SYSTEM DESIGN CONCEPT

BCA-302

Authors

Ashish Chandra Swami
M.Tech, MCA, DB2 Certified (IBM-USA)
Head & Asso. Proff.
MCA Institute
S.S. Jain Subodh
P.G. (Autonomous) College,
Jaipur

Dr. Vikram Jain
Ph.D., M. Tech, MCA, E-commerce
Certified (IBM-USA)
Microsoft Certifiedd Application Developer
Lecturer, MCA, Institute,
S.S. Jain SubodhP.G. (Autonomous)
College, Jaipur

New Edition
2019

Syllabus

System Concept Design

Paper 302

Question Paper pattern for Main University Examination **Max. Marks : 100**

Part-I (very short answer)

Consists 10 questions of two marks each two questions from each unit. Maximum limit for each question is up to 40 words.

Part-II (short answer)

Consists 5 questions of four marks each with one questions from each unit. Maximum limit for each question is up to 80 words.

Part-III (Long answer)

Consists 5 questions of twelve marks each with one question from each unit with internal choice.

UNIT -1

Introduction to Systems Design Environment:

Systems Development Approaches-Function Oriented. Data Oriented, Object Oriented. Development Process, Methodologies, Tools, Modeling Methods, Processing Types and Systems, Batch Processing, Realtime Processing.

System Development Life Cycle, Linear or Waterfall Cycle, Linear cycle phase problem definition, system specification, system design, system development, testing, maintenance Problems with Linear Life Cycle, Iterative Cycles, Spiral model Requirements analysis, Importance of Communication, Identifying Requirements, Data and Fact Gathering Techniques, Feasibility Studies, Introduction to Prototyping, Rapid Prototyping Tools, Benefits of prototyping.

1

INTRODUCTION TO SYSTEM DESIGN CONCEPT

Objective

1. What is System Analysis and Design?
2. Role of System Analyst
3. Who are the users of the Systems?

Any change in the existing policies of an organization may require the existing information system to be restructured or complete development of a new information system. In case, an organization functioning manually and planns to computerize its functioning, the development of a new information system would be required. Development and designing of any information system can be put into two major phases :

(i) Analysis (ii) Design.

Analysis phase observe the complete functioning of the system requirements are defined.

On the basis of this it will lead to designing of a new system.

Hence the development process of a system is also known as System Analysis and Design process. So let us now discuss in detail

a) What exactly System Analysis and Design is ?

b) Who is System Analyst and what are his various responsibilities ?

c) Users of the Systems ?

1. What is System Analysis and Design?

System development can generally be thought of having two major components: systems analysis and systems design.

In System Analysis more emphasis is given on understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system. Fig: 1 shows the various stages involved in building of new or an improved system.

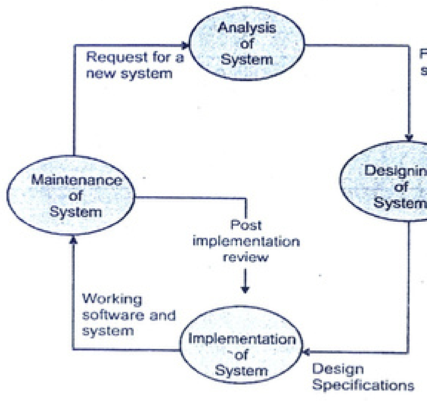


Fig. 1 : Stages in building an improved system

System design is the process of planning a new business system or one to replace or complement an existing system.

Analysis specifies what the system should do. Design states how to accomplish the objective. After the proposed system is analyzed and designed, the actual implementation of the system occurs. After implementation, working system is available and it requires timely maintenance, and post implementation changes.

2. System Analyst :

The system analyst is the person (or persons) who guides through the development of an information system. In performing these tasks the analyst must always match the information system objectives with the goals of the organization. Role of System Analyst differs from organization to organization.

2.1 Responsibilities of System Analyst :

Most common responsibilities of System Analyst are as following :

(a) Change Agent : The analyst may be viewed as an agent who will bring change in the organisation. A candidate system is designed to introduce change and reorientation in how the user organization handles information or makes decisions. It is important, that change be accepted by the user. The way to 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

to Imposer. In between, there are the catalyst and the confronter roles. When the user appears to have a tolerance for change, the persuader or catalyst (helper) 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, however, the goal is the same to achieve acceptance of the candidate system with a minimum of resistance.

(b) Investigator and Monitor : In defining a problem, the analyst pieces together the information gathered to determine why the present system does not work well and what changes will correct the problem. In one respect, 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 quality.

(c) Architect : The architect's primary function is to act as middle man between the client's abstract design requirements and the contractor's detailed building plan may be compared to the analyst's role as liaison between the user's logical design requirements and the detailed physical system design. At architect, the analyst also creates a detailed physical design of candidate systems. The analyst aids users in formalizing abstract ideas and provides details to build the end product – the candidate system.

(d) Psychologist : In system development, systems are built around people. This is perhaps a bit exaggerated, but the analyst plays the role of a psychologist in the way he/she reaches people, interprets their thoughts, assesses their behavior and draws conclusions from these interactions. Understanding inter functional relationships is important. It is important that the analyst 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.

(e) 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, then, are crucial to the success of the system.

(f) Motivator : A candidate system must be well designed and acceptable to the user. System acceptance is achieved through user participation in its development, effective user training and proper motivation to use the system. The analyst's role as a motivator becomes obvious during the first few weeks after implementation and during times when turnover results in new people being trained to work with the candidate system. The amount of dedication it takes to motivate users often taxes the analyst's abilities to maintain the pace. What was once viewed as a challenge can easily become a frustration if the user's staff continues to resist the system.

(g) Politician : Related to the role of motivator is that of politician. In implementing a candidate system, the analyst tries to appease all parties involved.

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/her consistency, so is the analyst's goal to have the support of the user's staff. He/she represents their thinking and tries to achieve their goals through computerization.

2.2 Interpersonal Skills Required in Systems Analyst :

An analyst must possess various skills to effectively carry out the job. Specifically, they can be divided into two categories: interpersonal and technical skills. Both are required for system development.

Interpersonal 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. Technical skills, on the other hand, focus on procedures and techniques for operations analysis, systems analysis and computer science.

The interpersonal skills relevant to systems work include the following:

(a) Communication - having the ability to articulate and speak the language of the user, a flare for mediation and the ability for working with virtually all managerial levels in the organization. Some indicators of a climate of closed communication are defensive memos, excessive correspondence and a failure to speak up for fear of being identified. Therefore, opening communication channels are a must for system development.

(b) Understanding - 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.

(c) Teaching - educating people in use of computer systems, selling the system to the user and giving support when needed.

(d) Selling - selling ideas and promoting innovations in problem solving using computers.

2.3 Technical Skills include:

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

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

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

(d) Dynamic interface - blending technical and non technical considerations in functional specifications and general design.

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

(f) Knowledge of the basics of the computer and the business function.

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

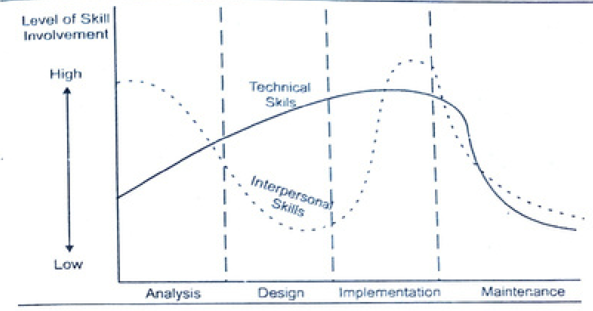


Fig. 2: Interpersonal & Technical Skills required for System Development

The above figure illustrates the skills expected in a systems analyst across the phases of system development.

3. Phases of Development

1. Analysis : It includes system's study in order to get facts about business activity. It is about getting information-and determining requirements. Here the responsibility includes only requirement determination, not the design of the system.

2. Analysis and Design : Apart from the analysis work, Analyst is also responsible for the designing of the new system/application.

3. Analysis, Design, and Programming : Analyst is also required to work as a programmer, where he actually writes the code to implement the design of the proposed application.

Due to the various responsibilities that a system analyst requires to handle, he has to be multifaceted person with all round skills required at various stages of the life cycle. In addition to the technical knowhow of the information system, development of a System Analyst should also have the following knowledge.

Business Knowledge: As the analyst might have to develop any kind of a business system, he should be familiar with the general functioning of all kind of businesses.

Interpersonal Skills: Such skills are required at various stages of development process for interacting with all the users and extracting the requirements out from them.

Problem Solving Skills : A system analyst should have enough problem solving skills for defining the alternate solutions to the system and also for the problems occurring at the various stages of the development process so in case of any misshaping alternate solutions are available.

4. Users of System :

(a) Systems Designer : A systems designer carries out systems design. Should be an expert in computer technology being used. The primary interaction is with the programmers to get the system developed as per his design.

(b) Programmer : A programmer develops programs according to the design specifications received from the systems designer. Experience in the language. DBMS being used for developing the system is preferable. A programmer is also responsible for testing and debugging the modules developed by him.

The programmers who develop the higher level modules may also be responsible for integrating the lower level modules into an integrated sub-system or the complete system. Integration may involve the testing and debugging of a sub-system or the complete system

(c) Technical Writer : A technical writer is supposed to develop the user and operations manual. He may also be involved in editing of all other documents produced during systems development

(d) User : A user is a person or a group of persons for whom the system is being developed. The user is involved during a number of steps in systems development, except for design, coding, testing and debugging. His role is extremeiy important during feasibility study and systems analysis

(e) Project Manager : Project manager is the person who is the overall incharge of the entire systems development project. He should be good manager with abilities to- plan, communicate, coordinate and control. He should also be able to project the resource and time requirements for developing the system.

(f) Management : Management here refers to the management of the organization for which the system is being developed. The management is responsible for sanctioning the development project and the resources required by it. The management at any stage may alter the parameters of the project and may even cancel a project at an advanced stage.

(g) Operations Staff : Operations staff is the group of persons who will be responsible for operating the new system. They are also referred to as the EDP or the MIS staff.

(h) User of System : Users of the system refer to the people who use computers to perform their jobs like desktop operators these users are known as end user. Further, end users can be divided into various categories.

Very first users are the hands on users who will actually interact with the system. They are the people who feed in the input data get output data. Like person at the booking counter of a gas authority. This person actually sees the records and registers requests from various customers for gas cylinders.

Other users are the indirect end users who do not interact with the systems hardware and software. However, these users benefit from the results of these systems. These types of users can be managers of organization using that system.

There are third types of users who have management responsibilities for application systems. These oversee investment in the development or use of the system.

Another types of users are senior managers. They are responsible for evaluating organization's exposure to risk from the system failure.

Now as we have discussed that, what are systems and what is system analysis and design, so let us take a case in which we'll apply the concepts we have learned in the chapter.

5. CASE STUDY : XYZ Public Library :

XYZ Public Library is the biggest library in Jaipur. Currently it has about 1500 members. A person who is 18 or above can become a member. There is a membership fee of Rs. 1,000 for a year. There is a form to be filled in which person fills personal details. These forms are kept in store for maintaining members' records and knowing the membership period.

A member can issue a maximum of three books. He/she has three cards to issue books. Against each card a member issue one book from library. Whenever a member wishes to issue a book and there are spare cards, then the book is issued. Otherwise that request is not entertained. Each book is to be returned on the specified due date. If a member fails to return a book on the specified date, a fine of Rs 5 per day after the due return date is charged. If in case a card gets lost then a duplicate card is issued. Accounts are maintained for the membership fees-and money-collected from the fines. There are two librarians for books return and issue transaction. Approximately 300 members come to library daily to issue and return books.

There are 80,000 books available out of which 5000 books are for reference and can not be issued. Records for the books ill the library are maintained. These records contain details about the publisher, author, subject, language, etc. There are suppliers that supply books to the library. Library maintains records of these suppliers.

Many reports are also produced. These reports are for details of the books available in the library, financial details, members' details and supplier's details.

Currently all functions of the library are done manually. Even the records are maintained on papers. Now day by day members are increasing. Maintaining manual records is becoming difficult task. There are other problems also that the library staff is facing. Like in case of issue of duplicate cards to a member when member or library staff loses the card. It is very difficult to check the genuinity of the problem.

Sometimes the library staff needs to know about the status of a book as to whether it is issued or not. So to perform this kind of search is very difficult in a manual system.

Also management requires reports for books issued, books in the library, members, and accounts. Manually producing the reports is a cumbersome job when there are hundreds and thousands of records.

Management plans to expand the library, in terms of books, number of members and finally the revenue generated. It is observed that every month there are at least 100-150 requests for membership. For the last two months the library has not entertained requests for the new membership as it was difficult to manage the existing 1500 members manually. With the expansion plans, the management of the library aims to increase its members at the rate of 200 per month. It also plans to increase the membership fees from 1000 to 2000 for yearly and 1000 for half year, in order to provide its members better services, which includes increase in number of books from 3 to 5.

Due to the problems faced by the library staff and its expansion plans, the management is planning to have a system that would first eradicate the needs of cards. A system to automate the functions of record keeping and report generation. And which could help in executing the different searches in a faster manner. The system to handle the financial details.

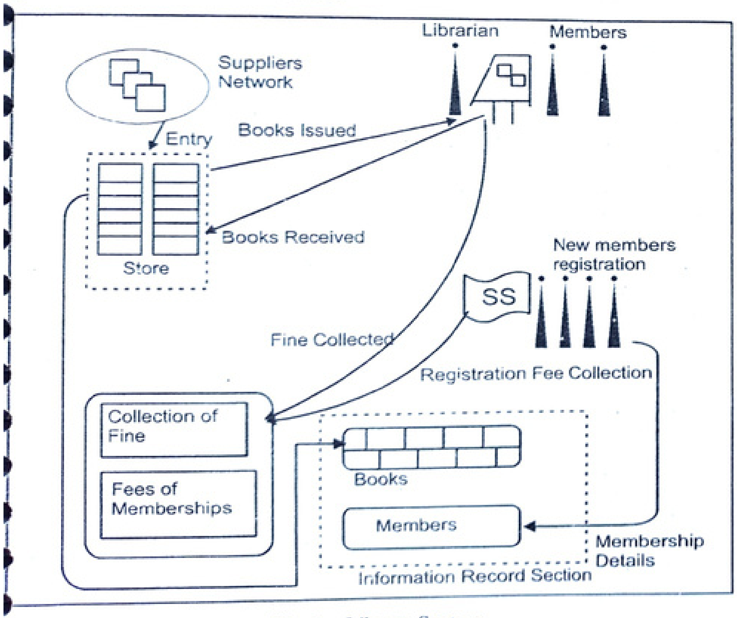


Fig. 3 : Library System

Applying the concepts studied in the chapter to the case study : The first thing we studied is systems. In our case study XYZ Public Library is our system. Every system is a set of some functional units that work together to achieve some objective. The main objective of library system is to provide books to its members without difficulty. Fig: 3 depicts our library system pictorially.

Our system has many functional units. Books issue and return section, books record unit members record unit, account and generation units are the different. Functional unit of the library. Each functional unit has its own task. However, each of these work independently to achieve the overall objective of the library.

Later in the session, we talked about different components and characteristics of the systems. Data is an important component of any system. Here, data is pertaining to the details of members, books, accounts, and suppliers. Since people can interact with the system this system is an open system. The system is mainly concerned with the management of data it is an information system.

If this system were to be automated as conceived by the management, then role of the system analyst would be to study the system, its workings, and its existing problems. Also the analyst needs to provide a solution to the existing problem. Now that the management has decided for an automated system the analyst would perform the above tasks.

As the analyst did the study of the system, the following problems were identified

- Maintaining membership cards
- Producing reports due to large amount of data
- Maintaining accounts
- Keeping records for books in library and its members
- Performing searches

Now that the analyst has studied the system and identified the problems, it is the responsibility of the analyst to provide the best possible solution the management of the library.

Questions

Very Short Questions :

1. Who is system analyst ?
2. What are the stages in building an improved system ?
3. Define system ?

Short Questions :

1. What is system ?
2. What are the stages involved in system development ?
3. What are the skills required in a system analyst ?
4. Who are the users of system ?

Long Question :

1. What is system Analysis and design?
2. Discuss all types of users of the system?
3. What is the role of system Analyst?

❑❑❑

2

INTRODUCTION TO SYSTEM

Objective

To understand System Design concepts one has to first understand what exactly are Systems. In this session, we explore the meaning of system in accordance with analysts and designers. This session gives the reader basic concepts and terminology associated with the Systems. It also gives the overview of various types of systems.

In the broadest sense, a system is simply

a set of components that interact with each other to accomplish some purpose.

They are all around us. For example, human body is a biological system. We experience physical sensations by means of a complex nervous system, a set of parts, including brain, spinal cord, nerves, and special sensitive cells under our skin, that work together to make us feel hot, cold, itchy, and so on.

Language is another example of a system. Each language has got its set of alphabets, vocabulary and grammar rules. Combination of all these make possible for one person to convey thoughts to other persons.

An organization may also be viewed as a system where all the employees interact with each other and also with the employer to make the organization a functional unit. The organization also interacts with their customers to make a complete business system.

In today's world most of the people study. To make this possible, there are education systems. Each education system contains educational institutes like preparatory schools, middle and high schools and colleges. It also contains governing bodies, people (teachers and students) and some commercial bodies, which fulfill the other needs like stationery, transportation, furniture, etc.

In our day to day life, we see many business systems. These businesses have varied objectives, which range from producing a notebook to producing aircraft. These systems have their information needs. It can be for maintaining the records for employee for their wages calculations, keeping track of their leave status, maintaining company's expenses, inquiries from customers in case the business provide some service, or for keeping track for some particular function. So maintaining data is an important and essential activity in any business. The overall data maintained constitutes what is known as Information system.

Information system is the means by which data flow from one "person or department to another and can encompass everything from interoffice mail and elephone links to a computer system that generates periodic reports for various users. Information systems serve all the systems of a business, linking the different omponents in such a way that they effectively work towards the same purpose.

. Systems Concepts :

The term "System" is derived from the Greek word SYSTEMA.

Definations :

It means an organized relationship among functioning units or components.

We can define a System as a combination of resources or functional' working together to accomplish a given task.

The term "working together" in system definition is very important as all the components are interrelated and interdependent and can not exist independently. As definition says, these components interact with each other to accomplish a given task, which is actually the objective of the system. The components that comprise a stem may be the various inputs required by the system, the outcomes or the outputs of the system, the resources required to make the system functional etc.

1.1. System components :

A big system may be seen as a set of interacting smaller systems known as ubsystems or functional units each of which has its defined tasks. All these work in rdination to achieve the overall objective of the system.

