# Analysis and Design of Information Systems

DCAP205/DCAP409

LOVELY PROFESSIONAL UNIVERSITY

LOVELY
PROFESSIONAL
UNIVERSITY

# ANALYSIS & DESIGN
# OF INFORMATION SYSTEMS

Produced & Printed by
**EXCEL BOOKS PRIVATE LIMITED**
A-45, Naraina, Phase-I,
New Delhi-110028
for
Lovely Professional University
Phagwara

# SYLLABUS

## Analysis & Design of Information Systems

***Objectives:*** To enable the student to understand tools and techniques to develop information system of an organization. After studying this course student will be able to: analyze any system, identify system boundaries and interfaces, capture user requirements by applying different techniques, generate logical design of user requirements using various structured design tools, understand various quality assurance issues related to system.

| Sr. No. | Description |
|---|---|
| 1. | ***Introduction to Information Systems Development:*** System Analyst, System Analysis & Design, Categories of Information Systems, System Development Strategies, Implementation and Evaluation |
| 2. | Managing the application development portfolio-Information system Planning, Managing project review & Selection, Information Systems & User-groups Committee Methods |
| 3. | ***Analysis:*** Preliminary Investigation, Scope of Study, Conducting the investigation, Testing Project Feasibility, Handling infeasible projects |
| 4. | ***Tools for System Requirements:*** Requirement Determination, Activities, Types.<br>**Fact-finding techniques: Interview, Questionnaire, Record Review, Observation.** Tools for documenting Procedures and Decisions: Decision Trees, Decision Tables, Structured English |
| 5. | ***Structured Analysis Development Strategy:*** Features, Data Flow Tools. |
| 6. | ***Tools for Structured Design:*** Data Flow Diagrams, Data Dictionaries. |
| 7. | ***Application Prototypes:*** Purpose, Steps, Use, Tools. Prototype Example.<br>**Computer Aided System Tools:** Role, Categories, CASE Tools. |
| 8. | ***Analysis To Design transition:*** Objectives, Features. Element of Design: Output, Files, Database Interaction, Input, Control, Procedures, Program Specifications. |
| 9. | ***Design of Computer output:*** Objective, Needs, Types. Design Input and Control: Objectives Capturing input data, Input validation. |
| 10. | Design of Online dialogues & its interface, design of files & Use of Auxiliary storage devices<br>**Systems Engineering & Quality Assurance**: Design Objectives, Design of Software & Documentation. |

# CONTENTS

# Unit 1: Introduction to Information System Development

Notes

CONTENTS

*Contd....*

Notes

## Objectives

After studying this unit, you will be able to:

- Understand the concept of information systems development
- Recognize the function of system analyst
- Discuss the system analysis and design
- Understand the categories of information systems
- Discuss the system development strategies
- Understand the implementation and evaluation

## Introduction

Most computer-based Information Systems are conceived, designed, and implemented using some form of systematic development process or methodology. Most methodologies consist of a finite number of phases. In this unit, you will understand various phases of information system development. A Systems analyst is a person who is overall responsible for development of a software. The role of system analyst is discussed in the unit. Also you will understand the concept of system analysis and design, categories of information systems, system development strategies, and discuss implementation and evaluation.

## 1.1 Information System Development

The number of phases in the information system development may vary from methodology to methodology. Each phase gives an output that becomes input of next phase. Mainly the process comprises of:

1. Investigation
2. Feasibility Study
3. Analysis
4. Design
5. Implementation
6. Maintenance

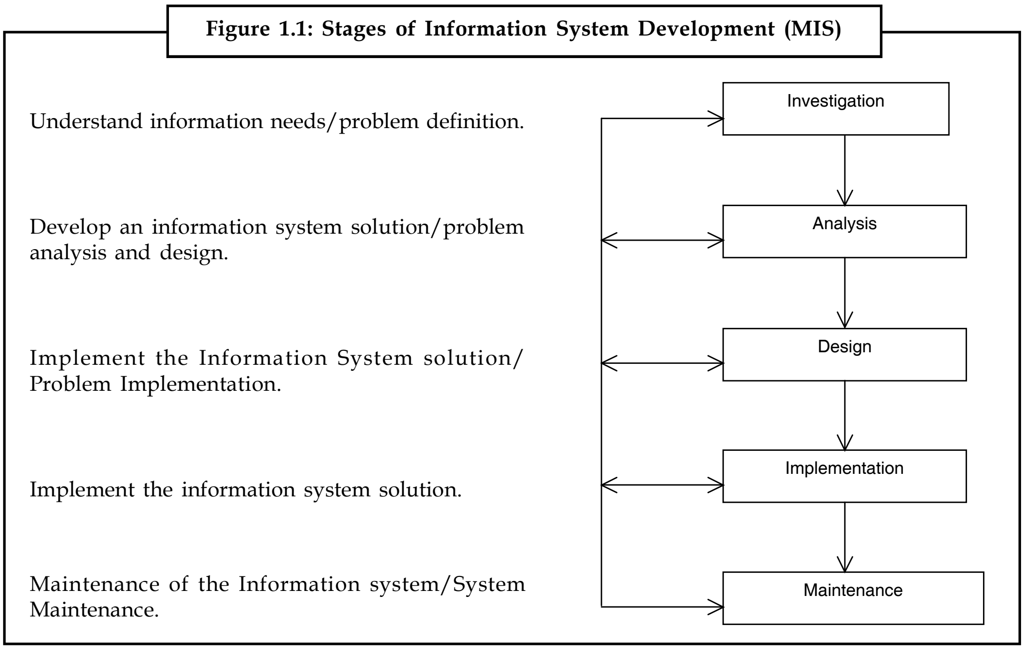
This can be represented by a flowchart shown in Figure 1.1.

Each stage of this development process is highly interrelated and interdependent on each other. The system developer has to perform many activities to accomplish each stage. Sometimes he has to go back to the previous stage to implement some changes so that a better system will be there or to produce a better solution to the organization problem or provide the best product to the users of Management Information System.

Now we take each stage individually and the activities related to each stage and how these activities are performed.

Notes

**Figure 1.1: Stages of Information System Development (MIS)**


Understand information needs/problem definition.
Develop an information system solution/problem analysis and design.
Implement the Information System solution/ Problem Implementation.
Implement the information system solution.
Maintenance of the Information system/System Maintenance.
Investigation
Analysis
Design
Implementation
Maintenance

## 1.1.1 Investigation

This is the first step in preparation of a Management Information System. This stage includes the preliminary study of proposed information system solution to the end users problems.

Because the development of Management Information System is a time consuming and costly effort, feasibility studies have to be conducted. This study is a preliminary study which investigates the information needs of the users and determines the resource requirement, costs, benefits and feasibility of a proposed project.

## 1.1.2 Feasibility Study

The goal of feasibility study is to evaluate an alternative system and to propose the most feasible and desirable systems for development. The feasibility study is conducted in four different areas:

1. Organizational feasibility
2. Economic feasibility
3. Technical feasibility
4. Operational feasibility.

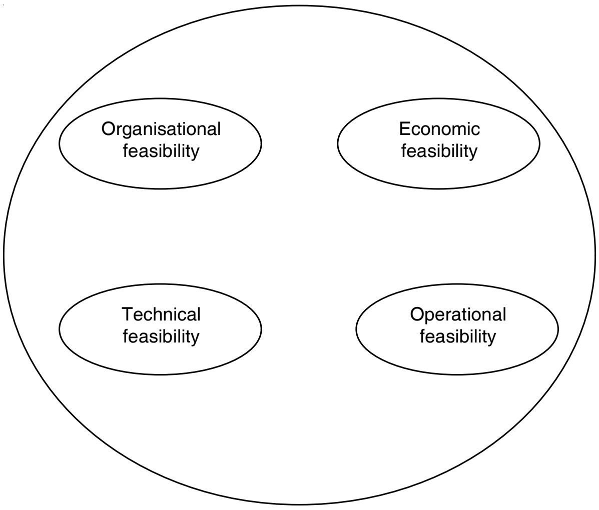
Figure 1.2 shows the various components of a feasibility study.

The focus of organizational feasibility is on how well a proposed system supports the values and objectives of the organization.

In economic feasibility, the developer conducts cost/benefit analysis, i.e., whether the cost of developing a system is more than benefit or loss because, if the development cost increases than the proposed benefit then the purpose of making Management Information System is defeated or it is not an efficient Management Information System.

This comprises of the study of the hardware and software requirements, availability of these media with in the organization or they have to be arranged. It also includes whether we have to develop our own system or buy some readymade solutions with some modification to meet our needs, etc.

**Figure 1.2: Components of Feasibility Study (MIS/D.K.)**



In this the willingness and ability of management, employees, customer suppliers and others to operate, use and support a proposed system is studied.

*Task* Make distinction between organizational feasibility and economic feasibility.

### 1.1.3 Analysis

This is the next stage of development process of Information System. It is applicable to both, i.e., for making an entirely new system or improving or replacing the existing one. Some activities of this stage are the next step of feasibility study but it is not a preliminary study like feasibility study. It is an in-depth study and results in the functional requirement hat provide the basis for system design. System analysis involves knowing the information needs of the end user and the organization employees, what are the resources activities and products presently available and the information system capability to meet these information needs.

The process of system analysis starts with analyzing the organization's present system if any, and functional requirement analysis.

To achieve effectiveness in Information System one must know something about the organization like, about its culture, management their experience, qualification, attitude towards computerized Information System, organizational values & norms/organizational working, etc. in detail, in general and specific to each organizational unit so that a balanced Information System comes into existence.

If the company has some information system in existence then it is very important to analyze that system. In this, one must analyze each component of system like hardware, software, people resource, networking and data resources. Whether the existing system serves the purpose of management or it is lacking at some end to serve the solution of information need, so that proper action should be taken at the time of designing the system so that these loop holes can be removed.

Functional requirements are end user information requirements that comprise of:

1. ***User interface requirements:*** In which one determines what type of input and output requirements of the user are there. It also includes source, formats, contents, etc., of each input & output media.

