT operation (insert or update)
- CALL – call a PL/SQL or Java subprogram
- EXPLAIN PLAN – interpretation of the data access path
- LOCK TABLE – concurrency Control

DCL

DCL is short name of Data Control Language which includes commands such as GRANT, and mostly concerned with rights, permissions and other controls of the database system.

- GRANT – allow users access privileges to database
- REVOKE – withdraw users access privileges given by using the GRANT command
- TCL
- TCL is short name of Transaction Control Language which deals with transaction within a database.

- COMMIT – commits a Transaction
- ROLLBACK – rollback a transaction in case of any error occurs
- SAVEPOINT – to rollback the transaction making points within groups
- SET TRANSACTION – specify characteristics for the transaction

Functions performed by the DBMS:

1. Storing, retrieving, and updating data.

2. Creating program and data independence. Data can be altered independently of the other.

3. Enforcing procedures for data integrity.

4. Reducing data redundancy. Data are stored and maintained only once.

5. Proving security facilities for defining users and enforcing authorizations. Access is limited to authorized users by passwords or similar schemes.

6. Reducing physical storage requirements by separating the logical and physical aspects of the database.

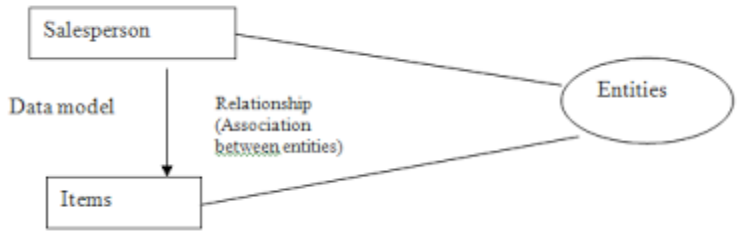
Logical and Physical View of Data

Logical and physical views of data:

In data base design, several views of data must be considered along with the persons who use them. In addition to data structuring, where relationships are reflected between and within entities, we need to identify the application program's logical views of data within an overall logical data structure. The **logical view** is what the data look like regardless of how they are stored. The **physical view** is the way the data exists in physical storage. It deals with how data are stored, accessed or related to other data in storage.

Data Structure

Data are structured according to the data model. In this example sales items are linked to the salesperson who sold them. The salesperson is called an entity and the item sold is also an entity. An entity is a conceptual representation of an object. Relationships between entities make up a data structure. A data model represents a data structure that is described to the DBMS in DDL.

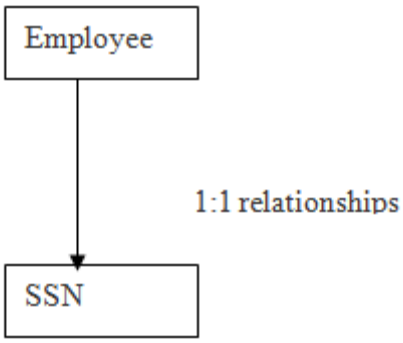


Types of relationships:

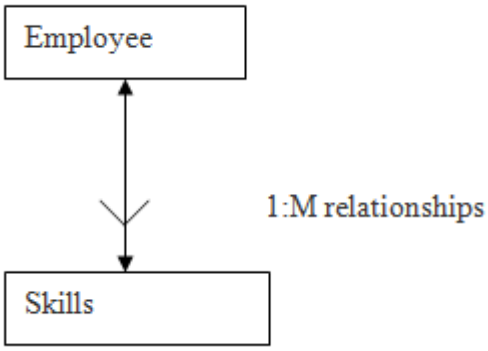
Three types of relationships exist among entities:

1. one-to-one
2. one-to-many
3. many-to-many

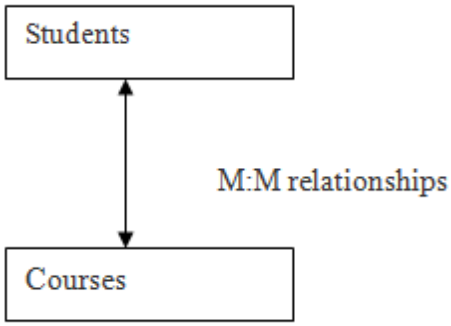
1. A one-to-one(1:1) relationship is an association between two entities. For ex an employee has one social security number.



2. A one-to-many (1:m) relationship describes an entity that may have two or more entities related to it. For ex a father may have many children, and an employee may have many skills.



3. A many-to-many (M:M) relationship describes entities that may have many relationships in both directions. For ex, children may have many toys and students may have many courses.



Procedures and Controls

There are various methods to control processing activities. Batch control is one such method where records are batched into small groups to control totals. Thus in batch processing, as soon as the error is detected, the batch that contains the errors are reviewed without disturbing other batches. The other programmed checks involved are

1.Completeness check ensures that all fields in a record are present and are read in the proper sequence. In a multiple record check, the program verifies the number of the records that make up a transaction. If an error is detected then the entire group of records is rejected.