As discussed above, a system is a set of components working together to achieve some goal. The basic elements of the system may be listed as:

- Resources
- Procedures
- Data/Information
- Processes

Resources : Every system requires certain resources for the system to exist. Resources can be hardware, software or liveware. Hardware resources may include the computer, its peripherals, stationery etc. Software resources would include the programs running on these computers and the livewere would include the human beings required to operate the system and make it functional. Thus these resources make an important component of any system. For instance, a Banking system cannot function without the required stationery like cheque books, pass books etc. such systems also need computers to maintain their data and trained staff to operate these computers and cater to the customer requirements.

Procedures : Ever system functions under a set of rules that 'govern the system to accomplish the defined goal of the system. This set of rules defines the procedures for the system to operate. For instance, the Banking systems have their predefined rules for providing interest at different rates for different types of accounts.

Data/Information: Every system has some predefined goal. For achieving the goal the system requires certain inputs, which are converted into the required output. The main objective of the System is to produce some useful output. Output is the outcome of processing. Output can be of any nature e.g. goods, services or information. However, the Output must conform to the customer's expectations. Inputs are the elements that enter the system and produce Output. Input can be of various kinds, like material, information, etc.

Intermediate Data : Various processes process system's Inputs. Before it is transformed into Output, it goes through many intermediary transformations. Therefore, it is very important to identify the Intermediate Data. For example, in a college when students register for a new semester, the initial form submitted by student goes through many departments. Each department adds their validity checks on it. Finally the form gets transformed and the student gets a slip that states whether the student has been registered for the requested subjects or not. It helps in building the System in a better way.

Intermediate forms of data occur when there is a lot of processing on the input data. So, intermediate data should be handled as carefully as other data since the output depends upon it.

Process : The systems have some processes that make use of the resources to achieve the set goal under the defined procedures. These processes are the operational element of the system. For instance in a Banking System there are several processes that are carried out. Consider for example the processing of a cheque as a process. A cheque passes through several stages before it actually gets processed and converted. These are some of the processes of the Banking system.

1.2 System Characterstics

All the components together make a complete functional system. Systems also exhibit certain features and characteristics, some of which are:

- Objective
- Standards
- Environment
- Feedback
- Boundaries and interfaces

Objective : Every system has a predefined goal or objective towards which it works. A system cannot exist without a defined objective. For example an organization would have an objective of earning maximum possible revenues, for which each department and each individual has to work in coordination.

Standards : It is the acceptable level of performance for any system. Systems should be designed to meet standards. Standards can be business specific or organization specific. For example take a sorting problem. There are various sorting algorithms. But each has its own complexity. So such algorithm should be used that gives most optimum efficiency. So there should be a standard or rule to use a particular algorithm. It should be seen whether that algorithm is implemented in the system.

Environment : Every system whether it is natural or man made co-exists with an environment. it is very important for a system to adapt itself to its environment. Also, for a system to exist it should change according to the changing environment. For example, we humans live in a particular environment. As we move to other places, there are changes in the surroundings but our body gradually adapts to the new environment. If it were not the case, then it would have been very difficult for human to survive for so many thousand years.

Another example can be Y2K problem for computer systems. Those systems, which are not Y2K compliant, will not be able to work properly after year 2000. For computer systems to survive it is important these systems are made Y2K compliant or Y2K ready.

Another example as the developmant in the technology field our banking experience has changed. Now we do not require to walk in bank for every small transaction. With the help of changes mode in working environment many transactions of banking system can be don using smart phone and internet.

Thus working enviorunment of system is very important characterstic.

Feedback : Feedback is an important element of systems. The output of a system needs to be observed and feedback from the output taken so as to improve the system and make it achieve the laid standards. In Fig. 1, it is shown that a system takes input. It then transforms it into output. Also some feedback can come from customer (regarding quality) or it can be some intermediate data (the output of one process and input for the other) that is required to produce final output.

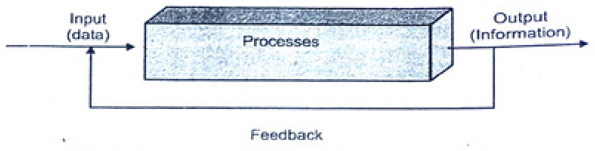


Fig. 1 : A system converts input data into output information

Boundaries and Interfaces: Every system has defined boundaries within which it operates. Beyond these limits the system has to interact with the other systems. For instance, Personnel system in an organization has its work domain with defined procedures. If the financial details of an employee are required, the system has to interact with the Accounting system to get the required details.

Interfaces are to another important element which the system interacts with the outside world. System interacts with other systems thorough its interfaces. Users of the systems also interact with it through interfaces. Therefore, these should be customized to the user needs. There should be as user friendly as possible.

1.3 Methodologies, Models, Tools and Techniques of System Development

- **A system development methodology** -provides guidelines to follow for completing every activity in the systems development life cycle.
- **A model** is a representation of some important aspect of the real world some models used in system development

  --Flowchart, DFD, ERD, Structure chart, Use case diagrams, class diagrams, seq. diagrams some models used to manage the development process

  PERT charts, Gantt chart organ. Hierarchy chart, Financial analysis models -NPV, ROI

Tools

- **A tool** is software support that helps create models or other components required in the project.

  smart editors, C-S help, debugging tools, CASE (Computer-Aided System Engineering)

  Tools- Help the analyst create the important system models.

Techniques

- A technique is a collection of guidelines that help an analyst complete a system development activity.

Some Techniques

OO analysis, Data Modeling, Relational database design, Structured analysis and design Software-testing

1.4 Approaches to System Development

System Design Concept has two closely related yet independent concepts approaches and methodologies. The first important concept is that there are two types of Systems Development Life Cycle approaches **Predictive approach and Adaptive approach.**

The second important concept is that there are two types of development methodologies **Structured approach and Object Oriented approach.**

These are two separate concepts. Projects can be any mix of these two approaches, the approach to the life cycle and the approach to the methodology predictive with structured, predictive with object oriented, adaptive with structured, or adaptive with object oriented.

We first presents and explains the differences in the life cycle approach the predictive and the adaptive approaches. These two approaches are really a continuum and any give project may have elements of both approaches. The predictive approach to the SDLC is used for projects that are well understood and low risk. The adaptive approach to the SDLC is used for projects that are not well understood and are higher risk. Adaptive SDLCs are more iterative and allow the project team to adapt the project to changing circumstances.

The object oriented approach refers to system development using newer object technologies that require a different approach to analysis, design, and programming.

One of the key concepts in system development is the systems development life cycle (SDLC). The SDLC refers to the entire process of building, deploying, using, and updating an information system. A predictive approach to the SDLC assumes that the development project is planned in advance and that the new information system can be developed according to the plan. An adaptive approach to the SDLC is used when the exact requirements or needs of the users are not well understood. A more flexible approach is needed that allows the plan to be modified as the project progresses.

We use the following techniques for software development.

(i) The Structured Approach

Structured System Development: The three techniques that make up the structured approach are Structured analysis, structured design, and structured programming. Sometimes, these techniques are collectively called as the Structured Analysis and Design Technique (SADT). Structured programming produces a program that has one beginning and one ending, with each step in the program execution consisting of one of three programming constructs

A sequence of program statements

A decision point at which one set or another set of statements executes

A repetition of a set of statements

Top down programming divides more complex programs into a hierarchy of program modules One module at the top of the hierarchy controls program execution by "calling" lower level modules as required. The modules and the arrangement of modules are shown graphically by using a model called a structure chart.

There are two main principles of structured design : Program modules should be (1) loosely coupled and (2) highly cohesive. Loosely coupled means that each module is as independent of the other modules. Highly cohesive means that each module accomplishes one clear task.

The structured analysis technique helps the developer define what the system needs to do (the processing requirements), what data the system needs to store and use (data requirements), what inputs and outputs are needed, and how the functions work together to accomplish tasks. The key graphical model of the system requirements that are used with structured analysis is called the data flow diagram (DFD). A model of the needed data is also created based on the types of things the system needs to store information (data entities) about. The data modeling is done with the entity relationship diagram (ERD).

(ii) The Object Oriented Approach

The object oriented approach views an information system as a collection of interacting objects that work together to accomplish tasks. An object is a thing in the computer system that is capable of responding to messages.

Given that the object oriented approach views information systems as collections of interacting objects, object oriented analysis (OOA) defines the objects that do the work and determines what user interactions (called use cases) are required to complete the tasks. Object oriented design (OOD) defines all the additional types of objects that are necessary to communicate with people and devices in the system, it shows how the objects interact to complete tasks. Object oriented programming (OOP) is the writing of statements in a programming language to define what each type of object does. This is done using "methods," which are smaller groups of program code. Within a given method programmers normally use structured programming constructs, of sequence, decision, and repetition.

An object is a type of thing. It could be a customer or an employee or it could be a button or a menu. Identifying types of objects means classifying things. A classification or "class" represents a collection (in reality it is a set) of similar objects; therefore, object oriented development uses a UML class diagram to show all the classes of objects that are in the system.

The object oriented approach yields several key benefits, among them naturalness and reusability. The approach is natural or intuitive for people because we tend to think about the world in terms of tangible objects. Because the object oriented approach involves classes of objects and many systems in the organization use the same objects, these classes can be used over and over again whenever they are needed.

2. Classification of Systems :

From previous section we have a firm knowledge of various system components and its characteristics. There are various types of system available which, can be categorized in many ways. Some of the categories are open or closed, physical or abstract and natural or man made information systems, which are explained below:

Classification of systems can be done in many ways.

2.1 Physical or Abstract Systems : Physical systems are tangible entities that we can feel and touch. These may be static or dynamic in nature. For example take a computer center. Desks and chairs are the static parts, which assist in the working of the center. Static parts don't change. The dynamic systems are constantly changing. Computer systems are dynamic system. Programs, data, and applications can change according to the user's needs.

Abstract systems are conceptual. These are not physical entities. They may be formulas, representation or model of a real system.

2.2 Open and Closed Systems : Systems interact with their environment to achieve their targets. Things that are not part of the system are environmental elements for the system. Depending upon the interaction with the environment, systems can be divided into two categories, open and closed.

Open Systems : System that interact with their environment. Practically most of the systems are open systems. An open system has many interfaces with its environment. It can also adapt to changing environmental conditions. It can receive inputs from, and delivers output to the outside of system. An information system is an example of this category.

Closed Systems : Systems that don't interact with their environment. Closed systems exist in concept only.

2.3 Man made Information Systems : The main purpose of information systems is to manage data for a particular organization. Maintaining files, producing information and reports are few functions.

An information system produces customized information depending upon the needs of the organization. These are usually formal, informal and computer based.

Formal information Systems: It deals with the flow of information from top. Management to lower management. Information flows in the form of memos, instructions, etc. But feedback can be given from lower authorities to top management.

Informal Information system: Informal systems are employee based. These are made to solve the day to day work related problems.

Computer-Based Information Systems: This class of systems depends on the use of computer for managing business applications. These systems are discussed in detail in the next section.

3. Information Systems :

In the previous section we studied about various classification of systems. Since in business we mainly deal with information systems we'll further explore these systems. We will be talking about different types of information systems prevalent in the industry.

Business Information system deals with data of the organizations. The purposes to business information system are to process input, maintain data produce reports, handle queries, handle on line transactions, generate reports, and other output. These maintain huge database, handle hundreds of queries etc. The transformation of data into information is primary function of information system.

These types of systems depend upon computers for performing their objectives. A computer based business system involves six interdependent elements These are (1) Hardware (machines) (2) Software (3) People (programmers, managers or users), (4) Procedures, (5) Data and (6) Information (processed data). All six elements interact to convert data into information. System analysis relies heavily upon computers to solve problems. For these types of systems, analyst should have a sound understanding of computer technologies.

In the following section, we explore three most important information systems namely, transaction processing system, management information system and decision support system, and examine how computers assist in maintaining Information systems.

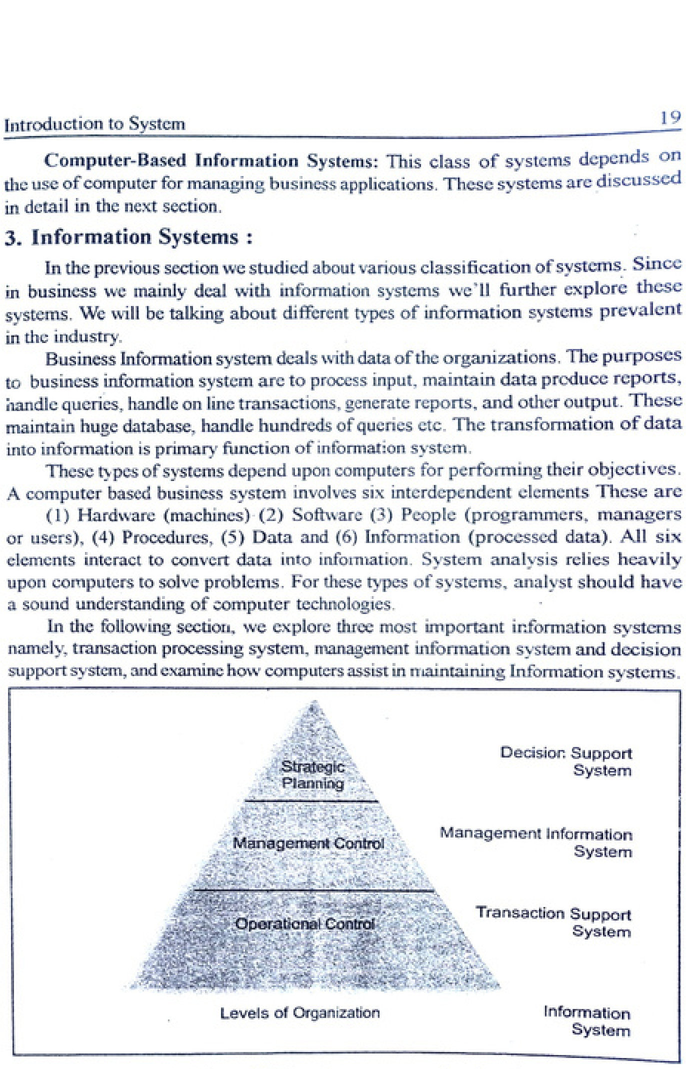


Fig. 2 : Relation of Information systems to levels of organization

3.1. Types of Information Systems :

Information systems differ in their business needs. Also depending upon different levels in organization information systems differ. Three major information systems are :

1. Transcation processing
2. Management information system
3. Decision support system

Figure 2 shows relation of information system to the levels of organization. The information needs are different at different organizational levels. Accordingly the information can be categorized as: strategic information, managerial information and operational information.

Strategic information is the information needed by top most management for decision making. For example the trends in revenues earned by the organization are required by the top management for setting the policies of the organization. This information is not required by the lower levels in the organization. The information systems that provide these kinds of information are known as Decision Support Systems.

The second category of information required by the middle management is known as managerial information. The information required at this level is used for making short term decisions and plans for the organization. Information like sales analysis for the past quarter or yearly production details etc. fall under this category. Management information system (MIS) caters to such information needs of the organization. Due to its capabilities to fulfill the managerial information needs of the organization, Management Information Systems have become a necessity for all big organizations. And due to its vastness, most of the big organizations have separate MIS departments to look into the related issues and proper functioning of the system.

The third category of information is relating to the daily or short term information needs of the organization such as attendance records of he employees. This kind of information is required at the operational level for carrying out the day-to-day operational activities. Due to its capabilities to provide information or processing transaction of the organization, the information system is known as Transaction Processing System or Data Processing System. Some examples of information provide by suc systems are processing of orders, posting of entries in bank, evaluating overdue purchaser orders etc.

1. Transaction Processing Systems : TPS processes business transaction of the organization. Transaction can be any activity of the organization. Transactions differ from organization to organization.

For example, take a railway reservation system. Booking, canceling, etc are all transactions. Any query made to it is a transaction. However, there are some

transactions, which are common to almost all organizations. Like employee new employee, maintaining their leave status, maintaining employees accounts, etc.

This provides high speed and accurate processing of record keeping of basic operational processes. These include calculation, storage and retrieval.

Transaction processing systems provide speed and accuracy, and can be programmed to follow routines functions of the organization.

2. Management Information Systems : These systems assist lower management in problem solving and making decisions. They use the results of transaction processing and some other information also. It is a set of information processing functions. It should handle queries as quickly as they arrive. An important element of MIS system is database. A database is a non-redundant collection of interrelated data items that can be processed through application programs and available to many users.

3. Decision Support Systems : These systems assist higher management to make long term decisions. These type of systems handle unstructured or semi structured decisions. A decision is considered unstructured It there are no clear procedures for making tile decision and if not all the factors to be considered in the decision can be readily identified in advance. These are not of recurring nature. Some recur infrequently or occur only once.

A decision support system must very flexible. The user should be able to produce customized reports by giving particular data and format specific to particular situations.

Summary of Information Systems	
Categories of Information System	**Characteristics**
Transaction Processing Systems	Substitutes computer-based processing for material procedures. Deals with well structured processes. Includes record keeping applications.
Management Information system	Provides input to be used in the managerial decision process. Deals with supporting well structured decision situations. Typical information requirements can be anticipated.
Decisions Support Systems	Provides information to managers who must make judgments about particular situations.Supports decision-makers in situations that are not well structured.

Table 1 : Characteristics of Information Systems

Approach	What?	When?	Why?	Why not?
SDLC	Building the system by completing 6 stages sequentially : 1. Project Definition 2. Systems Study 3. Design 4. Programming 5. Installation 6. Post-implementation	Medium to large mainframe based systems	1. Structured 2. Formal	1. Time consuming 2. Costly 3. Inflexible
Prototyping	Building and experimental and system equickly cheaply	Unclear user requirements involvement	1. User involvement quality 2. Fast	1. Poor system quality 2. Lack of standard
Package	Purchasing programs that have been written and tested	Common solution	1. Limited technical 2. Cost saving 3. Clear expectations	1. Not meeting all skills 2. Customization
End-user Development	Bilding the system by end-users with little or no formal technical assistance	Personal & small	1. No. misunder standing 2. Fast	1. Limited scope 2. Loss of control
Outsourcing	Using an external vendor develop or operate an organization's ISs	Mission non critical applications	1. Reduce costs 2. Predictability	1. Risky 2. Loss of control

Table 2 : Information System development approaches

4. Classification of Business Information System

4.1. Introduction :

There are various points on the basis of which Computer Based Business Information Systems may be classified. Some of the important areas which can be used to classify business information systems are discussed in this section.

4.2 Functional Areas :

Every organization has various functional areas which may be different for different organization.

The typical functional areas for a manufacturing company are listed Fig. 3.

Service to each of the functional areas will be provided of by a business information system. A business information system or its subsystem serving a particular functional area gets named or classified accordingly.