2. ***Processing requirements:*** What are decision rules, calculations are required to convert input into output. How much time it takes for processing the input into output?
3. ***Storage requirement:*** What is the size of data base, whether it is a common data base or distributed. What are the queries of the user?
4. ***Control requirements:*** What are the types of measures of accuracy, validity, safety, security and adaptability requirement for system input processing, output and storage function adopted.

## 1.1.4 Design

The previous stages depict what users demand from the system in order to fulfill their information needs and aims at answering "What is needed?" System design stage gives answer to the question "How" the system will accomplish the objective. Systems design consists of design activities that produce system specifications, satisfying the functional requirements developed at the analysis stage. System design consists of two steps – conceptual design and the a detailed design of the information system.

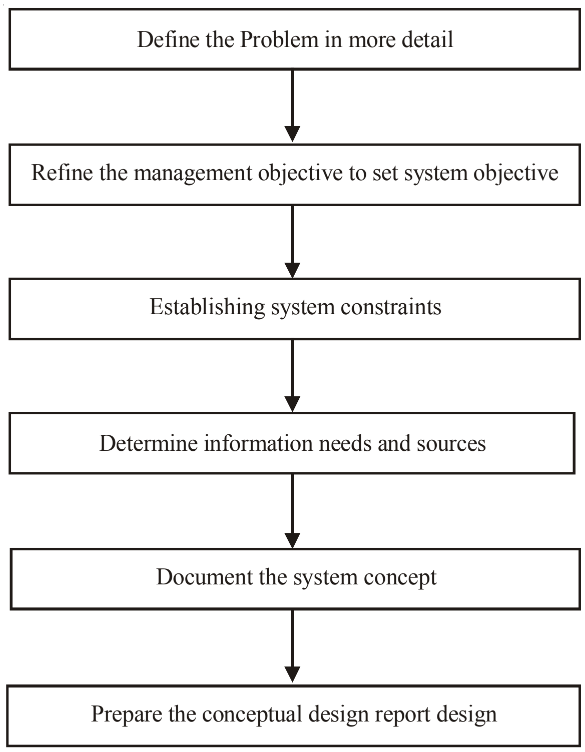
### Conceptual Design

Conceptual design represents the structure of Information System. The input to this stage is information requirement and management objective and the output is the performance requirement of those who will develop the detailed design.

The process of conceptual design involves showing the feasibility of meeting the management objective for Information System. This is the phase of system development which gives answer to the question "how" the system will work at gross or high level.

Figure 1.3 shows various activities at the time of design.

**Figure 1.3: Key Activities at the time of Conceptual Design**


Define the Problem in more detail
Refine the management objective to set system objective
Establishing system constraints
Determine information needs and sources
Document the system concept
Prepare the conceptual design report design

***Define the Problem***: With their dynamic nature, the problems must reside in an organization. What really or usually lacks is their proper definition and priority to solve them. Therefore, the first step in Management Information System design is to know the problem in detail. This is achieved by a continuous and iterative process of:

1. Stating the information need
2. Asking questions about that need
3. Suggesting the interpretations of that need
4. Detailing the original statement
5. Reviewing the more detailed statement of need with management.

This process will be repeated until we really understand the information needs and problems to be solved.

***Set System Objectives***: Like other functional areas, the manager should set objectives in terms of design of information system. Various organizations aiming for information of the records or the processing of the dates, over look the main objective of the information system, i.e., proper management of the information resources.

Although it is a difficult task to set objectives, because most of the organizations set their objectives very vaguely like "improve efficiency;" meet the production schedule" and so on. In order to achieve the objectives successfully they should be specific.

Information system objectives must ultimately be stated in terms of the objectives of the department, group, or in terms of the functions the information system is to perform. Information system objectives should be expressed in terms of what managers can do after their information needs have been full-filled. After setting the objectives, system designer should state them by using descriptive statement, or flowchart, or data flow diagrams and so on, to convert the objectives for the manager which they want to accomplish from the emerging information System.

The objective should be expressed not quantitatively rather qualitatively so that, alternative information system design in particular, and system performance in general can be measured, for information system's effectiveness and efficiency. To conclude, the information system objectives should be aligned with the overall objectives of the organization.

***Establish System Constraints***: Constraints mean problems, boundaries or restriction that enable the designer to stipulate the condition under which objective may be attained and to consider the limitation that restricts the design.

***Determine Information Need***: A clear statement of information need is fundamental and necessary for a good system design. If the manager does not convey his or her information needs, a good design will not be prepared. The type of information needs depends on two factors – one is personal managerial attributes of the individual manager and the organizations environment in which the decisions are made.

***Determine Information Sources***: Determination of information need and to determine the information source i.e., from where to collect the information are overlapping.

For designing a new system, the existing system needs to be analyzed. There are two approaches. One is in favor of this and other says there is no need of analysing the existing system. But sooner or later in a system design, the analysis of existing system becomes a necessary requirement. So the source of information is such which fits in the overall integrative sources of information and techniques of analysis.

There are two techniques for analysis and synthesis for discovering the information source, one is input/output analysis using decision table and other is multi dimensional flow using flow chart. So it is the next design step to prepare a list that matches needs and sources.

***Develop alternative conceptual designs and select one***: Every problem can be solved in a number of ways. So before selecting the most feasible alternative, one has to evaluate each alternative in the light of:

1. Compare anticipated performance of the conceptual design with the objectives of the system.
2. Prepare a rough or preliminary cost-effectiveness analysis of the information system.
3. Examine the flow chart for strength and weakness of each conceptual design.
4. Expand the conceptual design in more details, if none of these provides a preferred design.

***Document the System Concept***: In this stage manager participation to the design process shows what input, output, master files and rules for processing are required. The general system flow chart is a common method of indicating the general structure of a computer based information system. System flow also reflects the design efforts before and after this stage. At this stage the formats of input are designed. The input received from outside sources is then converted into machine usable forms. The output data definition includes the specifications of the destination like where they go and in what form, etc. Including the specification in what percentage one gets the output and at what frequency and form it will take, i.e., hard copy, soft copy, etc. After preparing the output and after input has been documented, the last step is to prepare a report for the manager.

***Prepare the Conceptual Design Report***: The conceptual design report, is in a sense a proposal for the expenditure of funds and for organizational change. In nutshell, it consists of performance specification, function to be performed by the system and means by which each function is measured. Along with this report separate documentation should be provided.

## Detailed Design

In order to make a detailed design, first of all the system designers have to gain the support of all the staff members from top to bottom level. To seek their acceptance it is better to involve them in the designing process.

The designer uses four sources for the design of the Information System. They are task force meeting; for a larger system an interview with the top, intermediate level manager and a selected group of operating staff; study the internal and external source document; and at last personal observation of operations and communications.

The detailed design is done for the areas of designing user interface, data design and process design.

1. ***User Interface Design:*** The user interface design activity is related to facilitate the interaction between the user and their computer based application. This involves:
   - On which operating system the Management Information System should be based.
   - Concentrate on the design of attractive and efficient forms of user input and output.
   - Design methods of converting source document into object document means converting human understandable form to machine understandable language.

   This activity produces the detailed design specifications for information product like display screen, forms, documents and reports.
2. ***Data design:***
   - It includes the design of the structure of databases i.e. what type of specific data element is carried.

Notes

- Entities and their characteristics.
- The relationships these entities have with each other.
- The integrity rules, i.e., how data is maintained and used in the information system.

3. ***Process Design:*** It involves the design of process i.e., the design of software programs, procedure needed by information system. At this stage the developer has to decide about the detailed specification of software, that is either software has to be purchased, or developed or it is purchased and modified according to the proposed system requirements.

   The product of this stage is to know system specifications means, the details of a proposed system. So the system analysts along with management and information system user can use their expertise to design a new or improved Information System.

   Finally, system design should be such, which specifies what types of hardware, software, data, people and network resources are required to make a better Information System. It also specifies how such resources will convert data resources into final information product which is needed or expected by the end user from the Information System.

*Caution* Before implementation and after designing the proposed system, the system must be tested because this is also one of the major activities of a detailed design.

There are three ways to get system feedback on the viability of designed Information System:

(a) Modeling

(b) Simulation

(c) Test planning.

The importance of testing at the designing stage is much because any fault or problem discovered at this stage will be easily removable, less costly and will reduce overall project cost.

*Task* Make distinction between data design and process design.

### 1.1.5 Cross Life-cycle Activities

Till now we have discussed the basic stages of Information System life cycle development approach. Actually information system development also involves a number of cross life cycle activities means, any activity that overlaps many or all stages of the information system development process.

*Example:* Documentation, presentation, feasibility analysis, fact finding, project and process management.

Fact finding activity relates to collect information by means of interview, personal observations, questionnaire and other technique regarding the problems, opportunities, requirements and their priorities. This activity can be required at many stages during the completion of the whole information system development process. At an early stage, i.e., investigation stage, it is related to learn about a business's problem, opportunities, requirements, constraints, objectives and their priorities. Fact finding is also required for design analysis, detailed decision final design implementation and maintenance stages.

But the required fact finding activity during these stages is required only to a lesser extent. Actually during these stages, with the help of fact fending activity the information system stakeholder collects feed back on technical design, standards used and so on.

Documentation is a continuous process of recording each fact finding and specification for correct and future references. Presentation is also a continuous activity of communicating either verbally or in writing the findings and recommendations for review by the information system user and information system owners.

Both of these activities are required at almost all stages of information system development project. In other words, these activities have a wider scope through-out the development process. Both are essential to each stage because poor communication will cause delay and cost over runs.

Feasibility analysis is a good estimation technique which estimates cost, time and other related resources required to develop information system. This is also a cross life cycle activity. Feasibility analysis as we know comprises of economic, technical, organizational and operational feasibility and are applicable to different stages of the information system life cycle development approaches. Process and project management is one of the principles for developing effective info system and is also a cross cycle activity. Because development of information system is a project so we have to adopt some process for successful completion of information system development project.

There are a large number of unsuccessful information systems. The main reason for this is that most of system analysts do not have managerial skill, are unfamiliar with how to properly employ tool technology to system development projects. Because of this lacuna there are cost overruns and a system with unfulfilled or unidentified information needs. So in order to have successful information system, an analyst must have managerial and good communication skills as well as he should be well versed in each stage and cross life cycle activities.