2. Consistency check refers to the relevance of one type of data to another.

3. Reasonableness check evaluates a transaction against a standard to determine whether it meets the test.

4. Sequence check verifies that data records are in a sequence prior to processing. A check to identify the duplicate records can also be used.

Computer Input and Control

Input Design

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. Input design is the process of converting user-originated inputs to a computer-based format. The information for the input design is obtained from the data flow diagram and the system flowchart.

Objectives of input design

The goal of designing input design is to make data entry as easy, logical, and free from errors as possible.

There are main five objectives as follows : -

- **Controlling amount of input**: design should control the quantity of data for input. Reducing the data requirement can lower costs by reducing labour expense. By reducing input requirement, the analyst can speed the entire process from data capture to providing results to user.
- **Avoiding delay:** A processing delay resulting from data preparation or data entry operation is called bottleneck. Avoiding bottleneck should always be one objective of the analyst in designing input.
- **Avoiding errors in data**: The third objective deals with errors. In one sense, the rate at which errors occur depends on the quantity of data, since the smaller the amount of data to input, the fewer the opportunities for error.
- **Avoiding extra steps**: Sometimes the volume of transactions and amount of data preparation cannot be controlled. When the volume of the transaction cannot be reduced, the analyst must be sure the process is as efficient as possible. For example, the effect of saving or adding a single step when feeding checks into the banking process is multiplied many times over in the course of a working day.
- **Keeping the process simple**: The best method is to achieve all of the objectives mentioned in the simplest manner possible. Simplicity works, and it is accepted by users. Complexity should be avoided when there are simple alternatives.

Data capture guidelines

1. What data in a transaction are important and should be collected for input and processing. The analyst should capture only those items that must actually be input. There are two types of data that must be inputted when processing transactions as follows
    - **Variable data** : Those data items that change for each transaction handled or decision made.
    - **Identification data:** The element of data that uniquely identifies the item being processed. The identifying data in each transaction record is called record key.
2. Data that should not be entered by the user at the time of input must be determined by the analyst. Some of those are as follows
    - **Constant data** : Data that are same for every record or entry.
    - **Details that system can retrieve**: Stored data that are quickly retrieved from system files.
    - **Details that the system can calculate**: Results that can be produced by the system, using the combination of stored data and entered data.

Input media and devices

Source data are input into the system in a variety of ways. The following media and devices are suitable for operation:

1. Punch cards are either 80 or 96 columns wide. Data are arranged in a sequential and logical order. Operations use a keypunch to copy data from source documents onto cards. This means that the source document and card design must be considered simultaneously.

2. Key-to-diskette is modeled after the keypunch process. A diskette replaces the card and stores op to 325,000 characters of data-equivalent to the data stored in 4050 cards. Like cards, data on diskettes are stored in sequence and in batches.

3. MICR translates the special fonts printed in magnetic ink on checks into directed computer input.

4. Mark-sensing readers automatically convert pencil marks in predetermined locations on a card to punched holes on the same card.

5. Optical character recognition (OCR) readers are similar to MICR readers, except that they recognize pencil, ink, or characters by their configuration rather than their magnetic pattern. They are often used in remote locations as free standing input preparation devices

Code design, Computer Based Methods

Design of source document

The source document is form on which the data are initially captured. Source data is initially captured on paper or a source document. Example: A cheque written against an account. Source documents can be entered into the system through punch cards, diskettes or through keyboards.

To design the source document, the analyst must

- First decide what data must be captured.
- Develop the layout of the document showing what items should be included and where they should be placed. The documents includes not only a place for data, but also contains and information telling the user how to complete the form and what information to provide.

Layout

The layout organizes the document by placing important information and establishing the appropriate sequence of items.

- Most people fill in document in from left to right and from top to bottom; the source document should be design in the same way.
- It should be possible for the user to provide information by following a logical sequence rather than by having to skip to different locations on the document.

- A well designed form will ask for each item of data only once, there are very few occasions where user has to feed the same information more than once.
- Forms that do not allow enough space to provide the information requested will not be completed correctly by users. The analyst must also consider how the form will be completed in judging the amount of space between lines.
- The actual document layout shows the position of each item of data and all headings and instructions to users.

Captions

Captions on source document tell the user what data to provide and when they should be entered.

- The caption should be brief but easily understood, with the standard terms that all persons use.
- Abbreviations generally should be avoided.
- Include simple example that will help the user how to feed the data.

A well designed document is easily completed and also speeds the process of recording data. If checkmarks and boxes will be sufficient for capturing data the respondents should not be asked to write longer responds.

Coding methods

Since information systems projects are designed with space, time, and cost saving in mind, coding methods, in which, words, ideas, or relationships are expressed by a code, are developed to reduce input control errors and speed the entire process.

A code is a brief number, title or symbols used instead of more lengthy or ambiguous description. When an event occurs the details of the event often are summarized by a code. The following are some coding techniques:

1. **Classification code**: classification code place separate entities, such as events, people or objects into distinct groups called classes. A code is used to distinguish one class from another.

   For example in an admission system to enter the category, it is categorized as SC – 1, ST – 2, BC – 3 and GE – 4.