For instance there are business information systems which may be termed as,

- Financial Accounting Systems,
- Inventory Management Systems,
- Production and Control Systems etc.,
- Personnel Management System,
- Planning Management System

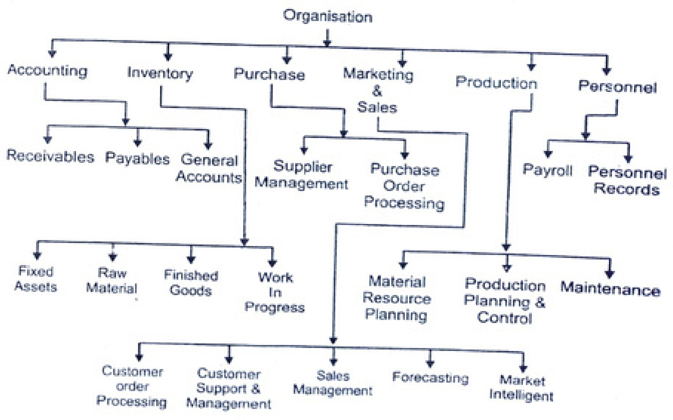


Fig. 3 : Functional Areas of Manufacturing Company

4.3. Usage Level :

Business information system may be viewed as having the following three layers.

- Transaction Processing System.
- Management Information System.
- Decision Support System.

4.4. Nature of Processing :

Depending on its processing modes, a business information system may be classified as a batch, online or realtime system. This classification is based on when and how fast the transactions are processed

(a) Batch Systems : In a batch system all the transactions for a period of given time are first collected as a batch. This batch of transactions is then processed at the end of the period.

- Payroll system is one of the good example of a batch system.
- The transactions of all the employees related to payroll like leaves, advances, increments, bonus, overtime, incentives etc. are collected for a month.
- At the end of the month these transactions are processed to produce the salary statement.
- This salary statement will gives the details of the salary to be paid to the employees of the organization for the previous month.

Generally Batch processing software is operated by the persons in the EDP staff and not used by the end user directly. Batch processing software need not be user friendly or interactive. Batch processing systems cannot provide latest information.

(b) Online Systems : In an online system all the transactions are captured at the origin and processed immediately.

Hotel Information System is typical example of online system where following actions are taken care off :

- Reservations
- Check Ins
- Check Outs

End users generally interacts with online software which should be interactive and user friendly. Online input devices like monitor are required for capturing data. Direct access files are required for mentaining database. Most online systems are being developed using RDBMSs The application programs are being written in either 3GL or 4GL Since transactions are processed immediately online systems can provide latest information A typical application may have both online and batch processing components.

(c) Realtime Systems : In this system the transactions are captured at the point of origin and are processed within a limited time period. A real time system may also be defined as one which controls the environment by receiving data processing them and returning the results sufficiently quickly to affect the environment at that time The term environment generally refers to a piece of equipment being controlled by the real time system.

A few examples of real time systems are listed in the table below.

System	Description
Process Control Systems	The computer systems that are used to control crude oil refineries, chemical manufacturing processes, mining and machining operations.
Automated Teller Systems	The ATMs that are used for making limited deposits and withdrawals at a bank ATM.
High Speed Data Acquisition Systems	Computers which receive data from satellites, laboratory equipment.
Computer Guided Missile System	Computer Systems that must track the trajectory of a missile and make continuous adjustments to its trajectory.

The term sufficiently quickly in the definition of a realtime system is subjective. There are many online systems- banking systems, airline reservation systems, stock brokerage systems that are expected to react very quickely.

However in most realtime systems the computer must react within nano seconds to the inputs that it receives. Another distinction between online and realtime systems is that, whereas online systems interact with people, the realtime systems interact with both people and environment.

A realtime system may receive inputs from the environment and may send results directly to the environment. The environment is controlled by the realtime system If the response is not fast enough the environment may go out of control.

5. Processing Location :

At one time, all data processing was centralized, with data recorded, input and processed at a central computer site. The cost of placing computers closer to the end-user was prohibitive.

Today because the computers have become cheaper, most of big organizations have decentralized their processing workload to multiple computer nodes.

Distributed computing can offer several advantages including improved responsiveness, better end user control over data and reduction in costs.

Computers used in distributed processing may be connected in a network Different types of network topologis can be used like .. point to point, bus, star hierarchical, ring etc.

6. Degree of Integration :

The information system for an organization may have multiple components e.g. financial accounting, inventory management, purchase sales etc

If all these different components share a common database of organization then the business information system will be considered integrated Otherwise each component will be considered a stand alone component.

Questions

Very Short Questions :

1. What is system ?
2. Define Resources in a system.
3. Define Procedures in a system.?
4. Define Intermediate Data in a System.
5. Define Process in a System.
6. What are different approaches of system development ?
7. What is information system ?

Short Questions :

1. What are types of Information System ?
2. What are the components of system ?
3. What are the characteristics of system ?
4. Classsify various types of system ?

Long Question :

1. Define the term system?
2. What are the various elements of system?
3. What is system analysis & design?
4. What are the role of system analyst?
5. Differentiate between
   (a) Open and closed system
   (b) Physical and abstract
6. What is business information system? Classify BIS with suitable Example.
7. What is degree of integration? Give comparison study between batch a relation system.

❑❑❑

3 SYSTEM DEVELOPMENT LIFE CYCLE

Objectives

1. Introduction to Life Cycle Models :

In the last session we studied about system's concepts and different types of systems. Also, a brief introduction to system analysis and design process was presented. Let us now look into various kinds of system development methodologies.

Nearly four decades ago the operations in an organization used to be limited and so it was possible to maintain them using manual procedures. But with the growing operations of organizations and use of Information Technology, the need to automate the various activities increased, since for manual procedures it was becoming very difficult and cumbersome. Like maintaining records for a thousand plus employees company on papers is definitely a cumbersome job. So, at that time more and more companies started going for automation.

Since there were a lot of organizations, which were opting for automation, it was felt that some standard and structural procedure or methodology be introduced in the industry so that the transition from manual to automated system became easy. The concept of system life cycle came into existence then. Life cycle model emphasized on the need to follow some structured approach towards building new or improved system. Many models were suggested for life cycle : waterfall model was among the very first models that came into existence. Later on many other models like prototype, rapid application development model, etc were also introduced.

System development begins with the recognition of user needs. Then there is-a preliminary investigation stage. It includes evaluation of present system, information gathering, feasibility study, and request approval. Feasibility study includes technical, economic, legal and operational feasibility. In economic feasibility cost-benefit analysis is done. After that, there are detailed design, implementation, testing and maintenance stages.

In this session, we will learn about various stages that exists in system's life cycle. In addition, different life cycles models will be discussed. These include Waterfall model, Prototype model, Object-Oriented Model, and Dynamic Systems Development Method (DSDM).

Activities involved in any Life cycle Model : Following activities are the part of any life cycle model. The sequence may be not being exactly same. However, each of these is necessarily covered in the life cycle. The following section describes each of these stages.

1.1. Preliminary Investigation :

Fig. 1 shows different stages in the system's life cycle. It initiates with a project request. First stage is the preliminary analysis. The main aim of preliminary analysis is to identify the problem. First, need for the new or the enhanced system is established. Only after the recognition of need, for the proposed system is done then further processing is possible.

Suppose in an office all leave-applications are processed manually. Now this company is recruiting many new people every year. So the number of employee in the company has increased. Thus the manual processing of leave application is becoming very difficult. Therefore the management is considering the option of automating the leave processing system. If this is the case, then the system analyst would need to investigate the existing system, find the limitations present, and finally evaluate whether automating the system would help the organization.

Once the initial investigation is done and the need for new or improved system is established, all possible alternate solutions are chalked out. All these systems are

known as "candidate systems". All the candidate systems are then weighed and the best alternative of all these is selected as the solution system, which is termed as the "proposed system".

1.2. Feasibility Study :

The proposed system is evaluated for its feasibility. Feasibility for a system means is it is practical and beneficial to build that system. Feasibility is evaluated from developer and customer's point of view. Developer sees whether they have the required technology or manpower to build the new system. Is building the new system really going to benefit the customer. Does the customer have the required money to build that type of a system? All these issues are covered in the feasibility study of the system. The feasibility of the system is evaluated on the three main issues: technical, economical, and operational. Another issue in this regard is the legal feasibility of the project.

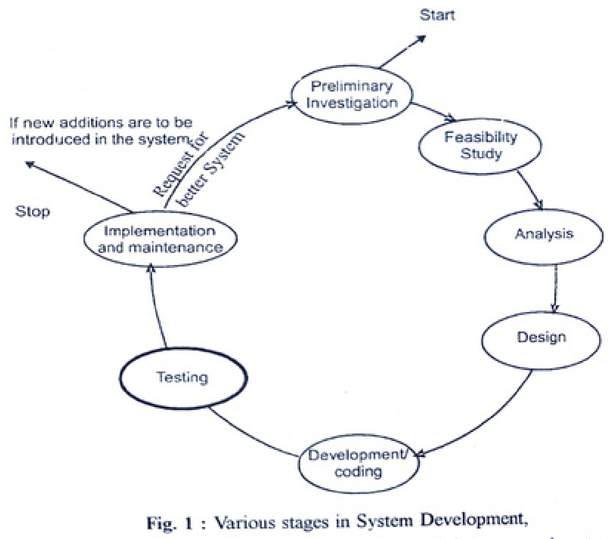

Start
Preliminary Investigation
Feasibility Study
Analysis
Design
Development/ coding
Testing
Implementation and maintenance
If new additions are to be introduced in the system
Request for better System
Stop

Fig. 1 : Various stages in System Development,

1. Technical feasibility: Can the development of the proposed system be done with current equipment, existing software technology, and available personnel? Does it require new technology?

2. Economic feasibility: Are there sufficient benefits in creating the system to make the costs acceptable? An-important outcome of the economic feasibility study is the cost benefit analysis.

3. Legal feasibility: It checks if there are any legal hassle in developing the system.

4. Operational feasibility: Will the system be used if it is developed and implemented?

Will there he resistance from users that will undermine the possible application benefits?

The result of the feasibility study is a formal document, a report detailing the nature and scope of the proposed solution. It consists of the following:

- Statement of the problem
- Details of findings
- Findings and recommendations in concise form

Once the feasibility study is done then the project is approved or disapproved according to the results of the study. if the project seems feasible and desirable then the project is finally approved otherwise no further work is done on it.

1.3. Analysis phase, Determination of System's requirements:

After preliminary investigation, analysis phase begins. Analysis is a detailed study of the various operations performed by the system, relationships among the various sub-systems or functional units and finally the relationships outside the system. Major questions under consideration during analysis are:

1. What is being done?
2. How is it being done?
3. Does a problem exist?
   If a problem exists, how severe is it?
5. How frequently does it occur?
6. What is the underlining cause for the problem?

Study is conducted to find user's information requirements. Proper functioning of the current system is also studied. Many tools are used during analysis. Data flow diagrams, on-site observations, and questionnaires are some examples. Once the analysis is completed, the analyst has a firm knowledge of what is to be done.

1.4. Design of System :

After the system has been analyzed by the analyst, the design stage of system life cycle begins. In design phase, the structure or design for the proposed system is finalized. Structure of files, databases, input, output, processes, and screens (interfaces) are decided. After design is finalized, it is clearly documented in what is called Design Document. This design document contains various graphical representations of reports, user interaction screens, etc. This design document is referred by developers during development of system.

Correct designing of the system is very important. If we have a wrong design we won't be able to a get correct desired system. From various studies it is observed that nearly 50% errors are made during design phase in any software development while 33% errors are pertaining to program logic and only 17% are syntactical errors. Also cost of fixing errors is maximum for design phase. Cost of fixing errors for design phase nearly amounts to 80% while it is only 20% for logic and programming.

Cost of change increases when errors are discovered during later stages for system development. Mostly design errors are discovered in development, testing, or maintenance phase. So the cost of change due to changes in design which might happen due to some error or improper design is very high. Thus the design process should be handled very carefully.

1.5. Development of Software :

In this phase, the actual development of the system takes place. That is, design representations are translated into actual programs. Software developers may install (or modify and then install) purchased software or they may write new, custom-designed programs.

Programmers are also responsible for documenting the program, providing an explanation of how procedures are coded. Documentation is essential to test the program and for further maintenance, after the application has been installed. It is also helpful to user in knowing the system well.

1.6. System Testing :

After a system has been developed, it is very important to check if it fulfills the customer requirements and is working as per requirement of user. For this purpose, testing of the system has to be done. For testing the systems, various test cases are prepared. A test case is a certain made up situation on which system is exposed so as to find the behavior of system in that type of real situation. These test cases require data. The data can be also made up artificial data or the real data provided by the user.

There are various types of tests which are used to test the system. These include unit, integration, and acceptance testing.

The smallest unit of software design is module. In unit testing these modules are tested. Since the modules are very small even individual programmer can test them. Once the individual modules are tested, these are integrated to build the complete system. But testing individual module doesn't guarantee if the system will work properly when these units are integrated.

Acceptance testing ensures that the system meets all the requirements. If it fulfils all the needs required by the system, it is accepted by the customer and put into use.

1.7. Implementation and Maintenance System :

System implementation means putting up system on user's site. Like any system, there is an aging process. Therefore, the system requires periodic maintenance and updations as per the requirements. Maintenance can be for software or hardware. User priorities, changes in organizational requirements, or environmental factors call for system enhancements. This is very crucial for the system's life.

2. Different Life Cycle Models :

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.

We have studied the various stages that are involved in the development of systems. There are system development models, which follow these stages. There is sequential traditional model also called waterfall model. Along with this there are many other approaches to system development. There is iterative Prototype Model, Dynamic Systems Development Model (DSDM), Spiral Model, Incremental Model and Object-Oriented Model. In the following section, we'll be discussing waterfall model, prototype model, incementak model, spiral model, rapid application development (RAD) Model, DSDM and object-oriented model.

2.1. Traditional/Waterfall Software Development Model :

Framework Type: Linear

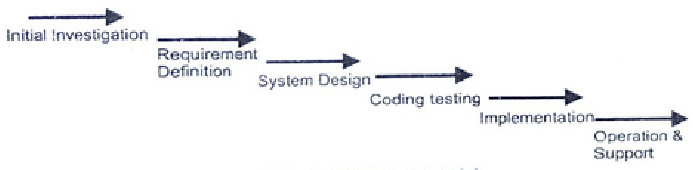


Fig. 2 : Waterfall Model

Basic Principles:

1. Project is divided into sequential phases, with some overlap and splash back acceptable between phases.

2. Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.

3. Tight control is maintained over the life of the project through the use of extensive

Written documentation, as well as through formal reviews and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase.

Strengths:

1. Ideal for supporting less experienced project teams and project managers, or project teams whose composition fluctuates.

2. The orderly sequence of development steps and strict controls for ensuring the adequacy of documentation and design reviews helps ensure the quality, reliability, and maintainability of the developed software.

3. Progress of system development is measurable.

4. Conserves resources.

Weaknesses:

1. Inflexible, slow, costly and cumbersome due to significant structure and tight controls.

2. Project progresses forward, with only slight movement backward.

3. Little room for use of iteration, which can reduce manageability if used. Depends upon early identification and specification of requirements, yet users may not be able to clearly define what they need early in the project.

5. Requirements inconsistencies, missing system components, and unexpected development needs are often discovered during design and coding.

6. Problems are often not discovered until system testing.

7. System performance cannot be tested until the system is almost fully coded, and under capacity may be difficult to correct.

8. Difficult to respond to changes. Changes that occur later in the life cycle are more costly and are thus discouraged.

9. Produces excessive documentation and keeping it updated as the project progresses is time-consuming.

10. Written specifications are often difficult for users to read and thoroughly appreciate.

11.Promotes the gap between users and developers with clear division of responsibility.

Situations where most Appropriate:

1. Project is for development of a mainframe-based or transaction-oriented batch system.

2. Project is large, expensive, and complicated.

3. Project has clear objectives and solution.

4. Pressure does not exist for immediate implementation.

5. Project requirements can be stated unambiguously and comprehensively.

6. Project requirements are stable or unchanging during the system development life cycle.

7. User community is fully knowledgeable in the business and application.

8. Team members may be inexperienced.

9. Team composition is unstable and expected to fluctuate.

10. Project manager may not be fully experienced.

11. Resources need to be conserved.

12. Strict requirement exists for formal approvals at designated milestones.

Situations where Least Appropriate:

1. Large projects where the requirements are not well understood or are changing for any reasons such as external changes, changing expectations, budget changes or rapidly changing technology.

2. Web Information Systems (WIS) primarily due to the pressure of implementing a WIS project quickly; the continual evolution of the project requirements; the need for experienced, flexible team members drawn from multiple disciplines; and the inability to make assumptions regarding the users' knowledge level.

3. Real-time systems.

4. Event-driven systems.

5. Leading-edge applications.

CASE STUDY: Library Management System :

Referring to the case of 'Library management system' introduced in the first chapter, let us develop this system on Waterfall approach.

In Waterfall model, first stage is preliminary analysis, which deals with the study of the current system, finding problems and establishing whether the new system will benefit the organization. In this library presently all transactions are done manually. Each member is allowed a certain number of books to borrow. He/she-has cards to borrow books. Against each card a member can borrow one book from library. Whenever a member wishes to issue a book and there are spare cards, then the book is issued. Otherwise that request is not entertained. The numbers of members are increasing day by day. It is becoming a problem to manage the cards of members. So an automated system is required that can maintain the details about the books borrowed by different system. As we have studied the system, we now know about the existing problems. And we can positively establish the need for the automated record keeping system. After this preliminary investigation, second phase of Waterfall model, i.e., analysis phase begins.

In analysis, a detail study of system is performed. Each operation is looked into more details. From the preliminary analysis we know that the members are increasing and managing their cards is now difficult task for the library staff Let us find out why increasing members is a problem. There might be a case when a card gets misplaced either by library staff or by member itself. If this is indeed the case then a duplicate card is made. But a member can lie about it and can make a duplicate card even if the card is not lost. In that case that particular member is having more than maximum allowed cards. There is no means to identity such members. Presently there is a need to have a system that can record the details about the books issued to members. Card system needs to be discarded. To solve this situation our library needs a computer application that has a database that contains details corresponding to each book issued. It should also have facilities to check if the number of books issued to a particular member doesn't exceed the maxlinum books allowed to him/her. Now we know what type of system is required, we can move to the next stage of waterfall model that is design stage.

From analysis phase it is clear that we need a database to implement our new system. In design stage it is decided as to which type of database system will be used. Secondly we'll identify what data should this database store and in what format. After that various operations like issuing and returning books are finalized. Various checks like number of books issued not to exceed the maximum number are finalized. Various interfaces that are to be used for input are decided.