The System Development Life Cycle (SDLC) is the traditional system development method used by most organizations today. The SDLC is a structured frame that consists of sequential processes by which information systems are developed. These include investigation, system analysis, system design, programming testing, implementation, operation and maintenance. These processes, in turn, consist of well-defined tasks. Some of these tasks are present in most projects, whereas others are present in only certain types of projects. That is, large projects typically require all the tasks, whereas, smaller development projects may require only a subset of the tasks.

In the past, developers used the 'Waterfall approach' to the SDLC in which tasks in one stage were completed before the work proceeded to the next stage Today, system developers go back and forth among the stages as necessary.

System development for business applications is not an easy task. In developing a large integrated system such as MIS, many people are involved and many months or even years are spent. However, a small independent application such as Payroll can be developed in few weeks or months by a single or few programmers. For such small systems, system development activities may be done implicitly without proper documentation. But, for large systems, these activities must be done explicitly with proper planning and documentation. Whether a system is small or large, system development revolves around a life cycle that begins with the recognition of users' needs and understanding their problems. Such a life cycle comprising various phases is called System Development Life Cycle (SDLC).

## Self Assessment

Fill in the blanks:

1. The focus of ................................ feasibility is on how well a proposed system supports the values and objectives of the organization.

Notes

2. The process of ................................ starts with analyzing the organization's present system if any, and functional requirement analysis.
3. The process of ................................ design involves showing the feasibility of meeting the management objective for Information System.
4. The ................................ activity is related to facilitate the interaction between the user and their computer based application.

## 1.2 System Analyst

The system analyst is overall responsible for the development of a software. He is the crucial interface between users, programmers and MIS managers. He conducts a system's study, identifies activities and objectives and determines a procedure to achieve the objective. He has a very important role in the development of a system. The concerned person should also have some special qualities which we are going to discuss in this unit.

A Systems analyst is a person who is overall responsible for development of a software. He is the computer professional charged with analyzing, designing and implementing computer-based information systems. He is the crucial interface among users, programmers and MIS managers. A Systems analyst can be defined as follows:

A Systems analyst is a computer specialist who translates business problems and requirements into information systems and acts as liaison between IS (Information Systems) department and rest of the organization.

The analyst conducts a systems study, identifies activities and objectives and determines a procedure to achieve the objectives. He is the key member of both MIS organization and the software project team. He is a person with unique skills, experience, personality and common sense. His role has been emerging with advances in technology.

### 1.2.1 Roles of a Systems Analyst

The Systems analyst performs the following roles during various phases of SDLC. He works as a:

(a) ***Problem Investigator:*** The analyst studies the problems and needs of an organization during feasibility and requirements analysis phases of SDLC. He visits the various departments of the organization and interviews the users. He analyses the problems of the current system and collects their new requirements. The analyst initially works as an investigator by extracting the real problems of the users.

(b) ***Problem Solver:*** The analyst solves the problems of the current system faced by the users. He determines how people, method and technology can improve the current system. After feasibility analysis, he presents the system proposal to the management.

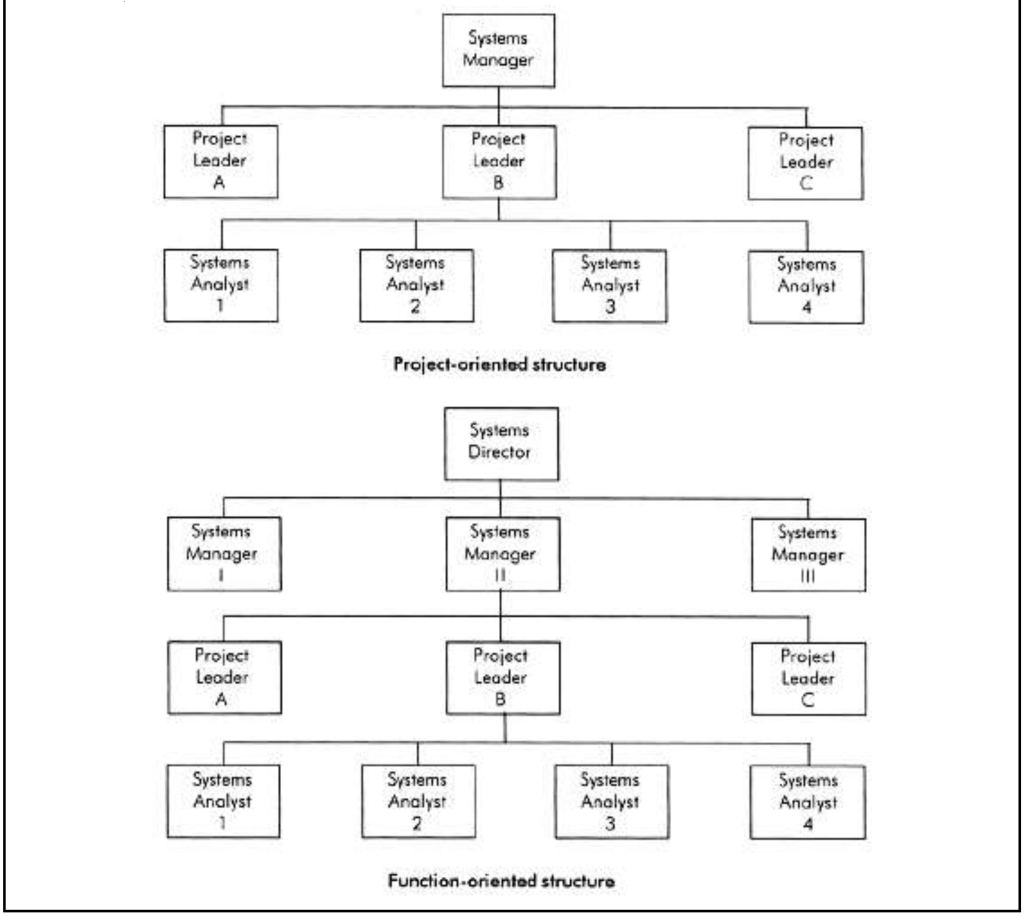
(c) ***Systems Designer:*** The analyst creates a detailed physical (current) and logical (proposed) design of the system.

(d) ***Motivator:*** The analyst motivates users to participate in development and implementation of the proposed system. This helps to understands user's feelings about the proposed system. The analyst interprets the thoughts of users and hence, draws conclusions. He appeals management and users for getting the support in development and implementation of the proposed system.

(e) ***Project Manager:*** The analyst monitors the development and implementation of software in relation to quality, cost and time. He works with the project leader for managing the project properly. For development of small systems, the Systems analyst is generally the project leader.

Notes

Figure 1.4: Two Types of Project Team Structures



Systems Manager

## 1.2.2 Qualities of Systems Analyst

Success in systems analysis requires interpersonal and technical skills of the analyst. The systems analyst is expected to possess the following qualities:

(a) ***Qualified:*** The analyst must be highly qualified in software technology. Besides software, he should have a good knowledge of hardware and latest communication and networking technology. He must have a thorough awareness about the working (manual and computerized) of financial accounting, sales and marketing, invoicing, inventory control, production and other information systems of different organization.

(b) ***Analytical Thinker:*** The analyst must be capable to extract real problems of the users by analyzing the existing system. He is expected to provide the best solutions to the problems. He should be able to provide more than one solution to a single problem so that the users can select the best one. The systems analyst must be capable of tackling any problem of the user. He must be a problem solver and not a problem creator.

(c) ***Good Communicator:*** The analyst must have a good communication and presentation skills. He must have an excellent command on the language which the user can understand. There should not be any communication gap between the systems analyst and users.

(d) ***Experienced:*** The analyst should be experienced in both information and management technologies. He should be associated with all types of business concerns (viz., Manufacturing, Trading, Financial, etc.). The present day systems analysts are expected to possess a good experience in development of software using 4GLS (such as Oracle, Sybase, etc.) and object-oriented languages (such as C++).

(e) ***Creator:*** The analyst should possess excellent creativity skills that help to convert ideas of the users into concrete plans. He/she should be capable of creating plans and designing systems by drawing diagrams, charts and other illustrations.

(f) ***Trainer:*** The analyst should be a good teacher for educating and training users in computer-based information systems.

### Self Assessment

Fill in the blanks:

5. ................................ is the crucial interface between users, programmers and MIS managers.

6. ................................ analyses the problems of the current system and collects their new requirements.

## 1.3 System Analysis & Design

The study of "Systems" is by no means a new or even recent endeavour. Systems have been in use for the last thousands of years. The Egyptians used bookkeeping system over 5000 years ago for keeping their accounts, while Phoenician astronomers tried to study the systems of stars for making predictions.

Systems today are very helpful in running the business efficiently. But a system can function in an effective way only if the users such as the accountants, business managers and other responsible individuals within the company make it function in a proper way. Many times, managers are told that they only need to know how to retrieve required information, thus making them ignorant of the operations of the system as a whole. In accepting such advice, these managers are, in essence, relinquishing a substantial part of the control of the organization to the system's designers. It is, therefore, necessary that these potential systems analysts should clearly understand many other things also such as what a system is, what its objectives are, what kinds of systems there are, what goes with their creation and maintenance, what are their costs and benefits and how to analyze and monitor systems.

All organizations need systems for processing of their routine transactions. An information system is a system that provides information to management and other people in an organization. As computers are becoming part of every activity in most of the organizations, many information systems now use computer systems for manipulating information. Systems analysis is the application of the systems approach to develop Computer-Based Information Systems (CBIS) or a computer-based MIS (Management Information System). Before studying the concepts of systems analysis, we must understand the basic systems concepts.

System analysis and design refers to the process of examining a business situation with the intent of improving it through better procedures and methods. Systems development can generally be thought of as having two major components: System Analysis and System Design. System design is the process of planning a new system or replace or complement an existing system. But before this planning can be done, we must thoroughly understand the existing system and determine how computers can best be used to make its operation more effective. System analysis, is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvement to the system. In brief, we can say that analysis specifies what the system should do. Design states how to accomplish the objective.

### 1.3.1 System

The term system is derived from the Greek word systems, which means an organized relationship among functioning unit's or components. A system exists because it is designed to achieve one or more objectives.