   Classification code vastly simplifies the input process because only a single digit code is required. The need of writing lengthy descriptions or making judgment is eliminated.

2. **Function code**: It states the activity or work to be performed without spelling it in detail or narrative statements.

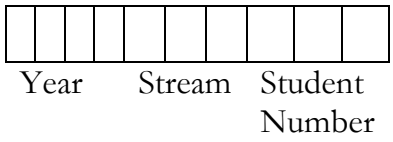
   This type of code is frequently used in transaction data to tell the system how to process the data. For example in file processing, addition of

records in one transaction by means of A, modification of transaction by M, Deletion of transaction by D, sorting of all the transaction by S.

3. **Sequence code**: Sequence codes are either number or letters assigned in series. They tell the order in which events have occurred.

   For example the banking system must keep the track of the transaction so that it is clear which transactions to process first, which second and so on.

4. **Significant digit subset codes**: A well conceived coding scheme, using significant digit subset codes, can provide a wealth of information to user and management. The code can be divided into subsets or sub codes, characters that are part of identification number and that have special meaning. The sub code gives the user additional information. Consider the following example Year Stream Student number



   In the above figure first four boxes is for the year, next three boxes for stream (i.e. science, commerce, etc.) and last four boxes for student number.

5. **Mnemonic codes**: It uses letters and symbols from the product to describe it in a way that communicate visually.

   For example to describe a 21-inch Black and white TV, a successful code set as TV-BW-21. It is difficult to confuse the mnemonic TV with that of other products. Universities frequently use mnemonic code MCA for Master of Computer Application.

Thus the data transaction though coding reduces the volume of data for input and simplifies the process, and lessens the likely hood of errors. Code selection depends on the nature and the objectives of the analyst.

or direct input media to the system.

6.Optical bar code readers detect combination of marks that represent data. The most widely known system is the Universal product code (UPC), Which codes retail items in stores. Automatic tag reading is a major breakthrough in speeding up customer service and eliminating costly data input errors at a point of sale.

7.Cathode-ray tube (CRT) screens are used for online data entry. CRT Screen display 20,40, or 80 characters simultaneously on a television like screen. They show as many as 24 lines of data.

Computer Outputs and Controls

Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts. Computer output is the most important and direct source of information to the user. The output should be efficient and intelligible and help the user in decision-making. A major form of output is a hard copy from computer. Printout should be designed around the output requirements of the user. The following media devices are available for providing computer-based output:

1. MICR readers.
2. Line, matrix, and daisy wheel printers
3. Computer Output Microfilm (COM)
4. CRT screen display
5. Graph plotters.
6. Audio responses

In addition to deciding on the output device the systems analyst must consider the print format and the editing for the final printout. Editing ranges from suppressing zeroes to merging selected records.

Objectives of Output Design

The objectives of input design are

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end users requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions

Types of outputs

External Outputs

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients. Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

Every printed output should have the following

- Each output should have a name or title.
- Provide a sample of the output layout.
- State the content of each field whether it has zeroes, spaces or characters.
- Specify the procedure for proving the accuracy of output data.

Internal outputs

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

There are three types of reports produced by management information −

Detailed Reports − contain present information which has almost no filtering or restriction generated to assist management planning and control.

Summary Reports − contain trends and potential problems which are categorized and summarized that are generated for managers who do not want details.

Exception Reports − contain exceptions, filtered data to some condition or standard before presenting it to the manager, as information.

Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and the verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

Forms Design

Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading. For example, order forms, employment and credit application, etc.

During form designing, the designers should know −

- who will use them
- where would they be delivered
- the purpose of the form or report

During form design, automated design tools enhance the developer's ability to prototype forms and reports and present them to end users for evaluation.

Objectives of Good Form Design

A good form design is necessary to ensure the following −

- To keep the screen simple by giving proper sequence, information, and clear captions.
- To meet the intended purpose by using appropriate forms.
- To ensure the completion of form with accuracy.
- To keep the forms attractive by using icons, inverse video, or blinking cursors etc.
- To facilitate navigation.

Types of Forms

Flat Forms

- It is a single copy form prepared manually or by a machine and printed on a paper. For additional copies of the original, carbon papers are inserted between copies.
- It is a simplest and inexpensive form to design, print, and reproduces

Unit Set/Snap out Forms

- These are papers with one-time carbons interleaved into unit sets for either handwritten or machine use.
- Carbons may be either blue or black, standard grade medium intensity. Generally, blue carbons are best for handwritten forms while black carbons are best for machine use.

Continuous strip/Fanfold Forms

- These are multiple unit forms joined in a continuous strip with perforations between each pair of forms.
- It is a less expensive method for large volume use.

No Carbon Required (NCR) Paper

- They use carbonless papers which have two chemical coatings (capsules), one on the face and the other on the back of a sheet of paper.
- When pressure is applied, the two capsules interact and create an image.

Unit 5
System Testing, Implementation and Maintenance

Implementation is a process of ensuring that the information system is operational. It involves −

- Constructing a new system from scratch
- Constructing a new system from the existing one.