Once all these details are finalized, these are properly documented. These documents are used in building the system during the development stage, which follows the design stage. Using the design details the new system is built. Only things that are identified during the design stage are incorporated into the system. There is no deviation from the design specifications. Suppose in the design phase the database to be used in the system is decided to be Oracle RDBMS then during development of the system only Oracle database will be used. Similarly each design specification is taken care of and whole new system is build.

Now we have developed our new system. But before it is implemented at the user's site it needs to be tested for performance and accuracy. So the system is tested to determine if it is performing correctly according to the requirements identified during the analysis phase. Each function of the system is tested. For example, we have issue book and return book functions. In return function it is checked if it is incrementing the count of variable that signifies how many books that member can issue more. In the issue function, it should be tested that this variable is decreased by one unit. Similarly whole of the system is tested.

After testing the system, it is implemented at the desired site of user. Now we have tested our new system for the library, it is implemented at the user's site i.e., library in our example.

After implementation, the systems require time to time maintenance. Maintenance can be for the software and hardware. Suppose two years after the implementation of the system, there occurs an alarming increase in number of members of the library. A situation can arise when all the memory for keeping the details have been exhausted then there will be a need to increase the memory of the system. Similarly the speed of processing requests might slow down. Then there will be a need for a faster processor. All these issues are maintenance issues. So maintenance, though it is last stage of the system development, is equally important. In this section we explored how we can solve our case study problem 'Library management system' using traditional Waterfall model.

Using Waterfall model is not mandatory. It is possible to follow other approaches also to solve the same problem. It depends upon the developer, which approach to follow. In the next section, we illustrate case study using another approach i.e. Prototyping. This is an iterative approach towards system development.

4. Alternate Development Models :

In this section, we will discuss in brief, some of the other models prevalent in the industry for systems development. Prototype, Dynamic System Development and Object Oriented methods are discussed.

4.1. Prototyping :

Framework Type: Iterative

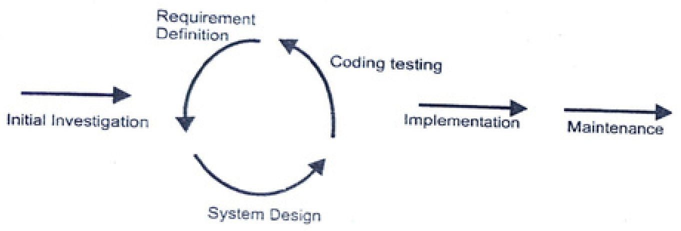


Fig. 3 : Prototyping

Basic Principles:

1. Not a standalone, complete development methodology, but rather an approach to handling selected portions of a larger, more traditional development methodology (i.e., Incremental, Spiral, or Rapid Application Development (RAD)).

2. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

3. User is involved throughout the process, which increases the likelihood of user acceptance of the final implementation.

4. Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users' requirements.

5. While most prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system.

6. A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problem.

Strengths:

1. "Addresses the inability of many users to specify their information needs, and the difficulty of systems analysts to understand the user's environment, by providing the user with a tentative system for experimental purposes at the earliest possible time." (Janson and Smith, 1985)

2. "Can be used to realistically model important aspects of a system during each phase of the traditional life cycle." (Huffaker, 1986)

3. Improves both user participation in system development and communication among project stakeholders.

4. Especially useful for resolving unclear objectives; developing and validating user requirements; experimenting with or comparing various design solutions; or investigating both performance and the human computer interface.

5. Potential exists for exploiting knowledge gained in an early iteration as later iterations are developed.

6. Helps to easily identify confusing or difficult functions and missing functionality.

7. May generate specifications for a production application.

8. Encourages innovation and flexible designs.

9. Provides quick implementation of an incomplete, but functional, application.

Weaknesses:

1. Approval process and control is not strict.

2. Incomplete or inadequate problem analysis may occur whereby only the most obvious and superficial needs will be addressed, resulting in current inefficient practices being easily built into the new system.

3. Requirements may frequently change significantly.

4. Identification of non-functional elements is difficult to document.

5. Designers may prototype too quickly, without sufficient up-front user needs analysis, resulting in an inflexible design with narrow focus that limits future system potential.

6. Designers may neglect documentation, resulting in insufficient justification for the final product and inadequate records for the future.

7. Can lead to poorly designed systems. Unskilled designers may substitute prototyping for sound design, which can lead to a "quick and dirty system" without global consideration of the integration of all other components. While initial software development is often built to be a "throwaway", attempting to retroactively produce a solid system design can sometimes be problematic.

8. Can lead to false expectations, where the customer mistakenly believes that the system is "finished" when in fact it is not; the system looks good and has adequate user interfaces, but is not truly functional.

9. Iterations add to project budgets and schedules, thus the added costs must be weighed against the potential benefits. Very small projects may not be able to justify the added time and money, while only the high-risk portions of very large, complex projects may gain benefit from prototyping.

10. Prototype may not have sufficient checks and balances incorporated.

Situations where most appropriate:

1. Project is for development of an online system requiring extensive user dialog, or for a less well-defined expert and decision support system.
2. Project is large with many users, interrelationships, and functions, where project risk relating to requirements definition needs to be reduced.
3. Project objectives are unclear.
4. Pressure exists for immediate implementation of something.
5. Functional requirements may change frequently and significantly.
6. User is not fully knowledgeable.
7. Team members are experienced (particularly if the prototype is not a throw-away).
8. Team composition is stable.
9. Project manager is experienced.
10. No need exists to absolutely minimize resource consumption.
11. No strict requirement exists for approvals at designated milestones.
12. Analysts/users appreciate the business problems involved, before they begin the project.
13. Innovative, flexible designs that will accommodate future changes are not critical.

Situations where least appropriate:

1. Mainframe-based or transaction-oriented batch systems.
2. Web-enabled e-business systems.
3. Project team composition is unstable.
4. Future scalability of design is critical.
5. Project objectives are very clear; project risk regarding requirements definition is low.

CASE STUDY: Library Management System :

Now we'll look at our case study 'Library management system' using Prototype model. Fig. 3 shows various stages involved in the Prototype model.

In our case the customer i.e., our library management are not technical people. All they know is that they need a system for their library that can automate the functions of the library. Only functions they have identified at this stage are issue and return of books. They want the new system as early as possible.

In this scenario, Prototyping approach can be used for system development.

First stage is requirement gathering. For our case the only requirements that the library staff has identified are having an automated system for books return and issue functions. So the analyst starts working on the presently identified requirements of system.

After requirements gathering, design stage begins. This is of very short period. Design is often very quick. Design considerations are same as discussed earlier in section 1.4. Database, procedures, checks, screen layouts, etc are decided. These are properly documented.

After design is finalized a working prototype or model is built. It is built to show the functionality of the system to the user before the development of the actual product. Once you have a prototype, it is shown to the customer. The customer verifies the prototype. In case there are some suggestions or alterations from customers' then again that functionality is added to the prototype and again it is evaluated by customer. This cycle gets repeated till the time the customer is fully satisfied with the prototype. Then the actual product is built. The same prototype can be used for the final product or a new product can be built based on the prototype.

In our case, first at the customer evaluation stage suppose our customer identifies that they require one more function that could handle fine amounts collected if the books return is delayed than due date. Then our designer will again go to the design stage and incorporate this function in the prototype. This cycle will be repeated till the customer is fully satisfied with the prototype.

The prototyping model of System Development is thus very different from the traditional Waterfall Approach. In this model, the phases of development are not sequential, but they are iterative, where any changes or additions suggested by the end user are incorporated by modifying the prototype model and again giving it to the user.

4.2. Incremental Model :

Framework Type: Combination Linear and Iterative

Basic Principles:

Various methods are acceptable for combining linear and iterative system development methodologies, with the primary objective of each being to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process:

1. A series of mini-Waterfalls are performed, where all phases of the Waterfall development model are completed for a small part of the system, before proceeding to the next increment; OR
2. Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of the system, OR
3. The initial software concept, requirements analysis, and design of architecture and system core are defined using the Waterfall approach, followed by iterative Prototyping, which culminates in installation of the final prototype (i.e., working system).

Strengths:

1. Potential exists for exploiting knowledge gained in an early increment as later increments are developed.

2. Moderate control is maintained over the life of the project through the use of written documentation and the formal review and approval/signoff by the user and information technology management at designated major milestones.

3. Stakeholders can be given concrete evidence of project status throughout the life cycle.

4. Helps to mitigate integration and architectural risks earlier in the project.

5. Allows delivery of a series of implementations that are gradually more complete and can go into production more quickly as incremental releases.

6. Gradual implementation provides the ability to monitor the effect of incremental changes, isolate issues and make adjustments before the organization is negatively impacted.

Weaknesses:

1. When utilizing a series of mini-Waterfalls for a small part of the system before moving on to the next increment, there is usually a lack of overall consideration of the business problem and technical requirements for the overall system.

2. Since some modules will be completed much earlier than others, well-defined interfaces are required.

3. Difficult problems tend to be pushed to the future to demonstrate early success to management.

Situations where most Appropriate:

1. Large projects where requirements are not well understood or are changing due to external changes, changing expectations, budget changes or rapidly changing technology.

2. Web Information Systems (WIS) and event-driven systems.

3. Leading-edge applications.

Situations where Least Appropriate:

1. Very small projects of very short duration.

2. Integration and architectural risks are very low.

3. Highly interactive applications where the data for the project already exists (completely or in part), and the project largely comprises analysis or reporting of the data.

4.3. Spiral Model :

Framework Type: Combination Linear and Iterative

Basic Principles:

1. Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle.

2. "Each cycle involves a progression through the same sequence of steps, for each portion of the product and for each of its levels of elaboration, from an overall concept-of operation document down to the coding of each individual program." (Boehm, 1986)

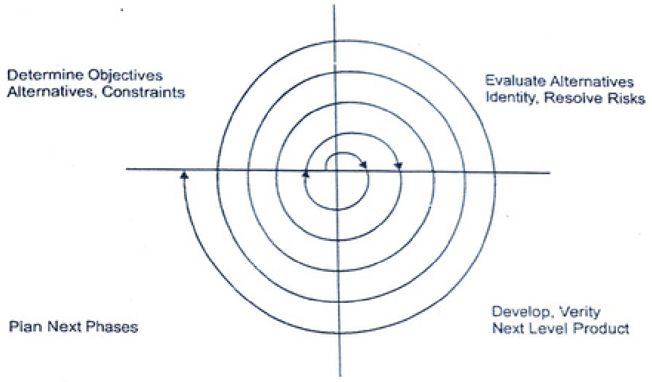


Fig. 4 : Spiral Model

3. Each trip around the spiral traverses four basic quadrants: (1) determine objectives, alternatives, and constraints of the iteration; (2) evaluate alternatives; identify and resolve risks; (3) develop and verify deliverables from the iteration; and (4) plan the next iteration. (Boehm, 1986 and 1988)

4. Begin each cycle with an identification of stakeholders and their win conditions, and end each cycle with review and commitment. (Boehm, 2000)

Strengths:

1. Enhances risk avoidance.

2. Useful in helping to select the best methodology to follow for development of a given software iteration, based on project risk.

3. Can incorporate Waterfall, Prototyping, and Incremental methodologies as special cases in the framework, and provide guidance as to which combination of these models best fits a given software iteration, based upon the type of project risk. For example, a project with low risk of not meeting user requirements, but high risk of missing budget or schedule targets would essentially follow a linear Waterfall approach for a given software iteration. Conversely, if the risk factors were reversed, the Spiral methodology could yield an iterative Prototyping approach.

Weaknesses:

1. Challenging to determine the exact composition of development methodologies to use for each iteration around the Spiral.

2. Highly customized to each project, and thus is quite complex, limiting reusability.

3. A skilled and experienced project manager is required to determine how to apply it to any given project.

4. There are no established controls for moving from one cycle to another cycle. Without controls, each cycle may generate more work for the next cycle.

5. There are no firm deadlines. Cycles continue with no clear termination condition, so there is an inherent risk of not meeting budget or schedule.

6. Possibility exists that project ends up implemented following a Waterfall framework.

Situations where most Appropriate:

1. Real-time or safety-critical systems.
2. Risk avoidance is a high priority.
3. Minimizing resource consumption is not an absolute priority.
4. Project manager is highly skilled and experienced.
5. Requirement exists for strong approval and documentation control.
6. Project might benefit from a mix of other development methodologies.
7. A high degree of accuracy is essential.
8. Implementation has priority over functionality, which can be added in later versions.

Situations where Least Appropriate:

1. Risk avoidance is a low priority.
2. A high degree of accuracy is not essential.
3. Functionality has priority over implementation.
4. Minimizing resource consumption is an absolute priority.

4.4. Rapid Application Development (RAD)

Framework Type: Iterative

Basic Principles:

1. Key objective is for fast development and delivery of a high quality system at a relatively low investment cost.

2. Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

3. Aims to produce high quality systems quickly, primarily through the use of iterative

4. Prototyping (at any stage of development), active user involvement, and computerized development tools. These tools may include Graphical User Interface (GUI) builders, Computer Aided Software Engineering (CASE) tools, Database Management Systems (DBMS), fourth-generation programming languages, code generators, and object-oriented techniques. Key emphasis is on fulfilling the business need, while technological or engineering excellence is of lesser importance.

5. Project control involves prioritizing development and defining delivery deadlines or "timeboxes". If the project starts to slip, emphasis is on reducing requirements to fit the timebox, not in increasing the deadline.

6. Generally includes Joint Application Development (JAD), where users are intensely involved in system design, either through consensus building in structured workshops, or through electronically facilitated interaction.

7. Active user involvement is imperative.

8. Iteratively produces production software, as opposed to a throwaway prototype.

9. Produces documentation necessary to facilitate future development and maintenance.

10. Standard systems analysis and design techniques can be fitted into this framework.

Strengths:

1. The operational version of an application is available much earlier than with Waterfall, Incremental, or Spiral frameworks.

2. Because RAD produces systems more quickly and to a business focus, this approach tends to produce systems at a lower cost.

3. Engenders a greater level of commitment from stakeholders, both business and technical, than Waterfall, Incremental, or Spiral frameworks. Users are seen as gaining more of a sense of ownership of a system, while developers are seen as gaining more satisfaction from producing successful systems quickly.

4. Concentrates on essential system elements from user viewpoint.

5. Provides the ability to rapidly change system design as demanded by users.

6. Produces a tighter fit between user requirements and system specifications.

7. Generally produces a dramatic savings in time, money, and human effort.

Weaknesses:

1. More speed and lower cost may lead to lower overall system quality.

2. Danger of misalignment of developed system with the business due to missing information.

3. Project may end up with more requirements than needed (gold-plating).

4. Potential for feature creep where more and more features are added to the system over the course of development.

5. Potential for inconsistent designs within and across systems.

6. Potential for violation of programming standards related to inconsistent naming conventions and inconsistent documentation.

7. Difficulty with module reuse for future systems.

8. Potential for designed system to lack scalability.

9. Potential for lack of attention to later system administration needs built into system.

10. High cost of commitment on the part of key user personnel.

11. Formal reviews and audits are more difficult to implement than for a complete system.

12. Tendency for difficult problems to be pushed to the future to demonstrate early success to management.

13. Since some modules will be completed much earlier than others, well-defined interfaces are required.

Situations where most Appropriate:

1. Project is of small-to-medium scale and of short duration (no more than 6 man-years of development effort).

2. Project scope is focused, such that the business objectives are well defined and narrow.

3. Application is highly interactive, has a clearly defined user group, and is not computationally complex.

4. Functionality of the system is clearly visible at the user interface.

5. Users possess detailed knowledge of the application area.

6. Senior management commitment exists to ensure end-user involvement.

7. Requirements of the system are unknown or uncertain.

8. It is not possible to define requirements accurately ahead of time because the situation is new or the system being employed is highly innovative.

9. Team members are skilled both socially and in terms of business.

10. Team composition is stable; continuity of core development team can be maintained.

11. Effective project control is definitely available.

12. Developers are skilled in the use of advanced tools.

13. Data for the project already exists (completely or in part), and the project largely comprises analysis or reporting of the data.

14. Technical architecture is clearly defined.

15. Key technical components are in place and tested.

16. Technical requirements (e.g., response times, throughput, database sizes, etc.) are reasonable and well within the capabilities of the technology being used. Targeted performance should be less than 70% of the published limits of the technology.

17. Development team is empowered to make design decisions on a day-to-day basis without the need for consultation with their superiors, and decisions can be made by a small number of people who are available and preferably co-located.

Situations where Least Appropriate:

1. Very large, infrastructure projects; particularly large, distributed information systems such as corporate-wide databases.

2. Real-time or safety-critical systems.

3. Computationally complex systems, where complex and voluminous data must be analyzed, designed, and created within the scope of the project.

4. Project scope is broad and the business objectives are obscure.

5. Applications in which the functional requirements have to be fully specified before any programs are written.

6. Many people must be involved in the decisions on the project, and the decision makers are not available on a timely basis or they are geographically dispersed.

7. The project team is large or there are multiple teams whose work needs to be coordinated.

8. When user resource and/or commitment is lacking.

9. There is no project champion at the required level to make things happen.

10. Many new technologies are to be introduced within the scope of the project, or the technical architecture is unclear and much of the technology will be used for the first time within the project

11. Technical requirements (e.g., response times, throughput, database sizes, etc.) are tight for the equipment that is to be used.

4.5. Object Oriented Methodology :

We live in a world of objects. These objects exist in nature, in man-made entities, in business, and in the products that we use. They can be categorized, described, organized, combined, manipulated and created. Therefore, an object-oriented view has come into picture for creation of computer software. An object-oriented approach to the development of software was proposed in late 1960s.

Object-Oriented development requires that object-oriented techniques to be used during the analysis and implementation of the system. This methodology asks the analyst to determine what are the objects of the system, how they will behav over time or in response to events, and what responsibilities and relationships ar object has to other objects. Object-oriented analysis has the analyst look at all th objects in a system, their commonalties, difference, and how the system needs t manipulate the objects.

Object Oriented Process : The Object Oriented Methodology of Building Systems takes the objects as the basis. For this, first the system to be developed is observed and analyzed and the requirements are defined as in any other method of system development. Once this is done, the objects in the required system are identified. For example in case of a Banking System, a customer is an object, a cheque book is an object: and even an account is an object.

In simple terms, Object Modeling is based on identifying the existing objects in a system and their interrelationships. After this the coding of the system is done. Object Modeling is somewhat similar to the traditional approach of system designing, in that it also follows a sequential process of system designing but with a different approach. The basic steps of system designing using Object Modeling may be listed as:

1. System Analysis
2. System Design
3. Object Design
4. Implementation

System Analysis : As in any other system development. model, system analysis is the first phase of development in case of Object Modeling too. In this phase, the developer interacts with the user of the system to find out the user requirements and analyses the system to understand the functioning. Based on this system study, the analyst prepares a model of the desired system. This model is purely based on what the system is required to do.' At this stage the implementation details are not taken care of Only the model of the system is prepared based on the idea that the system is made up of a set of interacting objects. The important elements of the system are emphasized.