A system is a set of interrelated elements that collectively work together to achieve some goal. For instance, accounting is a system with elements, viz., journals, ledgers, people, etc. and its basic goal is to maintain book of accounts along with preparation of financial and MIS statements. Computer is also a system with elements such as CPU (Central Processing Unit), input device, output device and users; and its basic goal is to process the data and provide information. There are hundreds of definitions of the word 'System', but here we define it as follows:

*A system is a set of interrelated elements that form an activity or a processing procedure in order to achieve a common goal or goals by operating on data to yield information.*

### 1.3.2 Subsystems

Most systems are part of a larger system. For instance, Financial Accounting System, Marketing System, and HRD (Human Resource Development) System are parts of a larger system, MIS (Management Information System) and are called subsystems. A system can be made up of many subsystems. A subsystem is defined as follows:

*A subsystem is that part of a system that carries one part of the system function.*

### 1.3.3 System Study

Systems study may be defined as "a study of the operations of a set of connected elements and of the interconnections between these elements". It shows clearly that one cannot ignore any part or element of a system without first finding out the effect that element has on the operation of the system as a whole. We can understand this with the help of systems analysis.

### 1.3.4 System Approach

The information systems (such as MIS) are designed on the basis of synergy of subsystems (such as Production, Inventory, Sales and Marketing systems) in order to achieve a net unified cohesive system.

The approach in developing information systems involves focus on the design of a whole integrated system rather than on independent subsystems in order to optimize the net results of the operations of an organization. This is called the systems approach.

*Example:* An invoicing system, an inventory control system and a financial accounting system can be designed independently. However, the net results of the operations of an integrated whole system are more than that of independent subsystems.

### 1.3.5 Difference between System Approach and System Analysis

There is a difference between "systems approach" and "systems analysis" also. The systems approach shows a set of procedure for solving a particular problem. It applies scientific methods to observe, clarify, identify and solve a problem with special care being taken to understand the inter-relatedness between elements and their system characteristics. However, systems analysis is a management technique which helps us in designing a new system or improving an existing system.

### 1.3.6 System Characteristics

A system has the following characteristics:

1. ***Organization:*** Organization implies structure and order. It is the arrangement of components that helps to achieve objectives. The various elements of a system are organized to achieve objectives. For instance, input devices, output devices and the CPU of a computer system are organized to process the data and produce information.
2. ***Interaction:*** Interaction refers to the procedure in which each component functions with other components of the system. The various elements of a system are interacted with others to achieve a common goal. For instance, the ledger, journals and people are interacted in a financial accounting system for preparing the final financial statements (e.g., Profit and Loss A/c, Balance Sheet, etc.) of an organization.
3. ***Interdependence:*** Interdependence means that components of the organization or computer system depend on one another. The various subsystems of a system depend on one another for sharing of input data. For instance, in a computerised MIS (a system), the financial accounting system (a subsystem) receives the input data (e.g., financial data from Invoices, cash memo etc.) from the invoicing system (a subsystem).
4. ***Integration:*** Integration is concerned with how a system is tied together. It is more than sharing a physical part or location. It means that parts of the system work together within the system even though each part performs a unique function. Successful integration will typically produce a better result as a whole rather than if each component works independently.
5. ***Central objective:*** Central objective is the last characteristic of a system. Objectives may be real or stated. Although a stated objective may be the real objective, it is quite common that organization may set one objective and operate to achieve another. The important point is that users must be aware about the central objective well in advance.

### Self Assessment

Fill in the blanks:

7. ................................ is the process of planning a new system or replace or complement an existing system.
8. The systems ................................ shows a set of procedure for solving a particular problem.
9. A ................................ is that part of a system that carries one part of the system function.

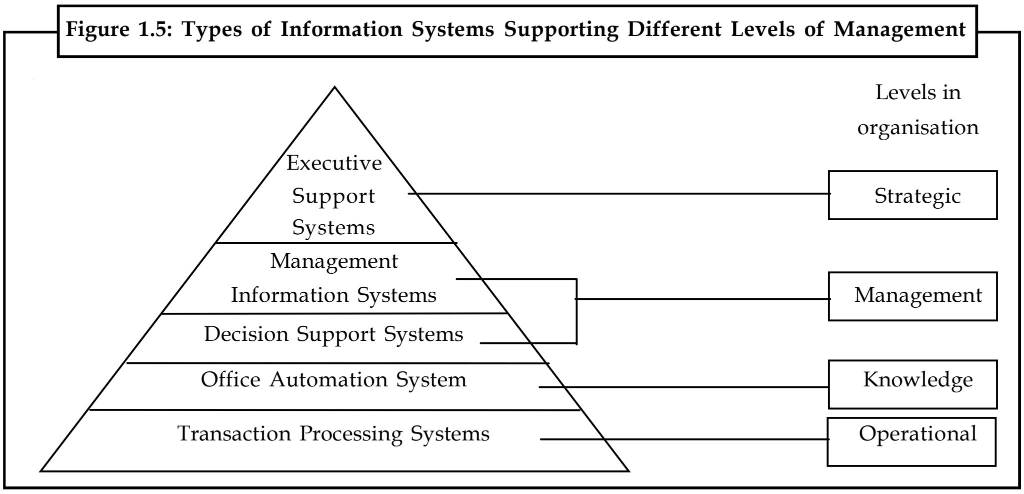
## 1.4 Categories of Information System

There are five major types of information systems for various management levels of an organization which are illustrated in Figure 1.5 and are discussed below:

### 1.4.1 Executive Support System (ESS)

This system is designed to address unstructured decision-making at the strategic level of an organization. The systems at strategic level help senior managers in long-term planning. ESS employ advanced graphics and communications software for creating a generalized computing and communications environment.

Figure 1.5: Types of Information Systems Supporting Different Levels of Management



### 1.4.2 Management Information System (MIS)

This system is designed to serve the functions of planning, controlling and decision-making at the management level of an organization. The system at management level support monitoring, controlling and decision-making activities of middle level managers.

While computers were ideal for routine transaction processing, managers soon realized that the computers' capability of performing rapid calculations and data comparisons could produce meaningful information for management. Management information systems thus evolved out of transaction processing systems. A Management Information System, or MIS (pronounced em-eye-ess), is an information system that generates accurate, timely and organized information so managers and other users can make decisions, solve problems, supervise activities, and track progress. Because it generates reports on a regular basis, a management information system sometimes is called a Management Reporting System (MRS).

Management information systems often are integrated with transaction processing systems.

*Example:* To process a sales order, the transaction processing system records the sale, updates the customer's account balance, and makes a deduction from inventory. Using this information, the related management information system can produce reports that recap daily sales activities; list customers with past due account balances; graph slow or fast selling products; and highlight inventory items that need reordering.

An MIS generates three basic types of information: detailed, summary and exception. Detailed information typically confirms transaction processing activities. A Detailed Order Report is an instance of a detail report. Summary information consolidates data into a format that an individual can review quickly and easily. To help synopsize information, a summary report typically contains totals, tables, or graphs. An Inventory Summary Report is an instance of a summary report.

Exception information filters data to report information that is outside of a normal condition. These conditions, called the exception criteria, define the range of what is considered normal activity or status.

*Example:* An example of an exception report is an Inventory Exception Report that notifies the purchasing department of items it needs to reorder.

Exception reports help managers save time because they do not have to search through a detailed report for exceptions. Instead, an exception report brings exceptions to the manager's attention in an easily identifiable form. Exception reports thus help them focus on situations that require immediate decisions or actions.

*Did u know?* A management information system focuses on generating information that management and other users need to perform their jobs.

### 1.4.3 Decision Support System (DSS)

This system also serves the information needs at management level of an organization. DSS differ from MIS in mainly having more analytical power and more user-friendly capabilities. DSS combine data and analytical/modeling tools to support semi-structured/unstructured decision- making.

Transaction processing and management information systems provide information on a regular basis. Frequently, however, users need information not provided in these reports to help them make decisions.

*Example:* A sales manager might need to determine how high to set yearly sales quotas based on increased sales and lowered product costs. Decision support systems help provide information to support such decisions.

A Decision Support System (DSS) is an information system designed to help users reach a decision when a decision-making situation arises. A variety of DSSs exist to help with a range of decisions.

A decision support system uses data from internal and/or external sources. Internal sources of data might include sales, manufacturing, inventory, or financial data from an organization's database. Data from external sources could include interest rates, population trends, and costs of new housing construction or raw material pricing. Users of a DSS, often managers, can manipulate the data used in the DSS to help with decisions.

Some decision support systems include query language, statistical analysis capabilities, spreadsheets, and graphics that help you extract data and evaluate the results. Some decision support systems also include capabilities that allow you to create a model of the factors affecting a decision.

*Example:* A simple model for determining the best product price would include factors for the expected sales volume at each price level.

With the model, you can ask what-if questions by changing one or more of the factors and viewing the projected results. Many people use application software packages to perform DSS functions.

*Example:* Using spreadsheet software, you can complete simple modeling tasks or what-if scenarios.

A special type of DSS, called an Executive Information System (EIS), is designed to support the information needs of executive management. Information in an EIS is presented in charts and tables that show trends, ratios, and other managerial statistics. Because executives usually focus on strategic issues, EISs rely on external data sources such as the Dow Jones News/Retrieval service or the Internet. These external data sources can provide current information on interest rates, commodity prices, and other leading economic indicators.

To store all the necessary decision-making data, DSSs or EISs often use extremely large databases, called data warehouses. A data warehouse stores and manages the data required to analyze historical and current business circumstances.

Notes

Caselet

## NSE.IT New Tool to Help in Real-time Trading

NSE.IT is all set to provide a new software tool — "decision support system," (DSS) — for the capital market and derivatives segments.

The software, currently in the developmental stage, is being touted as an advanced tool for real-time technical analysis and decision-making. It is being targeted towards investors and traders and aims at giving them the power to strategise their plans.

Commenting on the product, Mr Satish Naralkar, CEO, NSE.IT, told *Business Line* that the software was expected to be operational by the end of the first quarter. "We are still working out our implementation plan," he said.

The company had in November 2001, launched for the first time in the country an integrated single screen CTCL product, NEATxS, for the capital market and derivative segments. "The work on developing the DSS was started simultaneously," Mr Naralkar added.