Implementation allows the users to take over its operation for use and evaluation. It involves training the users to handle the system and plan for a smooth conversion.

Software Testing

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Software Validation

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – "Are we developing the product which attempts all that user needs from this software?"
- Validation emphasizes on user requirements.

Software Verification

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specification and methodologies.

- Verification ensures the product being developed is according to design specifications.
- Verification answers the question– "Are we developing this product by firmly following all design specifications?"
- Verifications concentrate on the design and system specifications.

Target of the test are -

- **Errors** - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.

- **Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.

- **Failure** - failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

Manual Vs Automated Testing

Testing can either be done manually or using an automated testing tool:

- **Manual** - This testing is performed without taking help of automated testing tools. The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager.

  Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

- **Automated** This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools.

A test needs to check if a webpage can be opened in Internet Explorer. This can be easily done with manual testing. But to check if the web-server can take the load of 1 million users, it is quite impossible to test manually.

There are software and hardware tools which helps tester in conducting load testing, stress testing, regression testing.

Testing Approaches

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analyzed.

Exhaustive tests are the best-desired method for a perfect testing. Every single possible value in the range of the input and output values is tested. It is not possible to test each and every value in real world scenario if the range of values is large.

Black-box testing

It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

Black-box testing techniques:

- **Equivalence class** - The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
- **Boundary values** - The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
- **Cause-effect graphing** - In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
- **Pair-wise Testing** - The behavior of software depends on multiple parameters. In pair-wise testing, the multiple parameters are tested pair-wise for their different values.
- **State-based testing** - The system changes state on provision of input. These systems are tested based on their states and input.

White-box testing

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing.

In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

The below are some White-box testing techniques:

- **Control-flow testing** - The purpose of the control-flow testing to set up a test case which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.

- **Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

Testing Levels

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping on the next stage, a stage is tested, validated and verified.

Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels -

Unit Testing

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

Integration Testing

Even if the units of software are working fine individually, there is a need to find out wheather they work properly when they are integrated

System Testing

The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:

- **Functionality testing** - Tests all functionalities of the software against the requirement.

- **Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.

- **Security & Portability** - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

Acceptance Testing

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

- **Alpha testing** - The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.

- **Beta testing** - After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

Regression Testing

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.

Testing Documentation

Testing documents are prepared at different stages -

Before Testing

Testing starts with test cases generation. Following documents are needed for reference –

- **SRS document** - Functional Requirements document

- **Test Policy document** - This describes how far testing should take place before releasing the product.

- **Test Strategy document** - This mentions detail aspects of test team, responsibility matrix and rights/responsibility of test manager and test engineer.

- **Traceability Matrix document** - This is SDLC document, which is related to requirement gathering process. As new requirements come, they are added to this matrix. These matrices help testers know the source of requirement. They can be traced forward and backward.

While Being Tested

The following documents may be required while testing is started and is being done:

- **Test Case document** - This document contains list of tests required to be conducted. It includes Unit test plan, Integration test plan, System test plan and Acceptance test plan.

- **Test description** - This document is a detailed description of all test cases and procedures to execute them.

- **Test case report** - This document contains test case report as a result of the test.

- **Test logs** - This document contains test logs for every test case report.

After Testing

The following documents may be generated after testing :

- **Test summary** - This test summary is collective analysis of all test reports and logs. It summarizes and concludes if the software is ready to be launched. The software is released under version control system if it is ready to launch.

Testing vs. Quality Control, Quality Assurance and Audit

We need to understand that software testing is different from software quality assurance, software quality control and software auditing.

Software quality assurance - These are software development process monitoring means, by which it is assured that all the measures are taken as per the standards of organization. This monitoring is done to make sure that proper software development methods were followed.

Software quality control - This is a system to maintain the quality of software product. It may include functional and non-functional aspects of software product, which enhance the goodwill of the organization. This system makes sure that the customer is receiving quality product for their requirement and the product certified as 'fit for use'.

Software audit - This is a review of procedure used by the organization to develop the software. A team of auditors, independent of development team examines the software process, procedure, requirements and other aspects of SDLC. The purpose of software audit is to check that software and its development process, both conform standards, rules and regulations.

Combating Resistance to Change

Training

The personnel in the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. The success or failure of well designed and technically elegant systems can depend on the way they are operated and used.

Training Systems Operators

Systems operators must be trained properly such that they can handle all possible operations, both routine and extraordinary. The operators should be trained in what common malfunctions may occur, how to recognize them, and what steps to take when they come.

Training involves creating troubleshooting lists to identify possible problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problems arise.

Training also involves familiarization with run procedures, which involves working through the sequence of activities needed to use a new system.

User Training

- End-user training is an important part of the computer-based information system development, which must be provided to employees to enable them to do their own problem solving.

- User training involves how to operate the equipment, troubleshooting the system problem, determining whether a problem that arose is caused by the equipment or software.

- Most user training deals with the operation of the system itself. The training courses must be designed to help the user with fast mobilization for the organization.