System Design : System Design is the next development stage where the overall architecture of the desired system is decided. The system is organized as a set of subsystems interacting with each other. While designing the system as a set of interacting subsystems, the analyst takes care of specifications as observed in system analysis as well as what is required out of the new system by the end user. As the basic philosophy of Object-Oriented method of system design is to perceive the system as a set of interacting objects, a bigger system may also be seen as a set of interacting smaller subsystems that in turn are composed of a set of interacting objects. While designing the system, the stress lies on the objects comprising the system and not on the processes being carried out in the system as in the case of traditional Waterfall Model where the processes form the important part of the system.

Object Design : During this phase, the details of the system analysis and system design are implemented. The Objects identified in the system design phase are designed. Here the implementation of these objects is decided as the data structures get defined and also the interrelationships between the objects are defined.

Let us here deviate slightly from the design process and understand first a few impoltant terms used in the Object-Oriented Modeling.

As already discussed, Object Oriented Philosophy is very much similar to real world and hence is gaining popularity as the systems here are seen as a set of interacting objects as in the real world. To in1plement this concept, the process-based structural programming is not used; instead objects are created using data structures. Just as every programming language provides various data types and various variables of that type can be created, similarly, in case of objects certain data types are predefined. For example, we can define a data type called pen and then create and use several objects of this data type. This concept is known as creating a class.

Class : A class is a collection of similar objects. It is a template where certain basic characteristics of a set of objects are defined. The class defines the basic attributes and the operations of the objects of that type. Defining a class does not define any object, but it only creates a template. For objects to be actually created instances of the class are created as per the requirement of the case.

Abstraction : Classes are built on the basis of abstraction, where a set of similar objects are observed and their common characteristics are listed. Of all these, the characteristics of concern to the system under observation are picked up and tile class definition is made. The attributes of no concern to the s stem are left out. This is known as abstraction. The abstraction of an object varies according to its application. or instance, while defining a pen class for a stationery shop, the attributes of concern might be the pen color, ink color, pen type etc., whereas a pen class for a manufacturing firm would be containing the other dimensions of the pen like its diameter, its shape and size etc.

Inheritance : Inheritance is another important concept. A new type of class can be defined using a similar existing class with a few new features. For instance, a class vehicle can be defined with the basic functionality of any vehicle and a new class called car can be derived out of it with a few modifications. This would save the developers time and effort as the classes already existing are reused without much change.

Coming back to our development process, in the Object Designing phase of the Development process, the designer decides onto the classes in the system based on these concepts. The designer also decides on whether the classes need to be created from scratch or any existing classes can be used as it is or new classes can be inherited from them.

Implementation : In this phase, the class objects and the interrelationships of these classes are translated and actually coded using the programming language decided upon. The databases are made and the complete system is given a functional shape.

The complete object oriented methodology revolves around the objects identified in the system. When observed closely, every object exhibits some characteristics and behavior. The objects recognize and respond to certain events. For example, considering a Window on the screen as an object, the size of the window gets changed when resize button of the window is clicked. Here the clicking of the button is an event to which the window responds by changing its state from the old size to the new size. While developing systems based on this approach, the analyst makes use of certain models to analyze and depict these objects. The methodology supports and uses three basic Models:

Object Model - This model describes the objects in a system and their interrelationships. This model observes all the objects as static and does not pay any attention to their dynamic nature.

Dynamic Model - This model depicts the dynamic aspects of the system. It portrays the changes occurring in the states of various objects with the events that might occur in the system.

Functional Model - This model basically describes the data transformations of the system. This describes the flow of data and the changes that occur to the data throughout the system.

While the Object Model is most important of all as it describes the basic element of the system. the objects. all the three models together describe the complete functional system.

As compared to the conventional system development techniques, object oriented modeling provides many benefits. Among other benefits, there are all the benefits of using the Object Orientation. Some of these are:

Reusability - The classes once defined can easily be used by other applications. This is achieved by defining classes and putting them into a library of classes where all the classes are maintained for future use. Whenever a new class is needed the programmer looks into the library of classes and if it is available, it can be picked up directly from there.

Inheritance - The concept of inheritance helps the programmer use the existing mode in another way, where making small additions to the existing classes can quickly create new classes.

Programmer has to spend less time and effort and can concentrate on other aspects of the system due to the reusability feature of the methodology.

Data Hiding - Encapsulation is a technique that allows the programmer to hide the internal functioning of the objects from the users of the objects. Encapsulation separates the internal functioning of the object from the external functioning thus providing the user flexibility to change the external behavior of the object making the programmer code safe against the changes made by the user.

The systems designed using this approach are closer to the real world as the real world functioning of the system is directly mapped into the system designed using this approach.

Advantages:

OO Model closely represents the problem domain. Because of this, it is easier to produce and understand designs.

The objects in the system are immune to requirement changes. Therefore, allows changes more easily.

OO designs encourage more re-use. New applications can use the existing modules, thereby reduces the development cost and cycle time.

OO approach is more natural. It provides nice structures for thinking and abstracting and leads to modular design.

4.6. Dynamic System Development Method :

Dynamic System Development Method is another approach to system development, which develops the system dynamically. This methodology is independent of tools, in that it can be used with both structured analysis and approach or object-oriented approach.

The DSDM development process is dynamic as it is a Rapid Application Development method that uses incremental prototyping. This method is particularly useful for the systems to be developed in short time span and where the requirements cannot be frozen at the start of the application building. Whatever requirements are known at a time design for them is prepared and design is developed and incorporated into system. In DSDM, analysis, design and development phase can overlap. Like at one time some people will be working on some new requirements while some will be developing something for the system. In DSDM, requirements evolve with time.

DSDM has a five-phase as shown in Fig. 5

Feasibility Study : In this phase the problem is defined and the technical feasibility of the desired application is verified. Apart from these routine tasks, it is also checked whether the application is suitable for Rapid Application Development (RAD) approach or not. Only if the RAD is found as a justified approach for the desired system, the development continues.

Business Study : In this phase the overall business study of the desired system is done. The business requirements are specified at a high level and the information requirements out of the system are identified. Once this is done, the basic architectural framework of the desired system is prepared. The systems designed using RAD should be highly maintainable, as they are based on the incremental development process. The maintainability level of the system is also identified here so as to set the standards for quality control activities throughout the development process.

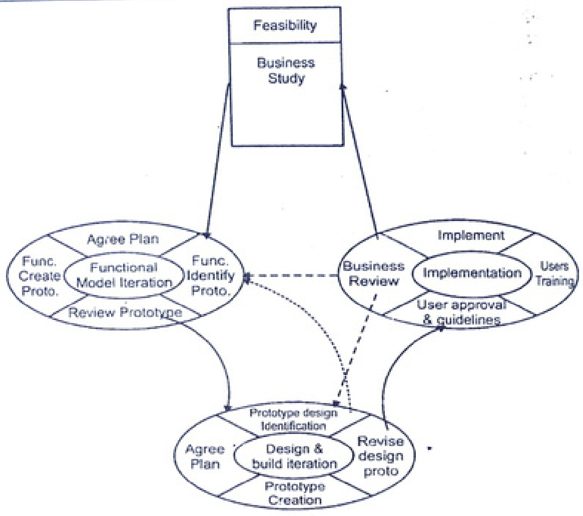


Fig. 5 : Dynamic Systems Development Model Source: DSDM consortium

Functional Model Iteration : This is one of the two iterative phases of the life cycle. The main focus in this phase is on building the prototype iteratively and getting it reviewed from the users to bring out the requirements of the desired system. The prototype is improved through demonstration to the user, taking the feedback and incorporating the changes. This cycle is repeated generally twice or thrice until a part of functional model is agreed upon. The end product of this phase is a functional model consisting of analysis model and some software components containing the major functionality.

Design and Build Iteration : This phase stresses upon ensuring that the prototypes are satisfactorily and properly engineered to suit their operational environment. The software components designed during the Functional modeling are further refined till they achieve a satisfactory standard. The product of this phase is a tested system ready for implementation.

There is no clear line between these two phases and there may be cases where while some component has flown from the functional modeling to the design and

build modeling while the other component has not yet been started. The two phases, as a result, may simultaneously continue.

Implementation : Implementation is the last and final development stage in this methodology. In this phase the users are trained and the system is actually put into. the operational environment. At the end of this phase, there are four possibilities, as depicted by figure:

- Everything was delivered as per the user demand, so no further development required.
- A new functional area was discovered, so return to business study phase and repeat the whole process
- A less essential part of the project was missed out due to time constraint and so development returns to the functional model iteration.
- Some non-functional requirement was not satisfied, so development returns to the design and build iterations phase.

DSDM assumes that all previous steps may be revisited as part of its iterative approach. Therefore, the current step need be completed only enough to move to the next step, since it can be finished in a later iteration. This premise is that the business requirements will probably change anyway as understanding increases, so any further work would have been wasted. According to this approach, the time is taken as a constraint i.e. the time is fixed, resources are fixed while the requirements are allowed to change. This does not follow the fundamental assumption of making a perfect system the first time, but provides a usable and useful 80% of the desired system in 20% of the total development time. This approach has proved to be very useful under time constraints and varying requirements.

Limitations:

- It is a relatively new model. It is not very common. So it is difficult to understand.

Advantages:

- Active user participation throughout the life of the project and development improves quality of the product.
- DSDM ensures rapid deliveries.
- Both of the above factors result in reduced project costs.

5. Software Development Methodologies

A framework that is used to structure, plan, and control the process of developing an information system is a software development methodology or system development methodology.

There are the following methodologies:

- Agile Software Development
- Crystal Methods
- Dynamic Systems Development Model (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD).
- Joint Application Development (JAD)
- Lean Development (LD)
- Rapid Application Development (RAD)
- Rational Unified Process (RUP)
- Scrum
- Spiral
- Systems Development Life Cycle (SDLC)
- Waterfall (Traditional / spiral)

5.1 Agile Software Development Methodology

Conceptual framework for undertaking software engineering projects is Agile software development. Most agile methods attempt to minimize risk by developing software in small time frames, called iterations, which typically last one to five weeks. Each iteration is like a miniature software project of its own, and includes all the tasks necessary to release the mini-increment of new functionality: planning, requirements analysis, design, coding, testing, and documentation. While iteration may not add enough functionality to warrant releasing the product, an agile software project intends to be capable of releasing new software at the end of every iteration. At the end of each iteration, the team re evaluates project priorities. Agile methods emphasize real time communication, preferably face-to-face, over written documents. Agile methods also emphasize working software as the primary measure of progress. With the focus on face-to-face communication, agile methods produce very less written documentation relative to other methods.

5.2 Crystal Methods Methodology

Crystal Methods approach was developed by Alistair Cockburn. His focus was on the people, interaction, community, skills, talents, and communications with

the belief that these are what have the first-order effect on performance. Process, he says, is important, but secondary. Cockburn's philosophy translate into a recognition that each team has a different set of talents and skills and therefore each team should use a process uniquely tailored to it. And it means that the process should be minimized - barely significant.

The use of the word "crystal" refers to the various facets of a gemstone - each a different face on an underlying core. The underlying core represents values and principles, while each facet represents a specific set of elements such as techniques, roles, tools, and standards.

Cockburn also differentiates between methodology, techniques, and policies. A methodology is a set of elements (practices, tools); techniques are skill areas such as developing use cases; and policies dictate organizational "musts".

5.3 Dynamic Systems Development Model Methodology (DSDM)

In U.K. during mid-1990s the Dynamic Systems Development Model was developed. It is the evolution of rapid application development (RAD) practices. DSDM boasts the best-supported training and documentation of any of the agile software development techniques, at least in Europe. DSDM favors the philosophy that nothing is built perfectly the first time and looks to software development as an exploratory endeavor.

The nine principles of DSDM are:

- Active user involvement.
- Empowered teams that the authority to can make decisions.
- A focus on frequent delivery of products.
- Using fitness for business purpose as the essential criterion for acceptance of deliverables.
- Iterative and incremental development to ensure convergence on an accurate business solution.
- Reversible changes during development.
- Requirements that is baselined at a high level.
- Integrated testing throughout the life cycle.
- Collaboration and cooperation between all stakeholders.

5.4 Extreme Programming (XP) Methodology

XP is a methodology for creating software within a very unstable environment. It allows flexibility within the modelling process. The main goal of XP is to lower the cost of change in software requirements. With traditional system development methodologies, like the Waterfall Methodology, the requirements for the system are determined and often freezed at the beginning of the development project. This means that the cost of changing the requirements at a later stage in the project - something that is very common in the real-world can be very high.

XP Core Practices

The core practices of Extreme Programming, as described in the first edition of "Extreme Programming Explained" can be grouped into four areas as follows:

Fine scale feedback Continuous process rather than batch Shared understanding Programmer welfare

The core practices are derived from generally accepted best practices, and are taken to extremes:

Interaction between developers and customers is good. Therefore, an XP team is supposed to have a customer on site, who specifies and prioritizes work for the team, and who can answer questions as soon as they arise. (In practice, this role is sometimes fulfilled by a customer proxy.)

If learning is good, take it to extremes: Reduce the length of development and feedback cycles. Test early.

Simple code is more likely to work. Therefore, extreme programmers only write code to meet actual needs at the present time in a project, and go to some lengths to reduce complexity and duplication in their code.

If simple code is good, re-write code when it becomes complex. **Code reviews are good.** Therefore XP programmers work in pairs, sharing one screen and keyboard (which also improves communication) so that all code is reviewed as it is written.

Testing code is good. Therefore, in XP, tests are written before the code is written. The code is considered complete when it passes the tests (but then it needs refactoring to remove complexity). The system is periodically, or immediately tested using all pre-existing automated tests to assure that it works. See test-driven development.

5.5 Feature Driven Development Methodology

In the development of Feature Driven Development methodology Jeff De Luca and Peter Coad were both greatly involved. Peter describes FDD as having just enough process to ensure scalability and repeatability while encouraging creativity and innovation.

More specifically, Feature Driven Development asserts that:

- A system for building systems is necessary in order to scale to larger projects.
- A simple, but well-define process will work best.
- Process steps should be logical and their worth immediately obvious to each team member.
- "Process pride" can keep the real work from happening.
- Good processes move to the background so team members can focus on results.
- Short, iterative, feature-driven life cycles are best.

FDD proceeds to address the items above with this simple process (numbers in brackets indicate the project time spent):

1. Develop an overall model (10 percent initial, 4 percent ongoing)
2. Build a features list (4 percent initial, 1 percent ongoing)
3. Plan by feature (2 percent initial, 2 percent ongoing)
4. Design by feature
5. Build by feature (77 percent for design and build combined)

5.6 Joint Application Development (JAD) Methodology

Chuck Morris and Tony Crawford, while working at IBM, developed the JAD methodology in the late 1970s and began teaching the approach in to the 1980s. Joint Application Development is a requirements definition and user-interface design methodology in which end-users, executives, and developers attend intense off-site meetings to work out a system's details. So the Joint Application Development (JAD) methodology aims to involve the client in the design and development of an application.

JAD rather than focusing on technical problems focuses on the business problem. It is most applicable to the development of business systems, but it can be used successfully for systems software. It produces its savings by shortening the elapsed time required to gather a system's requirements and by gathering requirements better, thus reducing the number of costly, downstream requirements changes. Its success depends on effective leadership of the JAD sessions; on participation by key end-users, executives, and developers; and on achieving group synergy during JAD sessions.

In contrast to the Waterfall approach, JAD is thought to lead to shorter development times and greater client satisfaction, both of which stem from the constant involvement of the client throughout the development process. On the other hand, with the traditional approach to systems development, the developer investigates the system requirements and develops an application, with client input consisting of a series of interviews.

Rapid application development (RAD), a variation on JAD, attempts to create an application more quickly through strategies that include fewer formal methodologies and reusing software components.

5.7 Lean Development (LD) Methodology

Bob Charette the founder of Lean Development focuses on the creation of change-tolerant software. This methodology embodies the notion of dynamic stability which can be thought of as similar to how Scrum embraces controlled chaos. It further states that measurable goal of LD is to build software with one-third the human effort, one-third the development hours and one-third the investment. There are 12 principles of Lean Development:

1. Satisfying the customer is the highest priority.
2. Always provide the best value for the money.
3. Success depends on active customer participation.
4. Every LD project is a team effort.
5. Everything is changeable.
6. Domain, not point, solutions.
7. Complete, don't construct.
8. An 80 percent solution today instead of 100 percent solution tomorrow.
9. Minimalism is essential.
10. Needs determine technology.
11. Product growth is feature growth, not size growth.
12. Never push LD beyond its limits.

5.8 Rapid Application Development (RAD) Methodology

"Rapid-development language" is a general term that refers to any programming language that offers speedier implementation than do traditional third-generation languages such as C/C++, Pascal, or Fortran. Rapid-Development Languages (RDLs) produce their savings by reducing the amount of construction needed to build a product. Although the savings are realized during construction, the ability to shorten the construction cycle has project wide implications: shorter construction cycles make incremental lifecycles such as Evolutionary Prototyping practical. Because RDLs often lack first-rate performance, constrain flexibility, and are limited to specific kinds of problems, they are usually better suited to the development of in-house business software and limited-distribution custom software than systems software.

RAD (rapid application development) proposes that products can be developed faster and of higher quality by:

- Using workshops or focus groups to gather requirements.
- Prototyping and user testing of designs.
- Re-using software components.
- Following a schedule that defers design improvements to the next product version.
- Keeping review meetings and other team communication informal.

There are commercial products that include requirements gathering tools, prototyping tools, software development environments such as those for the Java platform, groupware for communication among development members, and testing tools. RAD usually embraces object-oriented programming methodology, which inherently fosters software re-use. The most popular object-oriented programming languages, C++ and Java, are offered in visual programming packages often described as providing rapid application development.

5.9 Rational Unified Process (RUP) Methodology

The Rational Unified Process attempts to capture many of modern software development's best practices in a form suitable for a wide range of projects and organizations. This process recognizes that the traditional waterfall approach can be inefficient because it idles key team members for extended periods of time. Many feel that the waterfall approach also introduces a lot of risk because it defers testing and integration until the end of the project lifecycle. Problems found at this stage are very expense to fix.