The product will provide user definable query parameters on real-time data that help identify opportunities quickly. It will also enable query on volume, price, quantity, candlestick patterns etc.

Other features include events and alerts, charting, fundamental analysis, 'what if,' analysis, mathematical calculations and models for derivatives etc.

The DSS product will support interfaces to multiple data feed sources for the latest market data. The sources would include ASP-based data delivery model run by NSE.IT and CTCL (computer to computer link) products.

NSE.IT has also introduced a learning management system under the aegis of finvarsity, its e-learning education portal for financial markets.

"To be christened the 'Enlitor', the system will serve as an on-line tutor. Finvarsity is an environment powered by the enlitor," Mr Naralkar said.

While finvarsity has been designed to address the training needs of trading members, investors, students and corporates, the target audience for enlitor is the entire corporate world and educational sector.

Enlitor will enable educational institutions to deploy their classroom/distance education programmes on the Internet, LAN, broadband satellite etc. "Enlitor will integrate technologies such as text, audio, live motion, video graphics and streaming multi media technologies."

All services are customised to the specific requirements of the institution using the software engine. "While deploying the course content of the institution, emphasis would be laid on interactivity, multimedia presentations, animations, simulation quizzes that are needed for imparting quality education," Mr Naralkar explained.

According to Mr Naralkar, Enlitor, will use multiple technologies to deliver courseware, depending upon the needs of the target population and the environment in which they are situated.

***Source:*** http://www.thehindubusinessline.in/bline/2002/01/14/stories/2002011400550200.htm

Notes

### 1.4.4 Office Automation System (OAS)

This system serves the knowledge level of an organization for supporting knowledge workers like production managers, EDP managers, etc. OAS use computer system to increase the productivity of technical managers in the office.

An Office Information System, or OIS (pronounced oh-eye-ess), is an information system that uses hardware, software and networks to enhance work flow and facilitate communications among employees. With an office information system, also described as office automation; employees perform tasks electronically using computers and other electronic devices, instead of manually.

*Example:* With an office information system, a registration department might post the class schedule on the Internet and e-mail students when the schedule is updated. In a manual system, the registration department would photocopy the schedule and mail it to each student's house.

An office information system supports a range of business office activities such as creating and distributing graphics and/or documents, sending messages, scheduling, and accounting. All levels of users from executive management to non management employees utilize and benefit from the features of an OIS.

The software an office information system uses to support these activities include word processing, spreadsheets, databases, presentation graphics, e-mail, Web browsers, Web page authoring, personal information management, and groupware. Office information systems use communications technology such as voice mail, facsimile (fax), videoconferencing, and Electronic Data Interchange (EDI) for the electronic exchange of text, graphics, audio, and video. An office information system also uses a variety of hardware, including computers equipped with modems, video cameras, speakers, and microphones; scanners; and fax machines.

### 1.4.5 Transaction Processing System (TPS)

This system is designed to serve the operational level of an organization. TPS record and process the daily routine transactions of the organization like, accounting, payroll, order processing, etc.

A Transaction Processing System (TPS) is an information system that captures and processes data generated during an organization's day-to-day transactions. A transaction is a business activity such as a deposit, payment, order or reservation.

Clerical staff typically performs the activities associated with transaction processing, which include the following:

Recording a business activity such as a student's registration, a customer's order, an employee's timecard or a client's payment. Confirming an action or triggering a response, such as printing a student's schedule, sending a thank-you note to a customer, generating an employee's paycheck or issuing a receipt to a client. Maintaining data, which involves adding new data, changing existing data, or removing unwanted data.

Transaction processing systems were among the first computerized systems developed to process business data – a function originally called data processing. Usually, the TPS computerized an existing manual system to allow for faster processing, reduced clerical costs and improved customer service.

The first transaction processing systems usually used batch processing. With batch processing, transaction data is collected over a period of time and all transactions are processed later, as a group. As computers became more powerful, system developers built online transaction processing systems. With Online Transaction Processing (OLTP) the computer processes transactions as they are entered. When you register for classes, your school probably uses OLTP.

The registration administrative assistant enters your desired schedule and the computer immediately prints your statement of classes. The invoices, however, often are printed using batch processing, meaning all student invoices are printed and mailed at a later date.

> *Notes* Today, most transaction processing systems use online transaction processing. Some routine processing tasks such as calculating paychecks or printing invoices, however, are performed more effectively on a batch basis. For these activities, many organizations still use batch processing techniques.

### 1.4.6 Stationary and Non-stationary System

The operations and properties of a stationary system do not change significantly while those of a non-stationary system change with time.

*Example:* A computerized MIS is a stationary system because once designed, the MIS handles problems and provides information on a routine basis without any significant changes.

*Example:* An organizational system that tends to adapt to a changing environment is an example of a non-stationary system.

### 1.4.7 Adaptive and Non-adaptive System

Adaptive System tend to adapt to a changing environment while non-adaptive systems do not adapt.

*Example:* An organizational system is an adaptive system and a MIS system is a non-adaptive system.

Stationary system is always non-adaptive while non-stationary systems are adaptive systems.

### Self Assessment

Fill in the blanks:

10. A ................................ is an information system that generates accurate, timely and organized information so managers and other users can make decisions, solve problems, supervise activities, and track progress.

11. A ................................ is an information system designed to help users reach a decision when a decision-making situation arises.

12. A ................................ is an information system that captures and processes data generated during an organization's day-to-day transactions.

## 1.5 System Development Strategies

After designing the input and output, the analyst begins developing the software using a programming language. This is the phase, when the programmers play their major role in development. They start designing the data structures and writing of programs as per the documents prepared during design phase. So, this phase can be categorized into two

Notes

sub-phases, i.e., database design and program design. Database design is the most important aspect of developing a new system. As data is the basic component or raw material of any information system, it is needed to be stored in an organized way. How data has to be organized, depends on the requirement specifications, hardware configurations and the features of programming language and DBMS used. What is DBMS and how database can be organized and managed?

Program design is mainly concerned with writing of programs (coding), editing of programs using a text editor or word processor, debugging and finally testing them. There is generally a team of programmers, who work under guidance of their project leader/systems analyst and do all the codings.

Two method-dependent strategies for systems development can be recognized. The first strategy depends on methodology; and the second on technique.

*Did u know?* Systems development is, basically, a problem-solving action. A problem in an application domain is malformed by the systems development procedure into a solution in the computer's implementation field.

### 1.5.1 Unified Methodology Approach

Methodologies are a formal effort to address intricacy via the use of standard, conventional strategies to systems development. Current methodologies are apt to concentrate on chiefly one unit of disintegration, but they fluctuate on what that unit of disintegration is. Most general methodologies base disintegration on either process or data, or some combination of the two.

The procedure approach to managing intricacy is seen, for instance, in the structured techniques—structured analysis, design, and coding. It is, possibly, the oldest and most broadly accessed methodology, and also is the one most often referenced in the information systems concept. The structured techniques all mainly utilize process decomposition, even though the seminal functions on structured analysis also incorporated normalization of data as a secondary concentration of the methodology.

The data approach to managing intricacy is observed in information engineering. It has its origins in the entity relationship strategy to modeling data. Information engineering originally employs data decomposition at the enterprise or organization stage to manage problem intricacy, the motive being that an enterprise's data is, in common, more steady than the processes accessed to proceed on that data. After the preliminary data analysis, systems projects are produced by means of process decomposition, which is, as a result, a secondary importance.

*Caution* The dissimilarity between process and data-oriented methodologies is one of preliminary emphasis. So, ultimately, both orientations must be measured.

The object-oriented strategy to managing intricacy considers both data and procedure as a package. An object is a constituent of the problem's world, a consistent compilation of data coupled with the procedures (methods or functions) acting on that data. The function of systems development by means of the object-oriented approach interleaves analysis and design of objects with analysis and design of the processes concerning to those objects. The foundation for the object-oriented strategy is that application problems frequently develop around real-world objects and the manners in which they interrelate.

Whether the systems developer utilizes a process, data, or object-oriented strategy, the methodology concerning the strategy will be consistent and organized. Such a methodology is termed here as "unified."

### 1.5.2 Technique Approach

Methodologies are, certainly, compilations of connected techniques. Disaggregating most methodologies consequences in recognizes techniques of unstable utility—some techniques are exceptionally important, some are comparatively valuable, and some have only minor value.

*Example:* A previous constituent of the structured techniques was the concept of a "Chief Programmer Team." Over the years, as it became obvious that this was not a victorious part of the structured techniques, that concept was removed.

Due to this inconsistency in the value of constituent methods, there are those who utilize collections of suitable techniques instead of unified methodologies.

With this approach, systems developers are taught in the utilization of "best of practice" techniques recognized to have been victorious in resolving an enterprise's troubles.

### Self Assessment

Fill in the blanks:

13. Program ................................ is mainly concerned with writing of programs (coding), editing of programs using a text editor or word processor, debugging and finally testing them.

14. The foundation for the ................................ strategy is that application problems frequently develop around real-world objects and the manners in which they interrelate.

## 1.6 Implementation

Implementation means to introduce the designed system into practice or in use. The implementation process covers the following:

1. Acquisition of hardware and software resources required by the proposed system.
2. Develop the computer program or perform any modification in the existing programs or the software package purchased.
3. Train the end user - it involves (a) preparing training program and documents that explain how to operate the proposed system. i.e. to make manual of the Information System. (b) Educate and train managers, sales person, computer operators who operate the system.
4. Test the system and remove errors if any. This process is continued till system is free from errors.
5. Conversion process i.e. to introduce a new system.

The conversion process has many ways to convert from old system to new system.

A. ***Parallel:*** When new and old systems are run in parallel for a trial period and a comparison of both is done. If the proposed system gives a satisfactory solution to information need, it is accepted and the old one becomes obsolete.

B. ***Pilot:*** In this the new system is introduced at one location or site only for trial. If its performance is according to the need, it is introduced in whole of the company or organization.

C. ***Phasing:*** Introduce the system in phases i.e. the new system is introduce at one site at a time. This method is useful when upgrading of old system is done.