Training Guidelines

- Establishing measurable objectives
- Using appropriate training methods
- Selecting suitable training sites
- Employing understandable training materials

Training Methods

Instructor-led training

It involves both trainers and trainees, who have to meet at the same time, but not necessarily at the same place. The training session could be one-on-one or collaborative. It is of two types −

Virtual Classroom

In this training, trainers must meet the trainees at the same time, but are not required to be at the same place. The primary tools used here are: video conferencing, text based Internet relay chat tools, or virtual reality packages, etc.

Normal Classroom

The trainers must meet the trainees at the same time and at the same place. They primary tools used here are blackboard, overhead projectors, LCD projector, etc.

Self-Paced Training

It involves both trainers and trainees, who do not need to meet at the same place or at the same time. The trainees learn the skills themselves by accessing the courses at their own convenience. It is of two types −

Multimedia Training

In this training, courses are presented in multimedia format and stored on CD-ROM. It minimizes the cost in developing an in-house training course without assistance from external programmers.

Web-based Training

In this training, courses are often presented in hyper media format and developed to support internet and intranet. It provides just–in-time training for end users and allow organization to tailor training requirements.

Conversion

It is a process of migrating from the old system to the new one. It provides understandable and structured approach to improve the communication between management and project team.

Conversion Plan

It contains description of all the activities that must occur during implementation of the new system and put it into operation. It anticipates possible problems and solutions to deal with them.

It includes the following activities −

- Name all files for conversions.
- Identifying the data requirements to develop new files during conversion.
- Listing all the new documents and procedures that are required.
- Identifying the controls to be used in each activity.
- Identifying the responsibility of person for each activity.
- Verifying conversion schedules.

Conversion Methods

The four methods of conversion are −

- Parallel Conversion
- Direct Cutover Conversion
- Pilot Approach
- Phase-In Method

Method	Description	Advantages	Disadvantages
Parallel	Old and new	Provides	Causes cost

Conversion	systems are used simultaneously.	fallback when new system fails. Offers greatest security and ultimately testing of new system.	overruns. New system may not get fair trail.
Direct Cutover Conversion	New system is implemented and old system is replaced completely.	Forces users to make new system work Immediate benefit from new methods and control.	No fall back if problems arise with new system Requires most careful planning
Pilot Approach	Supports phased approach that gradually implement system across all users	Allows training and installation without unnecessary use of resources. Avoid large contingencies from risk management.	A long term phasein causes a problem of whether conversion goes well or not.
Phase-In Method	Working version of system implemented in one part of organization	Provides experience and line test before implementation When preferred	Gives impression that old system is erroneous and it is not reliable.

	based on feedback, it is installed throughout the organization all alone or stage by stage.	new system involves new technology or drastic changes in performance.	

File Conversion

It is a process of converting one file format into another. For example, file in WordPerfect format can be converted into Microsoft Word.

For successful conversion, a conversion plan is required, which includes −

- Knowledge of the target system and understanding of the present system
- Teamwork
- Automated methods, testing and parallel operations
- Continuous support for correcting problems
- Updating systems/user documentation, etc

Many popular applications support opening and saving to other file formats of the same type. For example, Microsoft Word can open and save files in many other word processing formats.

Post-Implementation Evaluation Review (PIER)

PIER is a tool or standard approach for evaluating the outcome of the project and determines whether the project is producing the expected benefits to the processes, products or services. It enables the user to verify that the project or system has achieved its desired outcome within specified time period and planned cost.

PIER ensures that the project has met its goals by evaluating the development and management processes of the project.

Objectives of PIER

The objectives of having a PIER are as follows −

- To determine the success of a project against the projected costs, benefits, and timelines.
- To identify the opportunities to add additional value to the project.
- To determine strengths and weaknesses of the project for future reference and appropriate action.
- To make recommendations on the future of the project by refining cost estimating techniques.

The following staff members should be included in the review process −

- Project team and Management
- User staff
- Strategic Management Staff
- External users

Every system requires periodic evaluation after implementation. A post implementation review measures the system's performance against predefined requirements. Unlike system testing, which determines where the system fails so that the necessary adjustments can be made, a post-implementation review determines how well the system continues to meet performances specifications. It is done after design and conversion are complete. It also provides information to determine whether major redesign is necessary.

Request for Review:

The initiating study begins with the review team, which gathers and reviews requests for evaluation. It also files discrepancy notices after the system has been accepted. Unexpected change in the system that affects the user or system performance is a primary factor that prompts system review. Once a request is filled, the user is asked how well the system is functioning. Suggestions regarding changes and improvements are also sought. This phase sets the stage for a formal post-implementation review.

A Review Plan:

This review team prepares a formal review plan around the objectives of the review, the type of evaluation to be carried out, and the time schedule required.

1. Administrative plan:

The review group probes the effect of the operational system on the administrative procedures of the user.

The following activities are reviewed:

1. **User objectives –** This is an extremely critical area since it is possible that over time either the system fails to meet the user's initial objectives or the user objectives change as a reflection of changes in the organizational objectives. We need to think in terms of problems and of further opportunities. The results of the evaluation are documented for future reference
2. **Operating costs and benefits –** Under the administrative plan, the cost structures of the system is closely reviewed. This includes a review of all costs and savings, a review and update of the non cost benefits of all the system and a current budget designed to manipulate the costs and savings of the system.