By contrast, RUP represents an iterative approach that is superior for a number of reasons:

- It lets you take into account changing requirements which despite the best efforts of all project managers are still a reality on just about every project.
- Integration is not one "big bang" at the end; instead, elements are integrated progressively.
- Risks are usually discovered or addressed during integration. With the iterative approach, you can mitigate risks earlier.
- Iterative development provides management with a means of making tactical changes to the product. It allows you to release a product early with reduced functionality to counter a move by a competitor, or to adopt another vendor for a given technology.
- Iteration facilitates reuse; it is easier to identify common parts as they are partially designed or implemented than to recognize them during planning.
- When you can correct errors over several iterations, the result is a more robust architecture. Performance bottlenecks are discovered at a time when they can still be addressed, instead of creating panic on the eve of delivery.
- Developers can learn along the way, and their various abilities and specialties are more fully employed during the entire lifecycle. Testers start testing early, technical writers begin writing early, and so on.
- The development process itself can be improved and refined along the way. The assessment at the end of iteration not only looks at the status of the project from a product or schedule perspective, but also analyzes what should be changed in the organization and in the process to make it perform better in the next iteration.

5.10 Scrum Methodology

Scrum is an agile method for project management developed by Ken Schwaber. Its goal is to dramatically improve productivity in teams previously paralyzed by heavier, process-laden methodologies.

Scrum is characterized by:

- A living backlog of prioritized work to be done.
- Completion of a largely fixed set of backlog items in a series of short iterations or sprints.

- A brief daily meeting (called a scrum), at which progress is explained, upcoming work is described, and obstacles are raised.
- A brief planning session in which the backlog items for the sprint will be defined.
- A brief heartbeat retrospective, at which all team members reflect about the past sprint.

Scrum is facilitated by a scrum master, whose primary job is to remove impediments to the ability of the team to deliver the sprint goal. The scrum master is not the leader of the team (as they are self-organizing) but acts as a productivity buffer between the team and any destabilizing influences.

Scrum enables the creation of self-organizing teams by encouraging verbal communication across all team members and across all disciplines that are involved in the project. A key principle of scrum is its recognition that fundamentally empirical challenges cannot be addressed successfully in a traditional "process control" manner. As such, scrum adopts an empirical approach - accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to respond in an agile manner to emerging challenges.

5.11 Spiral Methodology

The Spiral Lifecycle Model is a sophisticated lifecycle model that focuses on early identification and reduction of project risks. A spiral project starts on a small scale, explores risks, makes a plan to handle the risks, and then decides whether to take the next step of the project - to do the next iteration of the spiral. It derives its rapid development benefit not from an increase in project speed, but from continuously reducing the projects risk level - which has an effect on the time required to deliver it. Success at using the Spiral Lifecycle Model depends on conscientious, attentive, and knowledgeable management .It can be used on most kinds of projects, and its risk-reduction focus is always beneficial.

The spiral methodology extends the waterfall model by introducing prototyping. It is generally chosen over the waterfall approach for large, expensive, and complicated projects.

At a high-level, the steps in the spiral model are as follows:

1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.

2. A preliminary design is created for the new system.

3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

4. A second prototype is evolved using four steps:

- Evaluate the first prototype and identify its strengths, weaknesses, and risks.

- Define the requirements of the second prototype.
- Plan and design the second prototype.
- Construct and test the second prototype.

5. At the project sponsor's option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could result in a less-than-satisfactory final product.

6. The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

7. The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

8. The final system is constructed, based on the refined prototype.

9. The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

5.12 Systems Development Life Cycle (SDLC) Methodology

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved, including the waterfall model (which was the original SDLC method); rapid application development (RAD); joint application development (JAD); the fountain model; the spiral model; build and fix; and synchronize-and-stabilize.

Often, several models are combined into some sort of hybrid methodology. Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with the development process. Some methods work better for specific types of projects, but in the final analysis, the most important factor for the success of a project may be how closely the particular plan was followed.

In general, an SDLC methodology follows these steps:

1. If there is an existing system, its deficiencies are identified. This is accomplished by interviewing users and consulting with support personnel.

2. The new system requirements are defined including addressing any deficiencies in the existing system with specific proposals for improvement.

3. The proposed system is designed. Plans are created detailing the hardware, operating systems, programming, and security issues.

4. The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use, and all aspects

of performance must be tested. If necessary, adjustments must be made at this stage.

5. The system is put into use. This can be done in various ways. The new system can phased in, according to application or location, and the old system gradually replaced. In some cases, it may be more cost-effective to shut down the old system and implement the new system all at once.

6. Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

5.13 Waterfall (Traditional / Linear) Methodology

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is rigid and linear. Waterfall development has distinct goals for each phase of development where each phase is completed for the next one is started and there is no turning back.

The perceived advantages of the waterfall process are that it allows for departmentalization and managerial control. A schedule is typically set with deadlines for each stage of development and a product can proceed through the development process. In theory, this process leads to the project being delivered on time because each phase has been planned in detail.

In practice, waterfall development often falls short of expectations as it does not embrace the inevitable changes and revisions that become necessary with most projects. Once an application is in the testing stage, it is very difficult to go back and change something that was not thought of in the concept stage.

Questions

Very Short Questions

1. What is SDLC ?
2. What is RAD ?
3. What is Waterfall Model ?
4. What is System Testing ?
5. What is Maintanence ?
6. What is Dynamic System analysis ?
7. What do you mean by System Design ?

8. What is spiral model ?
9. What do you mean by RUP ?
10. What is DSDM ?

Short Questions :

1. What is preliminary Investigation ?
2. What is feasibality study ?
3. Name different types of System Development Model ?
4. What are different software development methodologies ?

Long Question :

1. What are the various stages in waterfall, prototyping A DSDM ?
2. What are the advantage of prototype methodology over traditional waterfall model ?
3. What is SDLC, discuss it's all stages in brief ?
4. What is object Oriented methodology ?
5. What is Rapid Development Methodology ?
6. Describe various types of Feasibality study ?

❑❑❑

4

PRELIMINARY ANALYSIS & FEASIBILITY STUDY

Objectives

In the previous chapters we had studied about systems, their components, different types of systems and different system development models. From this session onwards we'll be looking into details of various activities involved in the development of a System Department Life Cycle (SDLC). First stage of any system development is Preliminary Analysis. Understanding the customer request for the new or improved system is the first activity performed in preliminary analysis. Secondly thinking of possible solutions. After that, each possible solution is looked more carefully and it is ascertained whether that system is practically possible or not. In the end of preliminary analysis all the findings and recommendations for which solution to use, are presented to management, which finally decide whether to accept the proposal or reject it.

1. Preliminary Analysis :

The main objectives of Preliminary Analysis is to identify the customer's needs, evaluate system concept for feasibility, perform economic and technical analysis, perform cost benefit analysis and create system definition that forms the foundation for all subsequent engineering works.

There should be enough expertise available for hardware and software for doing analysis.

While performing analysis, the following questions arise.

- How much time should be spent on it?

As such, there are no rules or formulas available to decide on this. However, size, complexity, application field, end-use, contractual obligation are few parameters on which it should be decided.

- Other major question that arises is who should do it.

Well an experienced well-trained analyst should do it. For large project, there can be an analysis team.

After the preliminary analysis, the analyst should report the findings to management, with recommendations outlining the acceptance or rejection of the proposal.

1.1. Request Clarification :

Initially when the customer makes a request for a system, the customer itself is not sure about the exact requirements of the system. In order to know what exactly the customer expects out of system, the analyst personally meets the end user or customer. For this analyst should be very tactful and have a good communications skills so that he/she is able to extract maximum information. In this way analyst is able to know about the exact requirements for the desired system. All information gathered during the identification period is recorded in System Concept Document.

2. Feasibility Study :

The process of evaluating the options, and arriving at the best solution is what doing a Feasibility Study is all about. Essentially it means that if it is proposed to build a new system, some preliminary investigation is done at this stage to find out whether this new system is actually feasible and viable to implement.

An analyst at the end of this initial study will prepare a feasibility report. Based on this the management will decide either to go ahead with the project or to drop it.

Why is Feasibility Study Required? : Basically a Feasibility Study is done to find out whether for the system that is proposed, will be:

- Possible (to build it with the given technology and resources)
- Affordable (given the time and cost constraints of the organization)
- Acceptable (for use by the eventual users of the system)

You might wonder then, whether a feasibility study is mandatory for every project? In certain smaller projects where it is very clear as to what needs to be done, the benefits of which are very obvious and where there are minimal risks involved, a formal study of this kind may not be warranted. In larger projects where the variables and imponderables are more and the implications vary depending on the options chosen, the feasibility study is certainly a timely investment to ensure that there are no unforeseen surprises or shocks much later in the life of a project.

At what Point in Time should a Feasibility Study be done? : The feasibility study attempts to find out the cost of developing and implementing a new system and the benefits that are likely to accrue out of the same. In doing so it might well turn out that automation may not be the solution at all. So, basically the study tries to find out whether it is worth developing a new system, before actually proceeding to develop it.

You need to do a feasibility study to figure out whether it is worth proceeding with requirement analysis and design. On the other hand unless you have some idea of the requirements and the design, you will really not have sufficient information to do a proper cost benefit analysis. There are several schools of thought where the exact chronology of the feasibility study is concerned. Some textbooks and experts position it after the requirement analysis is done. Whereas some others argue that the feasibility study ought to be the first step. However one will generally find that in the latter case some form of an abridged requirement analysis is actually hidden within the feasibility study itself!

The solution that we would like to advocate is that at the outset of the project the feasibility study should indeed be done. This could make use of a broad overview of the requirements, some top level estimates on resource requirements, some assumptions regarding outlays and benefits based on past project experiences and expert judgements. This would be a good starting point, atleast in terms of having some objective assessment regarding the feasibility of the exercise. As the project proceeds the details of these can always be added, refined or modified. The feasibility can always be reevaluated and reassessed at various stage of the project. It is not very often that there are fundamental differences at a later stage, which might result in shelving the entire project.

What are the Different aspects of Conducting a Feasibility Study? : A feasibility study has to begin with understanding the issues on hand, including the various organization factors that may impact the system. This is followed by looking at the proposed system from various angles to figure out whether there are likely to be any hitches and whether it is worth pursuing the project any further. Broadly speaking therefore, a feasibility study has to look at the following:

1. Need Analysis
2. Economic feasibility
3. Technical feasibility
4. Legal feasibility
5. Evaluation of alternatives

Need Analysis:: The starting point of any system is to understand what are the issues on hand and therefore what needs to be done. Quite often, this task is not as simple as it may seem. In fact the same problem across different organizations could be handled differently depending on various issues like organizational culture, urgency of the issue, resource constraints etc. One could at this stage therefore look at the following:

Background Information of the Organization: This would include getting familiar with the industry in which organization operates, the performance of the organization over the years, its guiding philosophies, the overall organizational objectives and goals and plans for the next few years.

Understanding the Current Issues to be Tackled: One has to get a comprehensive understanding of what are the issues on hand and what are the management's expectations. The new system would need to handle not only the problems with the current system, but also attempt to give some additional advantages and improvements over the current system.

Understand the User Profile : A common mistake that people generally make is merely obtain the organizational chart and assume that they have understood all they need to know. This could prove to be a very expensive mistake. The reasons are two-fold. Firstly, the exceptions of each user group could be different as far as the proposed system is concerned. Secondly, the profiles of each of these user groups could be vastly different as far as capabilities and exposure to technology is concerned. Generally the people who operate the system (in terms of the daily input and output jobs) are people in the clerical cadre. Most organizational charts do not give information pertaining to them. It is important to get such information to get an idea of the quality of manpower resource available.

After request clarification, analyst proposes some solutions. After that for each solution it is checked whether it is practical to implement that solution. This is done through feasibility study. In this various aspects like whether it is technically or economically feasible or not. So depending upon the aspect on which feasibility is being done it can be categorized into four classes:

- Technical
- Economic
- Legal
- Operational

The outcome of the feasibility study should be very clear. It should answer the following Issues.

- Is there an alternate way to do the job in a better way?
- What is recommended?

2.1. Technical Feasibility :

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many details will be available on the detailed design of the system, making it difficult to assess issues like performance, costs (on account of the kind of technology to be deployed) etc. So, normally a number of assumptions are made at this stage and the reliability of the analysis depends almost exclusively on the technical expertise of the person(s) doing it. Analyzing the technical feasibility is dependent to a large extent on Risk Analysis. The greater the risk element, the less feasible the system.

A number of issues have to be considered while doing a technical analysis. The important ones are:

1. Understand the Different Technologies Involved in the Proposed System: Before commencing the project, we have to be very clear about what are the technologies that are going to be required for the development of the new system. It has to be understood as to what are likely to be the associated costs. If certain elements seem to be too expensive considering the investments that the organization can make, alternate technologies need to be explored.

2. Find Out whether the Organization Currently Possesses the Required Technologies : Here we need to ask ourselves questions such as - "Is the required technology available with us?", if so then is the capacity sufficient? For instance - "Will our current printer be able to handle the new reports and forms required of the new system?" If the answers to the above is no, then we must ask ourselves, "Can we get this technology or additional resources? If we cannot afford the technology, then the alternative that requires the technology is not practical.

If new and old technologies need to coexist, then the feasibility of the same needs to be explored. For instance, if the new system requires new communication equipment and software, can it be interfaced easily with the existing systems, and so on.

If there are any new tools/techniques/algorithms/methodologies that are proposed to be used, then availability of these and the cost of procuring them and the viability of the same need to be worked out.

3. Find Out whether the Necessary Expertise Exists to use the Techniques and Technologies Involved: It is not enough to come up with a solution which is based on the latest technologies and is also cost effective. It should be assured whether the development team possesses the necessary skills to use and apply these technologies. If not it has to be seen whether these skills can be imparted through training. This will have implications on both costs as well as project schedules and is hence a very important consideration.

It is also extremely important to find out whether the eventual users will be comfortable using these technologies. Even if the system is developed by an external agency, it will ultimately be operated by the users. If they are not familiar with the new technologies it has to be seen whether adequate training can bridge the gap and if so the costs for the same have to be worked out. But what is most important is to ensure that all the user groups are positively inclined towards using the new system. If at this stage a major resistance is anticipated from the users then it may not be advisable to go ahead with the system, without satisfactorily addressing this issues, however elegant the system may look on paper.

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

2.2. Economic Feasibility :

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system. In economic feasibility, the most important is cost-benefit analysis. As the name suggests, it is an analysis of the costs to be incurred in the system and benefits derivable out of the system.

Cost Benefit Analysis : Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the costs, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected costs.

The cost of an information system involves the development cost and maintenance cost. The development costs are one time investment whereas maintenance costs are recurring. The development cost is basically the costs incurred during the various stages of the system development.

Each phase of the life cycle has a cost. Some examples are:

Personnel	Equipment
Supplies	Overheads
Consultants' fees.	

Cost and Benefit Categories : In performing CBA it is important to identify cost and benefit factors. Cost and benefits can be categorized into the following categories.

There are several cost factors/elements. These are hardware, personnel, facility, operating, and supply costs.

In a broad sense the costs can be divided into two types

1. Development Costs- Development costs that are incurred during the development of the system are one time investment.

Wages
Equipment

2. Operating Costs- It includes day to day activity cost that will be incurred during the development of the system e.g.

Wages
Supplies
Overheads

Another classification of the costs can be:

Hardware/software Costs: It includes the cost of purchasing or leasing of computers and it's peripherals. Software costs involves required s/w costs.

Personnel Costs: : It is the money, spent on the people involved in the development of the system.

These expenditures include salaries, other benefits such as health insurance, conveyance allowance, etc.

Facility Costs : Expenses occurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lighting, and air conditioning.

Operating Costs : Operating costs are the expenses required for the day to day running of the system. This includes the maintenance of the system. That can be in the form of maintaining the hardware or application programs or money paid to professionals responsible for running or maintaining the system.

Supply Costs : These are variable costs that vary proportionately with the amount of use of paper, ribbons, disks, and the like. These should be estimated and included in the overall cost of the system.

Benefits : We can define benefit as

Profit or Benefit = Income - Costs

Benefits can be accrued by :

— Increasing income, or
— Decreasing costs, or
— both

The system will provide some benefits also. Benefits can be tangible or intangible, direct or indirect. In cost benefit analysis, the first task is to identify each benefit and assign a monetary value to it.

The two main benefits are improved performance and minimized processing costs. Further costs and benefits can be categorized as

Tangible or Intangible Costs and Benefits : Tangible cost and benefits can be measured. Hardware costs, salaries for professionals, software cost are all tangible costs. They are identified and measured.. The purchase of hardware or software, personnel training, and employee salaries are example of tangible costs. Costs whose value cannot be measured are referred as intangible costs. The cost of breakdown of an online system during banking hours will cause the bank lose deposits.

Benefits are also tangible or intangible. For example, more customer satisfaction, improved company status, etc are all intangible benefits. Whereas improved response time, producing error free output such as producing reports are all tangible benefits. Both tangible and intangible costs and benefits should be considered in the evaluation process.

Direct or Indirect Costs and Benefits : From the cost accounting point of view, the costs are treated as either direct or indirect. Direct costs are having rupee value associated with it. Direct benefits are also attributable to a given project. For example, if the proposed system that can handle more transactions say 25% more than the present system then it is direct benefit.

Indirect costs result from the operations that are not directly associated with the system. Insurance, maintenance, heat, light, air conditioning are all indirect costs.

Fixed or Variable Costs and Benefits : Some costs and benefits are fixed. Fixed costs don't change. Depreciation of hardware, insurance, etc are all fixed costs. Variable costs are incurred on regular basis. Recurring period may be weekly or monthly depending upon the system. They are proportional to the work volume and continue as long as system is in operation.

Fixed benefits don't change. Variable benefits are realized on a regular basis.

Performing Cost Benefit Analysis (CBA)

Example:

Cost for the proposed system (figures in Rs lakhs)

Year / Cost Category	1	2	3	4	5
Hardware	40				
Software	22				
Personnel	8	22	24	26	28
Maintenance	0	4	6	8	10
Cost at year end	70	16	30	34	38
Cumulative costs	70	96	126	150	188

Benefit for the proposed system

Year / Benefits	1	2	3	4	5
From finished reports	15	15	15	15	15
Increase in sales	25	40	50	60	70
Benefits at year end	40	55	65	75	85
Cumulative benefits	40	95	160	235	310

Profit = Benefits - Costs
= 310 -188
= Rs. 122 lakhs

Since we are gaining, this system is feasible. Steps of CBA can briefly be described as:

Estimate the development costs, operating costs and benefits Determine the life of the system

When will the benefits start to accrue?

When will the system become obsolete?

Determine the interest rate

This should reflect a realistic low risk investment rate.

Select Evaluation Method : When all the financial data have been identified and broken down into cost categories, the analyst selects a method for evaluation.

There are various analysis methods available. Some of them are following :

1. Present value analysis
2. Payback analysis
3. Net present value
4. Net benefit analysis
5. Cash-flow analysis
6. Break-even analysis

Present value analysis : It is used for long-term projects where it is difficult to compare present costs with future benefits. In this method cost and benefit are calculated in term of today's value of investment.