D. ***Plunge:*** It is also known as an immediate cut over or change-over. Introduce the new system as and when it is ready to work and remove the old one directly.

Except for the timing and for obvious variations, the implementation steps for all four methods may be covered together.

> *Notes* It should be pointed out that occasionally design and implementation are carried on simultaneously. Such a process provides operational testing of the design on a continuous basis, but it limits consideration of major design alternatives. It is a trial-and-error process. Completion of conceptual and analytical design in advance of equipment installation offers many advantages besides cost.

So these are the four basic methods of implementing Information System in an organization, after the completion of the design stage.

### Self Assessment

Fill in the blanks:

15. ................................ means to introduce the designed system into practice or in use.

16. ................................ is also known as an immediate cut over or change-over.

## 1.7 Evaluation and Maintenance

After introducing the system for some time usually after a month, the system developer takes feed back from the manager, sales person, operators and users of the system that whether the system is achieving its objective or not. This process is known as evaluation.

System maintenance is the last or concluding stage of Information System development process. But it's importance is not less being the last stage because an effective systems can fail if they are not maintained properly.

Maintenance involves control, evaluation and modification to make a better system. Maintenance is required, because sometimes operators develop their own private procedure, or make some short cuts, or some unauthorized person introduces some changes in the present system without taking permission. Maintenance activity is initiated by error reports, a user change request, a member of maintenance team, or by the management.

*Proper planning is done for maintenance.* It involves:

a. Collect all requests for change.

b. Give priority to each request after analysing their long run benefit and cost effects.

c. Prepare short plans.

d. Document the maintenance as it occurs.

*Again review the Information System design manual.*

But sometimes there are certain problems or barriers in performing maintenance. They are related to, when there is no proper plan for maintenance; resources are not allocated for this purpose, lack of qualified staff, lack of management and user's support and interest for maintenance.

*Maintenance activity is related to make some modification i.e.,*

1. To change the policy statement
2. To change forms
3. To change operating system
4. To change procedures, etc.

## Self Assessment

Fill in the blanks:

17. After introducing the system for some time usually after a month, the ................................ takes feed back from the manager, sales person, operators and users of the system that whether the system is achieving its objective or not.
18. ................................ involves control, evaluation and modification to make a better system.

*Case Study*

### A Nontraditional Systems Analysis and Design

The termed software crisis was coined in 1968 by the NATO Software Engineering Conference for the myriad of problems in the development of quality software. The field of Software Engineering grew as a response to those problems and system analysis and design were recognized as important components of quality software. Since 1968, hardware costs have dramatically dropped and many software problems can be addressed through the use of application utilities - word processors, spreadsheets, data bases, etc.

With the hardware and software in the hands of the users, program solutions can be created by the users, and thus, analysis and design may be done by "anyone."

We present a case study of the design of a system by an end user. The user was technology literate, but had no formal training on systems analysis and design. A comparison is drawn between this approach and classic system analysis and design. The benefits and problems with this modern approach are also considered.

This modern approach has the potential to provide systems solutions to many problems, but could lead to a different problem, a new software crisis – using modern hardware and software with ad hoc analysis and design. These system solutions are developed very quickly and cheaply, but many times without consideration for the users or proper data handling methods.

Producing reliable, robust, cost-effective software systems has always been a problem for the computing industry. In 1968 the term software crisis was coined by the NATO Software Engineering Conference for the myriad of problems in the development of quality software. In those days, computers were less common and more expensive; there were few programmers and analysts; every system was developed from "scratch." Even small projects took months to develop and were thus expensive undertakings.

*Contd.....*

Computer Science (CS) and Computer Information Systems (CIS) programs have responded to the problem in a number of ways. Early programming courses taught not only the particular language, but also how to develop quality programs; sometimes called program engineering. Both programs provided courses in Systems Analysis and Design and later in Software Engineering.

Students were taught a systematic approach to the development of computer systems.

They learned the lifecycle of a system and how to approach the analysis, design, and implementation to produce reliable and robust software. In many programs, students were expected to demonstrate their competence through a project courses before they graduated and joined an IT department and at the time, the only people developing systems were the IT departments.

However, computing has changed since 1968. Computers are now cheap and plentiful.

Modern programming languages have been developed, providing a number of improvements, most notably ease in developing a graphical user interface.

A useful development, for creating software systems, has been the creation of application utilities – word processors, spreadsheets, data bases, etc. Now many "programming" solutions can be developed primarily by customizing these packages.

Furthermore, the combination of inexpensive hardware and application utilities has allowed the end-user to develop their own unique software applications. Users no longer need to wait for an IT department or consultant. Those who are interested may develop their own. Development has moved from the hands of the CS/CIS, IT expert into the general population.

The following is a study of one particular system developed in this new style: a technology savvy user implementing a system with application utilities. The chosen problem is simple and straight forward; one that could be approached by a sophomore/junior CS/CIS major. Names and identities have not been used to ensure privacy.

**Case Study: Problem**

A local public school system provides special education services to preschool children by sending itinerant teachers to service these children at various preschool and day care facilities. These teachers must periodically (yearly) conduct "case conferences," which provide progress reports using multiple state and local mandated forms. These multi-copy forms are preprinted, filled out by hand by the teacher, and then distributed to the parents and the student's permanent file. The file is intended to follow the student as he/she enters the public school system.

The problem, one faced in many systems, is the massive paperwork. The itinerant teacher must fill out several forms for each student. These forms provide spaces for the teachers hand written information, however information is duplicated on the various forms and the process is time consuming. Current estimates are approximately one half hour per student with 60 or more students per teacher. In addition, the teachers must find time in their already busy schedule to complete the forms.

The teachers and administration felt this was an ideal situation for an improved system.

The goals of the system were to be (1) reduced time to prepare the forms for a case conference, (2) reduce re-entry of redundant information, and (3) provide more accurate records.

*Contd.....*

Notes

**Case Study: Solution**

The traditional approach would be to turn the problem over to the IT department or an outside firm to analyze, design, and implement and appropriate solution. This would take time and be costly, and school systems rarely have extra funds.

The special education department had their own technology specialist. This person was trained as a special education teacher, but demonstrated an interest and aptitude in technology and was moved into an administrative position supporting technology in the special education area. This person had an advantage over a normal IT specialist: as a member of the department they were already familiar with the forms and the procedures.

Therefore they initially had a better understanding of the problem than an IT designer.

However, there was no more real analysis. The itinerant teachers were not consulted either as a group or individually; i.e. there was no discussion with the end-users. The manual system was taken as a model and replaced with a computerized version.

The solution implemented was to create files that were word processing (Microsoft Word) templates for each form that must be completed. These files were distributed to the teachers, on a floppy or compact disk. The individual teacher then, knowing the forms needed, would selected the appropriate files for a particular student, filled in the appropriate fields in the template, and saved the updated copies of the files in a machine readable form, usually on floppy disk. The final step was the case conference where the itinerant teacher(s), and other teachers or specialists, met with the child's parents.

During the case conference, the teacher may need to add additional comments to the forms, so the files were again edited and then the appropriate number of hard copies was printed for signatures. The files were to be kept in machine readable form (floppy disk) for each student. Thus a computerized record was formed for each student to be carried on through the educational process.

**Analysis of Case Study**

The resulting system had several advantages over the previous manual system. The material was no longer hand written. This meant no chance of going back to a form at a later date and finding that information was unreadable. The material was kept in a machine readable form, so in theory, it could be reused in subsequent years.

A change in the design of the forms could be implemented quickly, with minimal cost.

The file containing the particular form could be edited, or replaced if there were major changes, and redistributed for the cost of a disk.

The system was developed quickly and cheaply. The only personnel involved were already part of the staff. The system was implemented using existing equipment and software.

Unfortunately, there are a number of defects in the resulting system. The teachers involved were itinerant and many were not provided their own portable computers or printers. They were expected to find a computer at the facility they were serving or use the desktop systems at their home office, which they visited only once or twice a week.

This was a major problem when, during a case conference, updates were needed to the forms. The teacher had to find a computer to make the changes, print the updates, and then return to the conference. Many times a second meeting had to be scheduled to sign the updated forms.

There was no training planned or provided for the end-users, the itinerant teachers. Many of the teachers had limited, or minimal, computer skills. They were expected to not only

*Contd.....*

understand the working of the word processor, but also expected to be able to load a file from one location device, edit, then save it on another device. Furthermore, since they were itinerant, they may have to use several different computer systems. There was no part of the system designed to handle backing up the critical data stored on an unreliable floppy disk. There was no plan for error recovery.

The new system had three goals; the first was to reduce the time to prepare the forms. In fact, it took longer to create the forms for an individual student! The estimated time increased from half an hour for the hand written forms to about an hour for the word processing system. Typing speed accounts for a portion of the delay, but the new system required opening multiple files and saving them on different media. While this is not a difficult task, it may be time consuming and stressful for a user who was uncomfortable with computers to begin with. Furthermore, there was no training to do this and no systematic approach to renaming and organizing the new files.

The second goal was to reduce the re-entry of redundant data. This problem was not even addressed. The computerized forms were identical to the paper forms and therefore required all the re-entry the manual system required. The teachers had to re-enter data - such as name, birth data, etc. – repeatedly on the separate forms. This presented the same problems that the manual hand written system had. Furthermore, in the manual system the teacher could lay out the forms and copy the redundant information from one form to the others, thus ensuring some consistency. In the computerized system, teacher worked on one form at a time and except for the most technologically savvy, could not compare the redundant data for consistency.

The third goal was more accurate records. It does not appear that the computerized system provided any advantage over the manual system, although it appears to be no worse than the manual system.

As a final observation: the newer system should have taken advantage of the potential to reduce data entry errors. The new system by default took advantage of the word processor's spell checking. However, additional information could have been checked.

Dates and ages could have been checked for reasonableness and consistency. Scores on standardized tests could be checked for validity and consistency.

**Conclusions**

It is fairly obvious that the designer in this example did not follow a traditional analysis and design approach. That, in itself, does not mean the approach is wrong. It illustrates the way many systems are being developed. Even well-planned, well-staffed projects can fail.