2. Personnel Requirement Plan:

This plan evaluates all activities involving system personnel and staff as they directly deal with the system. The emphasis is on productivity and job satisfaction. After the plan is developed, the review group evaluates the following:

1. **Personnel performance** objectives compared with current performance levels: Turnover, tardiness and absenteeism are also evaluated. The results are documented and made available to the maintenance group for follow-up.
2. **Training performance:** Through testing, interviews, and all other data gathering techniques, the review group attempts to answer questions about the adequacy of the training materials.

3. Hardware Plan:

The hardware of the new system is also reviewed, including terminals, CRT screens, software programs, and the communication network. The primary target is a comparison of current performance specifications. The outcome of the evaluation indicates any differences between expectations and realized results. It also points to any necessary modifications to be made.

4. Documentation Review Plan:

The reason for developing a documentation review plan is to evaluate the accuracy and completeness of the documentation compiled to date and its conformity with pre-established documentation standards. Irregularities prompt action where changes in documentation would improve the format and content.

Software Maintenance

Maintenance includes all the activity after the installation of software that is performed to keep the system operational. As we have mentioned earlier, software often has design faults. It is generally agreed that for large systems, removing all the faults before delivery is

extremely difficult and faults will be discovered long after the system is installed. As these faults are detected, they have to be removed

Removing errors is one of the activities of maintenance. Maintenance also needed due to a change in the environment or the requirements of the system. The introduction of a software system affects the working environment. This change in environment often changes what is desired from the system.

Maintenance changes the existing system, enhancement adds features to the existing system, and development replaces the existing system. It is an important part of system development that includes the activities which corrects errors in system design and implementation, updates the documents, and tests the data.

Maintenance Types

System maintenance can be classified into three types −

- Corrective Maintenance − Enables user to carry out the repairing and correcting leftover problems.
- Adaptive Maintenance − Enables user to replace the functions of the programs.
- Perfective Maintenance − Enables user to modify or enhance the programs according to the users' requirements and changing needs.

System Audit

It is an investigation to review the performance of an operational system. The objectives of conducting a system audit are as follows −

- To compare actual and planned performance.
- To verify that the stated objectives of system are still valid in current environment.
- To evaluate the achievement of stated objectives.
- To ensure the reliability of computer based financial and other information.
- To ensure all records included while processing.
- To ensure protection from frauds.

Audit of Computer System Usage

Data processing auditors audits the usage of computer system in order to control it. The auditor need control data which is obtained by computer system itself.

The System Auditor

The role of auditor begins at the initial stage of system development so that resulting system is secure. It describes an idea of utilization of system that can be recorded which helps in load planning and deciding on hardware and software specifications. It gives an indication of wise use of the computer system and possible misuse of the system.

Audit Trial

An audit trial or audit log is a security record which is comprised of who has accessed a computer system and what operations are performed during a given period of time. Audit trials are used to do detailed tracing of how data on the system has changed.

It provides documentary evidence of various control techniques that a transaction is subject to during its processing. Audit trials do not exist independently. They are carried out as a part of accounting for recovering lost transactions.

Audit Methods

Auditing can be done in two different ways −

Auditing around the Computer

- Take sample inputs and manually apply processing rules.
- Compare outputs with computer outputs.

Auditing through the Computer

- Establish audit trial which allows examining selected intermediate results.
- Control totals provide intermediate checks.

Audit Considerations

Audit considerations examine the results of the analysis by using both the narratives and models to identify the problems caused due to misplaced functions, split processes or functions, broken data flows, missing data, redundant or incomplete processing, and non-addressed automation opportunities.

The activities under this phase are as follows −

- Identification of the current environment problems
- Identification of problem causes
- Identification of alternative solutions
- Evaluation and feasibility analysis of each solution
- Selection and recommendation of most practical and appropriate solution
- Project cost estimation and cost benefit analysis

Hardware/software Selection in an information system

Selecting hardware and software for implementing information system in an organization is a serious and time-consuming process that passes through several phases.

The main steps of the selection process are listed below:

1. **Requirement analysis:** System configuration requirements are clearly identified and a decision to acquire the system is taken in this step.
2. **Preparation of tender specifications: -**After studying the feasibility and deciding upon the configuration, tender documents are prepared for the benefit of vendors to clarify the details of various specifications, as listed below.

   Purchase procedure and schedule:

   It includes

   a) Date of tender submission
   b) Evaluation criteria
   c) Scope for negotiations, if any and
   d) Expected usage environment and load pattern

   Equipment specification Detailed technical specifications of each item required for both mandatory and optional items.

   Quotation format:

   a) Format for stating technical details and quoting prices
   b) whether deviations from specifications should be specifically listed
   c) Prices and levies (duties, taxes etc.) could be quoted as lump sum or required separately.
   d) Required validity of the quotation.
   e) Earnest money deposit required, if any.