To compute the present value, we take the following formula

$$PV = \frac{F}{(I + I)^n}$$

Where,

i is the rate of interest &

n is the time

Example:

Present value of Rs. 3000 invested at 15% interest at the end of sth year is calculates as :

P = 3000/(1 + .15)5

= 1491.53

Table below shows present value analysis for 5 years

Year	Estimation Future Value	Present Value	Cumulative Present Value of Benefits
1	3000	2608.69	2608.69
2	3000	2268.43	4877.12
3	3000	1972.54	6849.66
4	3000	1715.25	8564.91
5	13000	1491.53	10056.44

Net Present Value : NPA

The net present value is equal to benefits minus costs. It is expressed as a percentage of the investment.

Net Present Value= Costs - Benefits

% = Net Present Value/Investments

Example : Suppose total investment is 50000 and benefits are Rs. 80000

Then Net Present Value= Rs. (80000 - 50000)

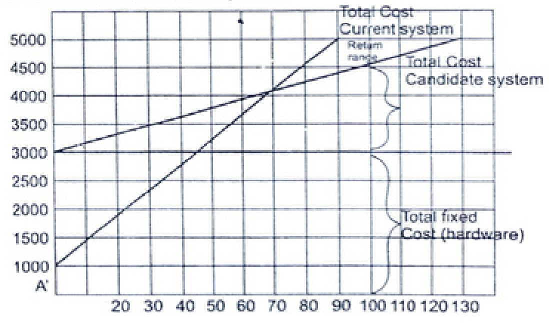
= Rs. 30000

% = 30000/80000

= .375

Break-even Analysis : Once we have determined what is estimated cost and benefit of the system it is also essential to know in what time will the benefits are realized. For that break-even analysis is done.

Break-even is the point where the cost of the proposed system and that of the current one are equal. Break-even method compares the costs of the current and candidate systems. In developing any candidate system, initially the costs exceed those of the current system. This is an investment period. When both costs are equal, it is break-even. Beyond that point, the candidate system provides greater benefit than the old one. This is return period.



Processing volume (1,000s)

Fig. 1 : Break-even Chart

Fig. 1 is a break-even chart comparing the costs of current and candidate systems. The attributes are processing cost and processing volume. Straight lines are used to show the model's relationships in terms of the variable, fixed, and total costs of two processing methods and their economic benefits. B' point is break-even. Area after B' is return period. A' AB' area is investment area. From the chart, it can be concluded that when the transaction are lower than 70,000 then the present system is economical while more than 70,000 transaction would prefer the candidate system.

Cash-flow Analysis: Some projects, such as those carried out by computer and word processors services, produce revenues: from an investment in computer systems. Cash-flow analysis keeps track of accumulated costs and revenues on a regular basis.

2.3. Operational Feasibility :

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance: from users that will effect the possible application benefits? The essential questions that help in testing the operational feasibility of a system are following.

Does Management Support the Project? : Are the users not happy with current business practices? Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.

Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system.

Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system effect the customers in considerable way?

2.4. Legal Feasibility :

It includes study concerning contracts, liability, violations, and legal other traps frequently unknown to the technical staff.

The development of certain sensitive systems may have some legal ramifications. In such cases it has to be ascertained whether the development of the new system may result in any infringement, violation, contracts or liabilities. This is an area where the technical people may not have the necessary information and therefore legal experts may need to be consulted on such issues.

3. Evaluating Alternatives :

Once the feasibility analysis is done for the different options that are being considered, it might turn out that more than one option is feasible. The next step then would be to decide which of these options is most suitable. In case the analysis has been purely quantitative, the comparison becomes rather straightforward. This option with the lowest cost and maximum returns naturally becomes the most viable option. Here it is important to consider the net present value of each of these options otherwise comparing absolute rupee figures over a time period could often become confusing and misleading.

However in most projects, the decision is seldom based on a mere quantitative comparison. There will always be a number of qualitative and intangible issues which greatly influence the decision. But even in such cases it is always useful to list these factors, so that the eventual decision making is simpler.

4. Documenting of the Feasibility Study

The findings of a feasibility study are generally documented in what is called a Feasibility Report. The degree of detail in such reports would be greatly dependent on the nature of project, the organization and the scale of investments. The contents of this report would be as mentioned below :

1. Introduction
   - 1.1 Statement of problem
   - 1.2 Implementation environment
   - 1.3 Constraints
2. Management summary and recommendations
   - 2.1 Important findings
   - 2.2 Comments
   - 2.3 Recommendations
   - 2.4 Impact
3. Alternatives
   - 3.1 Alternative system configurations
   - 3.2 Criteria used in selecting the final approach
4. System Description
   - 4.1 Abbreviated statement of scope
   - 4.2 Feasibility of allocated elements
5. Cost benefit analysis
6. Evaluation of technical risk Legal ramifications
7. Other project specific topics

Evolving the Project Proposal :

Once the feasibility study is completed and the best option is chosen, then the outcome of the same has to be presented to the organization. This is done in the form of Project proposal. The purpose of this is to present all the details of the chosen option and also give additional details in terms of calendar schedules etc., so that it gives the management an overview of the entire project. Strictly speaking, calendar schedules for a project should be arrived at based on the effort estimate for the project. However, in reality, calendar schedules are influenced by a whole host of other factors, the most important of which could be the stipulation made by the organization. So generally at this stage some rough idea of the likely schedule is given to the top management. This may be later adjusted or augmented by suitable staffing and providing other resources, once a proper estimation exercise is done. Broadly, this would contain the following:

- Enumeration of problem areas
- Identification of application portfolio to deal with these problems
- Expected benefits from automation
- Prerequisites for development and implementation including hardware, commodity software, manpower and training needs
- Phasewise demarcation of activities
- Indicative calendar schedules for software development and implementation

Based on the above the management decides whether or not to accept the proposal. Once they decide to accept it, the stage is set for proceeding with the next phase of requirement analysis.

Questions

Very Short Questions

1. Define Preliminary Analysis
2. What is Request clarification ?
3. What is Feasibality study
4. What is need analysis in Feasibality study
5. What is CBA ?

Short Questions

1. Describe steps involved in Preliminary Analysis.
2. What is Technical feasibility.
3. What is Operational feasibility
4. How will you propare documentation of feasibality study ?
5. What is legal feasibality ?

Long Question :

1. What is preliminary Analysis ?
2. What do you mean by cost-Benefit Analysis and discuss all type of benefits and cost ?
3. What is Estimation ? Discuss all Estimation methods.
4. What do you mean by feasibality analysis. Discuss different types of feasibality analysis ?
5. What is Economic feasibality ? Discuss in detail various methods of finding Economic feasibality ?

❑❑❑

5 REQUIREMENT ANALYSIS

Objectives

1. Introduction

In engineering, a requirement is a singular documented need of what a particular product or service should be or do. It is most commonly used in a formal sense in systems engineering or software engineering. It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user. In the classical engineering approach, sets of requirements are used as inputs into the design stages of product development.

The requirements development phase may have been preceded by a feasibility study, or a conceptual analysis phase of the project. The requirements phase may be broken down into requirements elicitation (gathering the requirements from stakeholders), analysis (checking for consistency and completeness), specification (documenting the requirements) and verification (making sure the specified requirements are correct).

Requirements Analysis is the process of understanding the customer needs and expectations from a proposed system or application and is a well-defined stage in the Software Development Life Cycle model.

Requirements are a description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of 'what' an application is expected to do.

Given the multiple levels of interaction between users, business processes and devices in global corporations today, there are simultaneous and complex requirements from a single application, from various levels within an organization and outside it as well.

The Software Requirements Analysis Process covers the complex task of eliciting and documenting the requirements of all these users, modeling and analyzing these requirements and documenting them as a basis for system design.

A dedicated and specialized Requirements Analyst is best equipped to handle the job. The Requirements Analysis function may also fall under the scope of Project Manager, Program Manager or Business Analyst, depending on the organizational hierarchy.

Software Requirements Analysis and Documentation Processes are critical to software project success. Requirements Engineering is an emerging field which deals with the systematic handling of requirements.

Why is Requirements Analysis necessary? : Studies reveal that inadequate attention to Software Requirements Analysis at the beginning of a project is the most common cause for critically vulnerable projects that often do not deliver even on the basic tasks for which they were designed. There are instances of corporations that have spent huge amounts on software projects where the end application eventually does not perform the tasks it was intended for.

Software companies are now investing time and resources into effective and streamlined Software Requirements Analysis Processes as a prerequisite to successful projects that align with the client's business goals and meet the project's requirement specifications.

2. Steps in the Requirements Analysis Process

I. Fix system boundaries : This initial step helps in identifying how the new application integrates with the business processes, how it fits into the larger picture and what its scope and limitations will be.

II. Identify the customer : In more recent times there has been a focus on identifying who the 'users' or 'customers' of an application are. Referred to broadly as the 'stake holders', these indicate the group or groups of people who will be directly or indirectly impacted by the new application.

By defining in concrete terms who the intended user is, the Requirements Analyst knows in advance where he has to look for answers. The Requirements Elicitation Process should focus on the wish-list of this defined group to arrive at a valid requirements list.

III. Requirements elicitation : Information is gathered from the multiple stakeholders identified. The Requirements Analyst draws out from each of these groups what their requirements from the application are and what they expect the application to accomplish.

Considering the multiple stakeholders involved, the list of requirements gathered in this manner could run into pages. The level of detail of the requirements list is based on the number and size of user groups, the degree of complexity of business processes and the size of the application.

(a) Problems faced in Requirements Elicitation

- Ambiguous understanding of processes
- Inconsistency within a single process by multiple users
- Insufficient input from stakeholders
- Conflicting stakeholder interests
- Changes in requirements after project has begun

A Requirements Analyst has to interact closely with multiple work-groups, often with conflicting goals, to arrive at a bona fide requirements list. Strong communication and people skills along with sound programming knowledge are prerequisites for an expert Requirements Analyst.

(b) Tools used in Requirements Elicitation

Traditional methods of Requirements Elicitation included stakeholder interviews and focus group studies. Other methods like flowcharting of business processes and the use of existing documentation like user manuals, organizational charts, process models and systems or process specifications, on-site analysis, interviews with end-users, market research and competitor analysis were also used extensively in Requirements Elicitation.

However current research in Software Requirements Analysis Process has thrown up modern tools that are better equipped to handle the complex and multi layered process of Requirements Elicitation. Some of the current Requirements Elicitation tools in use are:

- Prototypes
- Use cases
- Data flow diagrams
- Transition process diagrams
- User interfaces

IV. Requirements Analysis Process : Once all stakeholder requirements have been gathered, a structured analysis of these can be done after modeling the requirements. Some of the Software Requirements Analysis techniques used are requirements animation, automated reasoning, knowledge-based critiquing, consistency checking, analogical and case-based reasoning.

V. Requirements Specification : Requirements, once elicited, modeled and analyzed should be documented in clear, unambiguous terms. A written requirements document is critical so that its circulation is possible among all stakeholders including the client, user-groups, the development and testing teams. Current requirements engineering practices reveal that a well-designe , clearly documented Requirements Specification is vital and serves as a:

- Base for validating the stated requirements and resolving stakeholder conflicts, if any

- Contract between the client and development team
- Basis for systems design for the development team
- Bench-mark for project managers for planning project development lifecycle and goals
- Source for formulating test plans for QA and testing teams
- Resource for requirements management and requirements tracing
- Basis for evolving requirements over the project life span

Software requirements specification involves scoping the requirements so that it meets the customer's vision. It is the result of collaboration between the end-user who is often not a technical expert, and a Technical/Systems Analyst, who is likely to approach the situation in technical terms.

The software requirements specification is a document that lists out stakeholders' needs and communicates these to the technical community that will design and build the system. The challenge of a well-written requirements specification is to clearly communicate to both these groups and all the sub-groups within.

To overcome this, Requirements Specifications may be documented separately as

- **User Requirements** - written in clear, precise language with plain text and use cases, for the benefit of the customer and end-user
- **System Requirements** - expressed as a programming or mathematical model, addressing the Application Development Team and QA and Testing Team.

Requirements Specification serves as a starting point for software, hardware and database design. It describes the function (Functional and Non-Functional specifications) of the system, performance of the system and the operational and user-interface constraints that will govern system development.

VI. Requirements Management : Requirements Management is the comprehensive process that includes all aspects of software requirements analysis and additionally ensures verification, validation and traceability of requirements. Effective requirements management practices guarantee that all system requirements are stated unambiguously, that omissions and errors are corrected and that evolving specifications can be incorporated later in the project lifecycle.

3. Product versus process requirements

Projects are subject to three sorts of requirements. Business Requirements describe in business terms what must be delivered or accomplished to provide value. Product Requirements describe the system or product which is one of several possible ways to accomplish the business requirements. Process Requirements describe the processes the developing organization must follow and the constraints that they must obey.

The Product and Process requirements are closely linked. Process requirements are often imposed as a way of achieving some higher-level Product requirement. For example, a maximum development cost requirement (a Process requirement) may be imposed to help achieve a maximum sales price requirement (a Product requirement); a requirement for the product to be maintainable (a Product requirement) often is traced to by requirements to follow particular development styles (e.g., object-oriented programming), style-guides, or a review/inspection process (Process requirements). Both user along with system requirements are vital for every system to develop(Surafel)Requirements show what elements and function are necessary for the particular project.

4. Requirements in systems and software engineering

In systems engineering, a requirement can be a description of what a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called 'non-functional requirements', or 'performance requirements' or 'quality of service requirements.' Examples of such requirements include availability, testability, maintainability, and ease-of-use.

A collection of requirements define the characteristics or features of the desired system. A 'good' list of requirements generally avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Describing how the system should be implemented is known as implementation bias.

In software engineering, the same meaning of requirements apply, except that the focus of interest is the software itself.

5. Some factors in developing requirements

Classification : Requirements are typically placed into these categories:

Functional requirements describe the functions that the system is to execute; for example, formatting some text or modulating a signal. They are sometimes known as capabilities.

Functional requirements are-

Understanding functional needs-

a. Understanding High Level UseCases
   i. For example, At a time how many of them can use the website. What would happen if 10,000 people uses within a min?
b. Understanding User Interface Needs
c. Understanding Architecture & Technology Selection
   i. Selection of best suitable technology based on the client's requirements.
   ii. Manpower skills and infrastructure needs

Understanding process needs-

a. Process and Protocol specifications

Understanding business needs-

Non-functional requirements are the ones that act to constrain the solution. Nonfunctional requirements are sometimes known as constraints or quality requirements. Non-Functional requirements are usability, scalability, extensibility, performance, security and maintainability. They are all very important for a project but not non-functional requirements.

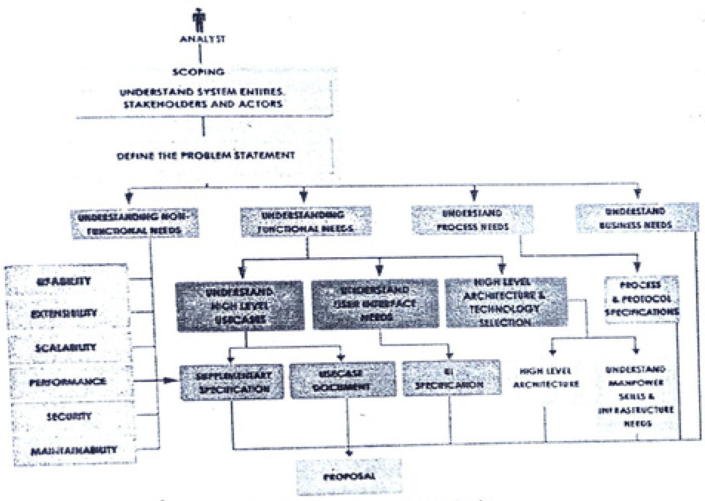

ANALYST
SCOPING
UNDERSTAND SYSTEM ENTITIES, STAKEHOLDERS AND ACTORS
DEFINE THE PROBLEM STATEMENT
UNDERSTANDING NON-FUNCTIONAL NEEDS
UNDERSTANDING FUNCTIONAL NEEDS
UNDERSTAND PROCESS NEEDS
UNDERSTAND BUSINESS NEEDS
USABILITY
EXTENSIBILITY
SCALABILITY
PERFORMANCE
SECURITY
MAINTAINABILITY
UNDERSTAND HIGH LEVEL USECASES
UNDERSTAND USER INTERFACE NEEDS
HIGH LEVEL ARCHITECTURE & TECHNOLOGY SELECTION
PROCESS & PROTOCOL SPECIFICATIONS
SUPPLEMENTARY SPECIFICATION
USECASE DOCUMENT
UI SPECIFICATION
HIGH LEVEL ARCHITECTURE
UNDERSTAND MANPOWER SKILLS & INFRASTRUCTURE NEEDS
PROPOSAL

Fig. 1 - Requirement Analysis

From the above diagram, analyst is the technical person. Scoping contains the Business entities, stake holders and actors. Stake holders and actors are business decision makers and influencers. Influencers will interconnect the client and the service provider like iLink. The Requirement analysis document contains both functional and non-functional requirements.

They can be further classified according to whether they are performance requirements, maintainability requirements, safety requirements, reliability requirements, or one of many other types of requirements.

Good requirements

The characteristics of good requirements are variously stated by different writers, with each writer generally emphasizing the characteristics most appropriate to his general discussion or the specific technology domain being addressed. However, the following characteristics are generally acknowledged.

Characteristic	Explanation
Cohesive	The requirement addresses one and only one thing.
Complete	The requirement is fully stated in one place with no missing information.
Consistent	The requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation.
Correct	The requirement meets all or part of a business need as authoritatively stated by stakeholders.
Current	The requirement has not been made obsolete by the passage of time.
Externally Observable	The requirement specifies a characteristic of the product that is externally observable or experienced by the user. "Requirements" that specify internal architecture, design, implementation, or testing decisions are properly constraints, and should be clearly articulated in the Constraints section of the Requirements document.
Feasible	The requirement can be implemented within the constraints of the project.
Unambiguous	The requirement is concisely stated without recourse to technical jargon, acronyms (unless defined elsewhere in the Requirements document), or other esoteric verbiage. It expresses objective facts, not subjective opinions. It is subject to one and only one interpretation. Vague subjects, adjectives, prepositions, verbs and subjective phrases are avoided. Negative statements and compound statements are prohibited.
Mandatory	The requirement represents a stakeholder-defined characteristic the absence of which will result in a deficiency that cannot be ameliorated.
Verifiable	The implementation of the requirement can be determined through one of four possible methods: inspection, analysis, demonstration, or test.

Testability

Most requirements should be testable. If this is not the case, another verification method should be used instead (e.g. analysis, inspection or review of design). Testable requirements are an important component of validation.

Certain requirements, by their very structure, are not testable. These include requirements that say the system shall never or always exhibit a particular property. Proper testing of these requirements would require an infinite testing cycle. Such requirements are often rewritten to state a more practical time period.