Whether we, as computer professionals, like it or not, this is the wave of the future. Sales people are developing their own customer databases. Doctors are setting up their own networks and developing customized applications. Any technologically savvy individual can now create a new system. When such a system is used only by the individual, then any effects of the system only involve that individual. However, if the system affects others, directly or indirectly, then we may hold the system to a higher standard. We don't want inaccurate patient records in a doctor's office and we don't want inaccurate records in student records.

In the case study, the designer was a knowledgeable user and therefore understood the problem area. The designer didn't need to consult the users to understand how the forms were being created and used. However, this also meant that the other users had no input into the design process; they had no chance to voice concerns or evaluate proposed solutions.

*Contd.....*

In the case study, the designer used the software product (word processing) that they knew best. While any designer does that; most professional systems designers have had a reasonably wide set of training and experiences. This designer was comfortable with word processing and little else. Thus, they didn't explore all the options available. A simple change to using a database package, like Microsoft Access, could have produced a much more useful system; one in which the teachers use one standard "form" for input of all relevant information and then produce the necessary reports. However, if the designer has had no experience with a database package then they have no way of knowing the option exists.

As noted before, the resulting system suffered from a number of other deficiencies. There was no method of error handling; there were no procedures for backups and failure; in fact, there were no procedures at all; the system provided no attempt at data consistency.

We are entering a new software crisis. The original crisis was concerned with the development, by professionals, of reliable and robust software in a cost-effective manner.

The new crisis does not deal with cost, since the investment in a computer system is minimal. Rather the crisis is with the production of reliable, robust software systems by anyone. More and more software, like the example case study, is being created by individuals whose only credentials are an interest in technology.

**Questions**

1. Computer Science (CS) and Computer Information Systems (CIS) programs have responded to the problem in a number of ways. Comment.
2. What are the advantages of resulting system over previous manual system?

*Source:* http://www.micsymposium.org/mics_2005/papers/paper114.pdf

## 1.8 Summary

- The number of phases in the information system development may vary from methodology to methodology. Each phase gives an output that becomes input of next phase.
- Each stage of this development process is highly interrelated and interdependent on each other. The system developer has to perform many activities to accomplish each stage.
- A Systems analyst is a computer specialist who translates business problems and requirements into information systems and acts as liaison between IS (Information Systems) department and rest of the organization.
- System analysis and design refers to the process of examining a business situation with the intent of improving it through better procedures and methods.
- Systems analysis is a management technique which helps us in designing a new system or improving an existing system.
- MIS system is designed to serve the functions of planning, controlling and decision-making at the management level of an organization.
- A Decision Support System (DSS) is an information system designed to help users reach a decision when a decision-making situation arises.
- A Transaction Processing System (TPS) is an information system that captures and processes data generated during an organization's day-to-day transactions.
- Systems development is, basically, a problem-solving action. A problem in an application domain is malformed by the systems development procedure into a solution in the computer's implementation field.
- Implementation means to introduce the designed system into practice or in use.

Notes

## 1.9 Keywords

***Decision Support System:*** System that serves the information needs at management level of an organization.

***Environment of a System:*** Anything outside a system which has an effect on the way the system operates.

***Executive Support System:*** System designed to address unstructured decision making at the strategic level of an organization.

***Information System:*** A people-machine sub-system of the business system that supports the operational, managerial and decision-making information needs of an organization.

***Management Information System:*** System designed to serve the functions of planning, controlling and decision-making at the management level of an organization.

***Office Automation System:*** System that serves the knowledge level of an organization for supporting knowledge workers like production managers, EDP managers, etc.

***Project Manager:*** A computer specialist who monitors the development ad implementation of software in relation to quality, cost and time.

***Sub-system:*** A part of the system that carries one part of the system function.

***System:*** A Coherent set of interdependent component which exists for some purpose, has some stability and can be usefully viewed as a whole, generally portrayed in terms of an input-process-output model existing within a given environment.

***System Analysis:*** Process of gathering and interpreter facts, diagnosing problems and using the information to recommend improvement to the system.

***System Analyst:*** A computer specialist who translates business problems and requirements into information system.

***System Approach:*** A set of procedure for solving a particular problem, to optimize the net results of the operations of an organization.

***System Design:*** The process of planning a new system or replace or complement an existing system.

***System Designer:*** A computer specialist who creates a detailed physical and logical design of the system.

***System Study:*** A study of the operations of a set of connected elements and of the interconnections between these elements.

***Transaction Processing System:*** System designed to serve the operational level of an organization.

## 1.10 Review Questions

1. Illustrate the various phases used in information system development.
2. What is the objective of feasibility study? Depict the components of feasibility study.
3. Make distinction between conceptual design and detailed design.
4. Illustrate significant attributes of a system analyst.
5. Which is in your opinion the most difficult job of a systems analyst?

Notes

6. Who is Systems analyst? Discuss the position of an analyst in a MIS organization.
7. Describe the roles of Systems analyst during various phases of SDLC.
8. Success in system analysis requires interpersonal and technical skills of the analyst. Discuss why.
9. Make distinction between "System approach" and "System analysis".
10. Describe the various approaches used for system development.
11. What is an information system? Elucidate the different types of information systems.

### Answers: Self Assessment

1. organizational
2. system analysis
3. conceptual
4. user interface design
5. System Analyst
6. Program Investigator
7. System design
8. approach
9. subsystem
10. Management Information System
11. Decision Support System (DSS)
12. Transaction Processing System (TPS)
13. design
14. object-oriented
15. Implementation
16. Plunge
17. system developer
18. Maintenance

## 1.11 Further Readings

*Books*

Charles S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*, 2006, Wiley-Interscience.

Dorothy J. Tudor; Ian J. Tudor, *System Analysis and Design: A Comparison of Structured Methods*, 1995, NCC Blackwell.

Elias M. Awad, *Business Data Processing*, 1980, Prentice Hall (Higher Education Division, Pearson Education).

Elias M. Awad, *Systems Analysis and Design*, 1985, R.D. Irwin.

Gary B. Shelly; Thomas J. Cashman, Harry J. Rosenblatt, *System Analysis and Design*, 2001, Course Technology.

Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich, *Modern Systems Analysis and Design*, Addison-Wesley.

Jeffrey L. Whitten; Lonnie D. Bentley; Kevin C. Dittman, *Systems Analysis and Design Methods*, 2003, McGraw-Hill.

John W. Satzinger, Jackson, Burd, *System Analysis & Design,* 2006, Thomson Learning.

Joseph S. Valacich; Joey F. George; Jeffrey A. Hoffer, *Essentials of System Analysis and Design*, 2005, Prentice Hall.

Leonard Fertuck, *System Analysis and Design,* 1994, McGraw-Hill Higher Education.

Penny A. Kendall, *Introduction to System Analysis and Design: A Structured Approach*, 1996, Irwin.

Preeti Gupta, *System Analysis and Design*, 2005, Firewall Media.

Richard Florentz Gonzalez, Claude Mac Millan, *Systems Analysis,* RDIrwin.

Ronald G Noice, *System Analysis & Design*, 1984, Oxford University Press Canadian Branch.

*Online links*

http://www.freetutes.com/systemanalysis/

http://media.wiley.com/product_data/excerpt/87/04700747/0470074787.pdf

# Unit 2: Managing Application Development Portfolio and Project Review & Selection

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the concept of managing the application development portfolio
- Illustrate the information system planning
- Discuss the managing project review and selection

## Introduction

An application is a compilation of computer programs and data that work mutually as a system to execute one or more business functions. It is sometimes known as "application system" or "system. The definition is proposed to permit for some suppleness in explanation. An application portfolio is a collection for a specified customer, of the high-level and thorough information essential for IT and the consumer to build up an investment strategy across the customers' applications (i.e. build up particular improvement or investment tactics for every application where enhancement or investment are valued). In this unit, you will understand the process of managing the application development portfolio. Also you will understand the process of managing project review and selection. As an organization always has many projects (concerned with MIS or other business functions) to do, the management must decide the priority of a system development in order to streamline the business.

Notes

## 2.1 Managing the Application Development Portfolio

In today's Information Technology (I/T) world, how many organizations have a formal procedure for evaluating already organized applications, their influence on the business, and I/T's aptitude to deliver? Each IT organization should have Application approaches that provide value, where:

Value = Benefits/Cost.

Costs can be reduced by dropping or eliminating low value work, decreasing cycle time, and diminishing defects.

Advantages can be maximized if we function on the right things, provide excellence applications, and are receptive to altering business requirements.

One of the main confronts is that of bringing some stage of objectivity to the mission of choosing which business application systems to restore, improve, minimally sustain or just scrap. This matter of "functioning on the correct things" is not often easy and inside IT the recognition of those "correct things" is frequently complexed by verbal proponents and the pressing requirement to "get the job completed" which is apt to unclear the actuality that what is "getting done" isn't "getting done" wonderfully.

Here we concentrate on a technique known as Application Portfolio Management (APM). Managing application development portfolio is used to make sure you are functioning on the correct things. Similar to your stock investment portfolio, applications require to be handled. Some applications may require to be retired (sell), some may necessitate new investment (buy), whereas others should be simply preserved (hold).

By means of APM, one company was competent to decrease their preservation/enhancement workload by 25% by removing low value work. In an additional case, an IT executive exposed that an application he considered had been very productively deployed, was no longer being utilized by its customers as it failed to provide value to its consumers.

Application Portfolio Management is the procedure consumers and IT utilize to maintain the portfolio data current and to formulate investment conclusions.

IT employees and its consumers use each application by means of three different measures. How an application scores comparative to the other applications in the application portfolio assist direct investment decisions.

The three measures utilized to evaluate each application are:

***Technical Quality –*** The application is assessed against factors like maintainability, constancy, and effectual utilization of technology. This assessment is executed in facilitated meetings with the staff that assists applications.

***Functional Quality –***This is a concern of how well the application fulfills the requirements of the business. Factors comprise system usefulness, user-friendliness, reliability, and receptiveness to variations to the business. This concern is determined in a facilitated review with the operational consumers of the system and supervisors/executives who access the system for administration or strategic decision-making.

***Strategic Value –*** This computes the relative significance of a business function to the company and how vital the application is in executing that function. The ratings (High, Medium, or Low) are allocated by company executives. A High specifies that the function and application provide the company a competitive advantage. A Low implies that the application is not vital to the business.