Proposed terms of contract

a) Expected delivery schedule.
b) Uptime warranties required
c) Penalty clause, if any
d) Payment terms (Whether advance payment acceptable)
e) Arbitrary clauses
f) Training needs.
g) Post warranty maintenance terms expected.

Any additional information required.

3. Inviting tenders: - After the preparation of tender specifications, tenders are invited. Invitation of tenders may depend upon the magnitude of purchase (estimate equipment cost). It may be through

i) Open tender (through newspaper advertisement)
ii) Limited tender (queries sent to a few selected vendors)
iii) Propriety purchase (applies mostly to upgrade requirements)
iv) Direct purchase from market (applies mostly to consumables)

4. Technical scrutiny and short listing:

This step involves the following activities.

i) All tendered bids are opened on a pre-defined date and time.
ii) Deviations from the specifications, if any, in each bid are noted.
iii) A comparative summary is prepared against the list of tendered technical features.

Additional factors to considered are:

i) **Financial health of the vendor**

(from balance sheets)

ii) Nature and extent of support

(from information provided on number of support staff per installed site an cross-check with selected customers)

iii) Engineering quality pf products

(factory inspection of product facilities, QA procedures and R&D)

5. Detailed evaluation of short listed vendors: - This step primarily involves getting any finer technical clarifications. Visits to customer sites and factory inspections may be planned. If any specific performance requirement is stipulated, the offered product is to be examined

at this stage through suitable benchmark tests. For benchmark tests, standard benchmarks may be used as adequate performance indicators.

6. Negotiation and procurement decision: - Because of the extensive competition, computer system vendors may offer significant concessions. Negotiations are held to maximize these concessions. However, price negotiations are often not permitted by some organizations.

When price negotiations are permitted, the committee members should have a good knowledge of the prevailing market prices, current trends, and also the duty/tax structure.

- Computer magazines
- Vendor directories.
- Contact with other users
- Past personal experience.

7. Delivery and installation: - In this step, the vendor delivers the hardware/software to the buyer's organization, where it is matched with the specifications mentioned in the purchase order. If conforms to these specifications, the vendor installs the system in the premises of the organization.

8. Post-installation review: - After the system is installed, a system evaluation is made to determine how closely the new system conforms to the plan. A post-installation review, in which system specifications and user requirements are audited, is made. The feedback obtained in this step helps in taking corrective decision.

Security

System security refers to protecting the system from theft, unauthorized access and modifications, and accidental or unintentional damage. In computerized systems, security involves protecting all the parts of computer system which includes data, software, and hardware. Systems security includes system privacy and system integrity.

- **System privacy** deals with protecting individuals systems from being accessed and used without the permission/knowledge of the concerned individuals.
- **System integrity** is concerned with the quality and reliability of raw as well as processed data in the system.

Control Measures

There are varieties of control measures which can be broadly classified as follows −

Backup

- Regular backup of databases daily/weekly depending on the time criticality and size.
- Incremental back up at shorter intervals.
- Backup copies kept in safe remote location particularly necessary for disaster recovery.
- Duplicate systems run and all transactions mirrored if it is a very critical system and cannot tolerate any disruption before storing in disk.

Physical Access Control to Facilities

- Physical locks and Biometric authentication. For example, finger print
- ID cards or entry passes being checked by security staff.
- Identification of all persons who read or modify data and logging it in a file.

Using Logical or Software Control

- Password system.
- Encrypting sensitive data/programs.
- Training employees on data care/handling and security.
- Antivirus software and Firewall protection while connected to internet.

Risk Analysis

A risk is the possibility of losing something of value. Risk analysis starts with planning for secure system by identifying the vulnerability of system and impact of this. The plan is then made to manage the risk and cope with disaster. It is done to accesses the probability of possible disaster and their cost.

Risk analysis is made by a team of experts with different backgrounds like chemicals, human error, and process equipment.

The following steps are to be followed while conducting risk analysis −

- Identification of all the components of computer system.

- Identification of all the threats and hazards that each of the components faces.
- Quantify risks i.e. assessment of loss in the case threats become reality.

Risk Analysis – Main Steps

As the risks or threats are changing and the potential losses are also changing, management of risk should be performed on periodic basis by senior managers.

Risk management is a continuous process and it involves the following steps −

- Identification of security measures.
- Calculation of the cost of implementation of security measures.
- Comparison of the cost of security measures with the loss and probability of threats.
- Selection and implementation of security measures.
- Review of the implementation of security measures.

Recovery

Recovery/Restart Requirements:

Restoring a damaged database is done by roll forward or rollback procedure.

Rollforward method involves updating a valid copy of the database with the necessary changes to produce a current version of the database.

Rollback method starts with the current invalid state and removes the records of the activity to produce the prior valid state of the database.

Backup is essential for recovery/restart procedure. If the database is physically damaged then it cannot be rolled back. Only roll forward can be done. For a sequential file a grandfather-father-son approach is followed. In a database environment, master files are not copied as they are updated.. Instead transactions are posted directly to the file which replaces the original data. So to recover documents in such type of storage, backup is required.