Un-testable non-functional requirements may still be kept as a documentation of customer intent; however they are usually traced to process requirements that are determined to be a practical way of meeting them. For example, a non-functional requirement to be free from backdoors may be satisfied by replacing it with a process requirement to use pair programming.

Testability is essentially a form of clarity, which indeed is necessary but can divert attention from other important issues. A requirement can be testable yet incorrect; and assessing testability often will not detect incorrect requirements. Moreover, testability is totally irrelevant with regard to a requirement which has been overlooked. Mere analysis, inspection, or review alone will find some of these issues but generally is far weaker than usually is realized. There are more than 21 more powerful ways to test or evaluate requirements adequacy and more than 15 ways to strengthen testing or evaluation of design adequacy.

Requirements analysis

Requirements are prone to issues of ambiguity, incompleteness, and inconsistency. Techniques such as rigorous inspection have been shown to help deal with these issues. Ambiguities, incompleteness, and inconsistencies that can be resolved in the requirements phase typically cost orders of magnitude less to correct than when these same issues are found in later stages of product development. Requirements analysis strives to address these issues.

There is an engineering trade off to consider between requirements which are too vague, and those which are so detailed that they take a long time to produce begin to limit the implementation options available are costly to produce

Documenting requirements

Requirements are usually written as a means for communication between the different stakeholders. This means that the requirements should be easy to understand both for normal users and for developers. One common way to document a requirement is stating what the system shall do. Example: "The contractor shall deliver the product no later than xyz date." Other ways include use cases and user stories.

Changes in requirements

Requirements generally change with time. Once defined and approved, requirements should fall under change control. For many projects, requirements are altered before the system is complete. This is partly due to the complexity of computer software and the fact that users don't know what they want before they see it. This characteristic of requirements has led to requirements management studies and practices.

Disputes regarding the necessity of rigour in software requirements

Some modern software engineering methodologies like extreme programming question the need for rigorously describing software requirements, which they consider a moving target. Instead, they describe requirements informally using user stories (short summaries fitting on an index card explaining one aspect of what the system should do), and compose a series of acceptance test cases for this user story.

Questions

Very Short Questions :

1. What is Request Analysis ?
2. What is Feasibality study ?
3. What is need analysis in Feasibality study ?
4. What is CBA ?

Short Questions :

1. What is requirment elicitation ?
2. What are the steps involved in requirement analysis process ?
3. What do you mean by product requirement ?
4. What do you mean by process requirement ?

Long Question :

1. Name some requirement gathering techniques
2. When should you end the process of gathering requirements?
3. Why is a simple change control process necessary for agile projects?
4. Why do requirements change?
5. What are the three levels of requirements?
6. What do you mean by Requirement Analysis ?
7. Why is necessary Requirement Analysis ? Discuss all steps involved in Requirement Analysis Process

❑❑❑

6

FACT GATHERING TECHNIQUES

Objectives

1. Fact Gathering Techniques :

Fact finding/gathering is an important activity in system investigation. In this stage, the functioning of the system is to be understood by the system analyst to design the proposed system. Various methods are used for this and these are known as fact-finding/gathering techniques. The analyst needs to fully understand the current system.

The analyst needs data about the requirements and demands of the project undertaken and the techniques employed to gather this data are known as fact-finding techniques. Various kinds of techniques are used for gathering data and the most popular among them are interviews, questionnaires, record reviews, case tools and also the personal observations made by the analyst himself. Each of these techniques is further dealt with.

1.1. Interviews :

Interview is a very important data gathering technique as in this the analyst directly communicates with system and the potential user of the proposed system.

One very essential aspect of conducting the interview is that the interviewer should first establish a proper environment with the interviewee. It should also be taken into account that the interviewee may or may not be a technician and the analyst should prefer to use day to day language instead of jargon and technical terms.

For the interview to be a success the analyst should be appropriately prepared, as he needs to be beforehand aware of what exactly needs to be asked and to what depth. Also he should try to gather maximum relevant information and data. As the number and type of respondents vary, the analyst should be sensitive to their needs and nature.

The advantage of interviews is that the analyst has a free hand and he can extract most of the information from the concerned people but then as it is a very time consuming method, he should also employ other means such as questionnaires, record reviews, etc. This may also help the analyst to verify and validate the information gained. Interviewing should be approached, as logically as possible and from a general point of view the following guides can be very beneficial for a successful interview.

1. Set the stage for the interview.

2. Establish the environment put the interviewee at case. 3. Phrase questions clearly and sequentially.

4. Be a good listener; avoid arguments.

5. Evaluate the outcome of the interview.

The interviews can be of two types namely structured and unstructured.

1.1.1. Structured Interview : Structured interviews are those where the interviewee is asked a standard set of questions in a particular order. All interviewees are asked the same set of questions.

The questions are further divided in two kinds of formats for conducting this type of interview. The first is the open-response format in which the respondent is free to answer in his own words. An example of open-response is "Why are you not satisfied with the current attendance processing method?" The other option is of closed-response format, which limits the respondents to opt their answer from a set of already prescribed choices. An example of such a question might be "Are you satisfied with the current attendance processing methods?" or "Do you think that the manual attendance processing procedure be changed with some automated procedure?"

1.1.2. Unstructured Interviews : The unstructured interviews are undertaken in a question-and-answer format. This is of a much more flexible nature than the structured interview and can be very rightly used to gather general information about the system.

Here the respondents are free to answer in their own words. In this way their views are not restricted. So the interviewer get a bigger area to further explore the issues pertaining to a problem.

1.1.3. Structured Vs Unstructured : Each of the structured and unstructured interview methods has its own merits and demerits. We will consider the structured format first and that too its advantages. This method is less time consuming and the interviewer need not be a trained person.

It is easier to evaluate objectively since answers obtained are uniform. On the other hand a high level of structure and mechanical questions are posed which may earn the disliking of the respondents. Also this kind of structure may not be appropriate for all situations and it further limits the respondent spontaneity.

In unstructured interviews the respondents are free to answer and present their views. Also there may be case that some issue might surface spontaneously while answering some other question. In that case the respondent can express views on that issue- also.

But at times, it may happen that the interview goes in some undesired direction and the basic facts for which the interview was organized do not get relieved. So the analyst should be careful while conducting interviews.

1.2. Questionnaires :

Questionnaires is another way of information gathering where the potential users of the system are given questionnaires to be filled up and returned to the analyst.

Questionnaires are useful when the analyst need to gather information from a large number of people. It is not possible to interview each individual. Also if the time is very short, in that case also questionnaires are useful. If the anonymity of the respondent is guaranteed by the analyst then the respondent answers the questionnaires very honestly and critically.

The questionnaire may not yield the results from those respondents who are busy or who may not give it a reasonable priority.

The analyst should sensibly design and frame questionnaires with clarity of it's objective so as to do justice to the cost incurred on their development and distribution. Questionnaires are of two types i.e. open-response based and the closed-response based.

1.2.1. Open Response : The objective of open-response questionnaire is to gather information and data about the essential and critical design features of the system. The open-ended question requires no response direction or specific response.

This form is also used to learn about the feelings, opinions and experiences of the respondents. This information helps in the making the system effective because the analyst can offer subsequent modifications as per the knowledge gained.

1.2.2. Closed Response : The objective of closed-response questionnaire is to collect the factual information of the system. It gives an insight in how the people dealing with the system behave and how

comfortable are they with it. In this case the respondents have to choose from a set of given responses. Thus the respondent can express their liking for the most favorable one from the possible alternatives.

The closed questions can be of various types and the most common ones are listed below.

1. Fill-in-the-blanks.
2. Dichotomous i.e. Yes or No type.
3. Ranking scale questions ask the respondents to rank a list of items in the order of importance or preference.
4. Multiple-choice questions which offer respondents few fixed alternatives to choose from.
5. Rating scale questions are an extension of the multiple-choice questions. Here the respondent is asked to rate certain alternatives on some given scale.

1.2.3 Open Vs Closed : The basic comparison between the two can be made on the grounds of the format used. Open form offers more flexibility and freedom to the respondents whereas the closed form is more specific in nature.

Open-ended questionnaires are useful when it is required to explore certain situation. They also require a lot of time of the analyst for evaluation.

Closed questionnaires are used when factual information is required. Closed questions are quick to analyze but typically most costly to prepare but they are more suitable to obtain factual and common information.

It is the job of the analyst to decide which format should be employed and what exactly should be it's objective. The care should be taken to ensure that all the parts of the form are easy to understand for the respondents so that they can answer with clarity.

1.3. Record Reviews :

Records and reports are the collection of information and data accumulated over the time by the users about the system and it's operations. This can also put light on the requirements of the system and the modifications it has undergone. Records and reports may have a limitation if they are not up-to-date or if some essential links are missing. All the changes, which the system suffers, may not be recorded. The analyst may scrutinize the records either at the beginning of his study which may give him a fair introduction about the system and will make him familiar with it or in the end which will provide the analyst with a comparison between what exactly is/was desired from the system and it's current working.

One drawback of using this method for gathering information is that practically the functioning of the systems is generally different from the procedure shown in records. So analyst should be careful in gathering in gathering information using this method.

1.4. On-site Observation :

On-site observations are one of the most effective tools with the analyst where the analyst personally goes to the site and discovers the functioning of the system. As an observer, the analyst 'can gain first hand knowledge of the activities, operations, processes of the system on-site, hence here the role of an analyst is of an information seeker. This information is very meaningful as it is unbiased and has been directly taken by the analyst. This exposure also sheds some light on the actual happenings of the system as compared to what has already been documented, thus the analyst gets closer to the system. This technique is also time-consuming and the analyst should not jump to conclusions or draw inferences from small samples of observation rather the analyst should be more patient in gathering the information. This method is however less effective for learning about people's perceptions, feelings and motivations.

1.5 CASE STUDY: Library Management System :

In previous chapters we discussed how the analyst performed the preliminary analysis. But we didn't look into the actual methods that the analyst employed to gather the information about the system In our case the analyst used on-site observations, interviewed the staff members and used questionnaires for both staff and members of the library. Now, we will see how our analyst employed these methods.

Fact Finding Techniques

On-site Observation

Our analyst wanted to see the functioning of library. So analyst visited the library for two days and observed librarian issuing and returning books. 'The analyst also inspected the place where the cards are stored and from that it was seen that it was a real mess. To see if a particular book is already issued it is a difficult and effort intensive process. The analyst also saw the records for books, members, and accounts. From site visit our analyst had a good understanding of the functioning of the system.

After this, the analyst performed some personal interviews of library staff and few members. In the next section we'll look at these interviews.

Interviews : Interviews are useful to gather information from individuals. Given below is the interview between the analyst and one of the librarians, during the information gathering stage of the development of our library system.

Analyst's interview with Librarian :

Analyst : Hi, I have come to know regarding the functioning of your library.

Librarian : Hello, do come in. I was waiting for you.

Analyst : I'll come straight to the point. Don't hesitate, you can be as much open you want. There are no restrictions.

Librarian : I'll give you my complete support

Analyst : Tell me are you excited about the idea of having an automated system for your library?

Librarian : Yes, I do. Very much. After all it's gonna reduce our loads of work.

Analyst : Will you elaborate on it?

Librarian : Major problem is managing the cards of members. There are so many of them. Many times cards get lost. Then we have to issue a duplicate card for it. But there is a flaw in it. It is difficult to find out if it is genuinely the case. Member can lie about it so that he/she gets an extra card. And we can't do anything about it.

Analyst : What do you think be ideal solution to this?

Librarian : There should be no cards at all. All the information should be put into computer. It'll be easy for us to check how many books we have already issued to a particular member.

Analyst : How often you get new members?

Librarian : Very often. At about 100 to 150 members in a month. But for two months we have free zed the membership because it is already very difficult to manage the existing 1500 members. But if this whole system gets computerised then we'll open the membership. From this system, the management hopes to earn huge revenues.

Analyst : Could you explain how?

Librarian : Look every month we get about 100-150 memberships requests. After the new system is built, we will open membership to our library. There is a membership fees to be paid. Management is planning to change the membership criteria. It is planning to increase fee from 1000 to 1000 for half yearly and 2000 for the whole year. So in this way, we plan to get huge revenues after we have an automated system.

Analyst : Do you have different member categories?

Librarian : No, we don't have any categorisation for members. All are treated at par.

Analyst : How many books are there? About 80,000 books

Librarian : Do you people keep records for them? Yes.

Analyst : Do you want facility of booking a particular title in advance?

Librarian : No we don't want any such facility. It is an overhead. So we don't have any such facility presently.

Analyst : How do you categorise your books?

Librarian : By subject.

Analyst : Would you prefer online registration for users rather than the printed form?

Librarian : Yes, we really would. Sometimes we lose these forms then we don't have any information about that particular member. It will be better to have it on computer.

Analyst : Do you have any other expectation or suggestion for the new system?

Librarian : It should be able to produce reports faster.

Analyst : Reports? I completely forgot about them. What reports you people produce presently?

Librarian : Well first is for books in the library, another for members listing, one for our current supplier of books, and reports for finance.

Analyst : Do you have some format for them?

Librarian : Yes we do have and we want that the same format be used by the new system.

Analyst : Yes we'll take care of that. Any other suggestions you want to give?

Librarian : No. You have already covered all the fields.

Analyst : Thanks for your cooperation. It was nice talking to you.

Librarian : My pleasure. Bye.

Our analyst took interviews of few members of the library in order to know about their viewpoint about the new system. One of such interview is given below:

Analyst interview with one member :

Venue : Reading Room

Analyst : Hello. If you are free, I need to ask you few questions.

Member : Sure. I pleasure.

Analyst : Do you know the library people are planning to have an automated system?

Member : Yes, I do and I'm feeling good about it.

Analyst : Are you ready to pay more if there is a computerised system?

Member : In the overall functioning is going to improve then I think no one will object to paying more. It should help us finding the books easily. But by what amount, it should matter.

Analyst : Well as far as I know they are planning to hike the membership fee from 700 to 1000 for half year and 2000 for full year.

Member : That would be too much. Then in that case, they should increase the number of books to be issued. Also the number of days a book can be kept by member should also be increased.

Analyst : What you do think, how much books to be allowed for issue and for how many days.

Member : Well these people should increase number of books from 3 to at least 5. And the number of days for which a book is kept should be increased by 4 days. Presently it is for 10 days. It should be 14 days. Only then the fee hike will be justified.

Analyst : Yes, they have such plans.

Member : Then it should not bother members.

Analyst : Are you keen on online registration of members instead of normal paper one?

Member : Yes. It'll be a good practice.

Analyst : Should there be a facility to reserve a book in advance?

Member : Presently they have many copies of a single title. Usually a book is always available. I never have felt the need to reserve a book in advance.

Analyst : On what basis a book should be categorised?

Member : Well, it should be on the basis of subject.

Analyst : What do you think on what basis a search for a particular book can be done?

Member : It can be searched using subject or title.

Analyst : How often you visit this library?

Member : Daily

Analyst : Do you think magazines and cassettes should be made available in the library?

Member : I think it's a good idea.

Analyst : Do you like this library?

Member : Yes, very much. That's why I come here daily.

Analyst : Have you ever recommended this library to your friends, relatives, or to your acquaintances?

Member : Yes I very often do.

Analyst : Till now, to how many you have recommended?

Member : About 35 people.

Analyst : And how many of them have actually become its members?

Member : 20 people.

Analyst : That's really nice. People actually take your recommendation very seriously. Thank You. It was nice talking to you.

Member : Thank You.

After interviewing different people, analyst got to know about their opinions.

Questionnaires :

Since the time was less it was not practical to interview every library staff So to get the opinion of all staff, the analyst distributed questionnaires to all of them. The questionnaire given to library staff is given below.

Instructions: Answer as specified by the format. Put NA for non-applicable situation.

1. What are your expectations out of the new system (computer based)? Rate the following on a scale of 1-4 giving a low value for low priority.
   a) better cataloguing
   b) better managing of users
   c) better account and books management
   d) computer awareness
   e) any other
2. How many users are you expecting?
   ____
3. How many books are there in library?
   ____
4. How you want the books to be categorized for searching (like by title, author name or by subject) ?
   ____
   ____
   ____
5. Is there any difference in the roles (privileges) of two members?
   Yes\No Please specify if Yes
   ____
   ____
6. Do you want facility of booking a title in advance Yes\No
7. Do you have data of book entered into some kind of database? Yes\No
8. How do you want users to be categorized?
   ____________________ or
   ____
9. Would you like online registration for users rather than printed form?Yes/No
10. Do you already have some existing categorization of books on the basis as specified in question 4 above Yes/ No
11. Any other specific suggestion / expectation out of the proposed system.
   ____
   ____

Fig. 1 : Questionnaire for library staff

In order to get the views of the existing members, the analyst also distributed questionnaires to the member also. The questionnaires used by analyst for the library members is shown in fig. 8.

Instruction: Answer as specified by the format. Put NA for non-applicable situation.

1. Are you willing to pay extra for a library if it is fully computerized and eases finding of book, etc.
   Yes\No
   __________ (if Yes, how much extra are you willing to pay)
2. What you feel should be necessary for a book to be searched? (by topic, by title, by author,. . . .)
   ____
   ____
   ____
   ____
3. Are you keen on online registration instead of the normal paper one.Yes/No
4. How many titles do you feel should be issued to a single member?
5. What should be the maximum duration for the issue of certain book to a member? __________ Days.
6. Should there be a facility to reserve a book in advance? Yes/No
7. How often do you visit the library? Choose One.
   a) Daily
   b) once in two days
   c) weekly
   d) bi-weekly
   e) monthly
8. Should there be a facility to reserve a book on phone?
   Yes/No
9. Should magazines and cassettes be included in the library
   Yes/No
10. Do you recommend this library to your friends, relatives, or acquaintances?
   Yes/No (if yes, to how many you recommended and out of them how many actually became the members)
   Recommended: __________ Became Members: __________

Fig. 2 : Questionnaire for members of library

Questions

Very Short Questions :

1. What is Interview ?
2. What is Questionnaries ?
3. What is Record review ?
4. What is Qusite Observation ?

Short Questions :

1. Describe types of Interview
2. Describe types of Questionnaries.
3. What is difference between Interview and Questionnaries ?
4. What is On-Site Observation ?

Long Question :

1. What are the two types of interviews used in fact finding technique ?
2. How many types of questionnaires are used ? Compare them.
3. What is fact finding ?
4. List different types of questionnaires formats
5. What are the different between open and closed form questionnaires ? For what types of information is each form most useful ?
6. What type information is often best obtained through interview discus in details.
7. Summarize the advantages and limitations of interviews and questionnaires.
8. Explain the differences between (I) Structured and unstructured interviewing and (II) Open ended and closed questions. Give an example of each.