System failures and recovery:

There are three types of failures

1.Catastrophic failure is one where part of the database is unreadable. To restore from this failure, use 'rollback' method of recovery.

2. Logical error occurs when the activity of the database is interrupted with no chance to complete the current transactions. So when the system runs again, it is not sure if the changes have been updated or not. Data though available may be inaccurate. To restore the original contents, rollback method is used.

3. Structural damage An example is a pointer incorrectly stored in a record that points to an unrelated or nonexistent data. If the problem cannot be corrected by software utility, then the database must be recovered to the most recent up-to-date point before the damage occurred.

Disaster/Recovery Planning

Disaster/recovery planning is a means of addressing the concern for system availability by identifying potential exposure, prioritizing applications and designing safeguards and minimizes loss if a disaster occurs. There are several alternatives. They range from having an entire facility in one location with a complete redundancy of hardware to leasing a site with no computer but adequate electricity and air conditioning to support a computer facility on temporary basis. After an alternative has been determined a decision must be made about the applications to be processed. The hardware to process the applications and what should be relocated after the disaster. In disaster/recovery the management's role is to accept the plan select an alternative and recognize the benefits.

The user's responsibilities are as follows:

1. Identify critical applications why they are critical and how computer unavailability would affect the department.

2. Approving data protection procedures.

3. Funding the cost of backup.

Disaster/Recovery Plan

When a disaster/recovery procedure is planned, several questions have to be answered:

1. The time taken to rebuild the computer center or aspects of it
2. The type of accommodation should we look for in a backup installation. How quickly is it available?
3. The equipment is needed to keep the corporation functioning
4. How would reports be transmitted to the user? That is, is there going to be telecommunications network or simply a courier service?
5. The utilities(electric power, air conditioning, etc) are required when a disaster occurs
6. Would there be sufficient experienced staff available for proper recovery?

When these questions are answered and management gives its support for a disaster/recovery plan, the next step is to initiate a plan that involves four phases:

1. Appoint a disaster/recovery team and a team coordinator to develop the plan or procedure.
2. Prepare planning task.
3. Compile a disaster/recovery manual.
4. Dummy run to test the procedure.

Disaster/Recovery Team

A disaster/recovery team should include a cross section of system designers, users, and computer operators. Under the leadership of a coordinator, the team's main functions are to organize the project, monitor progress on the plan, and oversee its completion. The team meets periodically to ensure that the plan is kept up to date, considers new vulnerabilities or exposures to loss, and implements new technology or procedures as needed. More specifically, the objectives of a disaster/recovery team include the following:

1. Secure backup sites for occupation and use.
2. Contract for hardware to meet minimum processing needs.
3. Supply working copies of all operating systems and application programs to meet minimum processing requirements.
4. Supply communication facilities to make reports promptly available to the user.
5. Supply consumables and administrative support.

Disaster/Recovery Planning

Disaster/recovery planning tasks are prepared in a cycle similar to that of system development. Briefly, the cycle entails the following:

1. Definition phase sets the objectives of the disaster/recovery project.
2. Requirements phase evaluates applications against disaster. Recovery Objectives, determines what is to be included in the plan, and specifies priorities. The team takes inventor of the hardware, software, telecommunications, backup and clerical procedures, utilities, and personnel assignments. Design phase evaluates design alternatives, potential vendors, and prices and chooses the final design.
3. Testing and implementation phase runs backup systems, compares results, and correct errors. During implementation, procedures are written, sites are prepared, and maintenance plans are developed.

Disaster/ Recovery Manual

Once the team has completed the assignment, a disaster/recovery manual is prepared and copies are made available to team members and management.

Information Systems and Ethics

Information systems bring about immense social changes, threatening the existing distributions of power, money, rights, and obligations. It also raises new kinds of crimes, like cyber-crimes.

Following organizations promote ethical issues:

- The Association of Information Technology Professionals (AITP)
- The Association of Computing Machinery (ACM)
- The Institute of Electrical and Electronics Engineers (IEEE)
- Computer Professionals for Social Responsibility (CPSR)

The ACM Code of Ethics and Professional Conduct

- Strive to achieve the highest quality, effectiveness, and dignity in both the process and products of professional work.
- Acquire and maintain professional competence.
- Know and respect existing laws pertaining to professional work.
- Accept and provide appropriate professional review.
- Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis and possible risks.
- Honor contracts, agreements, and assigned responsibilities.
- Improve public understanding of computing and its consequences.
- Access computing and communication resources only when authorized to do so.

The IEEE Code of Ethics and Professional Conduct

IEEE code of ethics demands that every professional vouch to commit themselves to the highest ethical and professional conduct and agree:

- To accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;
- To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
- To be honest and realistic in stating claims or estimates based on available data;
- To reject bribery in all its forms;
- To improve the understanding of technology, its appropriate application, and potential consequences;
- To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
- To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
- To treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;
- To avoid injuring others, their property, reputation, or employment by false or malicious action;
- To assist colleagues and co-workers in their professional development and to support them in following this code of ethics