*Notes* A Critical Success Factor in all of these concerns is "fair & balanced" assessments. It is usually best if one person or a very minute team help all of the measurement sessions to assure a reliable strategy.

*Task* Make distinction between technical quality and functional quality.

### 2.1.1 Benefits

Following are advantages to be gained by means of an application portfolio management process:

1. It assures that investments in I/T applications are guided in regions that offer the most value to the company.
2. It allows IT and its user community to interrelate and to together make decisions concerning applications and investments.
3. Highlights regions where the company can save money by recognizing applications where support should be condensed or eliminated.
4. It assists the business unit and IT recognize and concentrates on applications that are high risk to the company (i.e., high strategic significance, but low in both technical and functional excellence).
5. It builds a complete inventory of all applications.
6. It offers the IT user community the capability to observe their whole application portfolio and each application's comparative value to the industry.

### Self Assessment

Fill in the blanks:

1. An .................................. is a collection for a specified customer, of the high-level and thorough information essential for IT and the consumer to build up an investment strategy across the customers' applications.
2. .................................. is the procedure consumers and IT utilize to maintain the portfolio data current and to formulate investment conclusions.
3. .................................. application is assessed against factors like maintainability, constancy, and effectual utilization of technology.
4. .................................. is a concern of how well the application fulfills the requirements of the business.

## 2.2 Information System Planning

Information system planning is defined as a process for producing a strategy and plans for arranging information systems with the business approaches of an organization.

To support supervisors and executives in producing information systems that aid in attaining an organization's corporate assignment, formal planning techniques have been produced and made commercially obtainable. The objective of these planning techniques is to explain an organization-wide way for a firm's information systems. This involves recognizing key elements on which applications are based and are built. It may also comprise illustrating the interrelation of these elements and possibly documenting the present position of the organization's information requirements or outlining future plans. Three broadly used strategies to planning information systems are Business Systems Planning (BSP) method; Nolan, Norton & Co.'s computer architecture strategic planning method; and the critical success factors method.

The BSP approach, which is considered as the most extensively used planning methods, concentrates on recognition of the data essential to run an organization. Nolan, Norton & Co.'s computer architecture strategic planning method associates an organization's present potentials with its future requirements. This method highlights development of a strong technical infrastructure, or basis, to support the applications. Critical success factors method asks for identifying areas that are key to an organization's endurance and to make sure that these elements are included into the organization's information systems.

An organization's corporate work should be reflected all through the grounding and assessment of its information system project requests. Information systems planning needs a vision-a view of the impact that information systems have on an organization's long-term corporate success, tactically and operationally. Having a vision is necessary, but it must also be conversed if the organization's personnel and other resources are to be productively mobilized to put up and preserve a viable information systems plan.

ISP fundamentally entails:

- Identification of the stage of IS in the organization.
- Identification of the applications of organizational ISs.
- Assessment of each of these applications, depending on established evaluation criteria.
- Establishing a main concern ranking for these application.
- Identifying the 'optimum' architecture of IS for serving the top precedence applications.

*Caution* The plan of an organization's information systems must consider the firm's existing systems, in addition to its future needs.

*Did u know?* Information system planning objectives are preferred future situations and destinations, the organizations intend to reach so as to accomplish its task.

## Self Assessment

Fill in the blanks:

5. .................................. planning is defined as a process for producing a strategy and plans for arranging information systems with the business approaches of an organization.

6. To support supervisors and executives in producing information systems that aid in attaining an organization's corporate assignment, .................................. planning techniques have been produced and made commercially obtainable.

7. Information system planning objectives are preferred future situations and destinations, the organizations intend to reach so as to accomplish its .................................. .

Caselet

## Enterprise Management — Building Intelligence into Information Systems

Enterprise Applications (enterprise resource planning, supply chain, customer relationship management, and the like), coupled with business process redesign, are a part of the IT and corporate firmament worldwide. Enterprise Applications are not used just to computerise ways of doing business, but as drivers of the change business corporations must undergo to compete successfully in the information age.

New versions of Enterprise Applications do not limit transformation to individual business corporations. Equipped with powerful adjunct capabilities such as Composite Applications and Service-Oriented Architecture (SOA), they are making wholesale changes to the entire supply chain: suppliers, suppliers' suppliers, customers and customers' customers. The complexities of implementing these packages and the consequent obsession with data management have led many to believe, incorrectly, that there is no life after Enterprise Applications deployment.

While the power of Enterprise Applications is awesome, it falls short of supporting the most vital tasks of management. Sooner than later, business corporations will migrate from mere data management to sophisticated knowledge management efforts. Such efforts will result in a set of powerful information systems called Enterprise Management Systems (EMS) that will be a legitimate sequel to Enterprise Applications.

**Task of Management**

Many management thinkers, including Peter Drucker, have presented diverse models to describe the task of management. But none argues the case more cogently than the Shewhart Cycle, known better to the world as the Deming Cycle. The Deming Cycle was originally proposed by Walter Shewhart and popularised in Japan by Edwards Deming after which it came to be known as the Deming Cycle. The Deming Cycle, sums up the task of management as a closed loop activity system.

The four principal steps in the Deming Cycle are:

***Plan:*** Define the destination you are aspiring for. The destination could be that of the enterprise, of a business process or of a small team. Using appropriate techniques validate that this destination is worthy of reaching. Generate alternative ways of reaching the destination and choose one.

***Do:*** Carry out the plan of action. The plan is implemented in one or more business processes.

***Check:*** Using appropriate measurement systems determine if the destination was reached. If you did, go back to generating fresh plans.

If not, move to the next step, Act.

***Act:*** As in the Plan step, except for being informed by the results from the Check step, generate alternatives to close the gap between the current state and the destination. Migrate to the Plan step to add plans for new initiatives.

The endless repetition of this P-D-C-A cycle constitutes the task of managing and running the business enterprise.

*Contd.....*

The Deming Cycle is a fractal. One cycle spawns others. The actions emerging out of the 'Do' step at a higher level of abstraction become the contents of the 'Plan' step of the next lower level. For example, the 'Plan' of increasing market share at one level may spawn a 'Do' step of running a media campaign with a designated effectiveness. This becomes the content for the 'Plan' step of the business process that will implement the campaign. Thus the fractal chain is used at practically all levels in the organisation: Corporate, business unit, business process or individual.

The Deming Cycle can be applied equally to a strategy or an operations problem. It can, for example, be applied to the operational problem of cutting down the cycle time of the new product development process. It can be applied equally to the strategic problem of gaining market share by the addition of new features to a product that would change its positioning in the marketplace.

**Role of Enterprise Applications**

The domain of Enterprise Applications is primarily the 'Do' and 'Check' steps. For the greater part, ERP implementation has been an "equalising" effort rather than a "differentiating" effort. A handful of discerning enterprises have used the strength of ERP with the power of Business Process Redesign to develop signature processes that lead to a distinct competitive advantage.

Business Intelligence extensions to Enterprise Applications, performance measurement systems and their ilk assist management with the 'Check' step.

Retrospective Analytics work more at the 'Check' step than at the 'Plan' or 'Act' steps to provide discernible patterns that might account for the gap between actual results and the goals evolved during the Plan step. The emergence of Predictive Analytics is providing the first glimmer of substantive assistance to managers, both at the Plan and Check levels. Yet Predictive Analytics is incapable of using expert rules and heuristic reasoning.

Today's Enterprise Applications concentrate on the Do and Check steps and offer scanty support for the Plan and Act steps. Therefore, IT-enabled management is still not a closed-loop system. However, the proposition of Enterprise Management Systems (EMS) will permit the installation of a powerful, revitalised Deming Cycle.

**Enterprise Management Systems**

Enterprise Management Systems (EMS) are intelligent information systems that embed specialist components to support the 'Plan' and 'Act' steps. An EMS consists of three principal components:

Enterprise Applications (Transactions) that contain data relating to the thousands of transactions by which an enterprise conducts its business Enterprise Applications (Business Intelligence) that incorporate the power of both Retrospective Analytics and Predictive Analytics that reveal patterns, both past and prospective, contained in the transaction layer.

Enterprise Planning Systems (EPS) that contain the knowledge base and the inference mechanisms which act upon both transactions and patterns to generate alternatives for the 'Plan' and 'Act' steps.

EMS help management teams continuously evolve plans at the various levels of management, such as corporate, strategic business unit, and business process. They support the P-D-C-A fractal evolution and make IT-enabled management a closed loop.

The process of generating 'Plan' and 'Act' steps has to be understood from a computational and reasoning perspective. Computational sciences, in general, and Artificial Intelligence

*Contd.....*

(AI), in particular, have dealt with planning as state transition. The Plan and Act steps deal with the transition from the current state to a desired state.

For example, in the new product development process, the current state of the process could be a cycle time of 130 days. The desired state could be a cycle time of 80 days. The Plan step, using the knowledge repository and inference mechanisms of EPS, would generate alternatives for transitioning from the current state (cycle time of 130 days) to the desired state (cycle time of 80 days).

These alternatives are implemented in the form of business transactions by Enterprise Applications (in this case, perhaps a PLM application). The Check step determines whether the desired state (actual cycle time of 80 days) was achieved. If not, the Act step, enabled by appropriate EPS components, generates a set of alternatives to close the gap. At the heart of EPS would be knowledge representation schemes, inference mechanisms, search strategies and heuristic reasoning mechanisms.

The need for EMS is now being filled by manual, cumbersome and unreliable methods. Most often the success of reasoning is dependent on a few experts and intuition. Some smart companies have written bespoke applications that perform the function of EMS, but are finding such applications incredibly difficult to maintain and grow.

The EMS architecture and EPS in specific will draw significantly on techniques of AI and computational sciences. There has been some disappointment in the industry with the use of these techniques in the not-too-distant past. With Enterprise Applications acting as a solid bedrock of data, elegant user interfaces, and the dramatic reduction in the cost of computing with an equally dramatic rise in computing power, there is a strong case for revisiting extant assumptions and feelings about intelligent search and planning techniques.

**Knowledge Management**

Clearly data management now is a victim of the law of diminishing returns. New incremental investments in data management are returning less and less. The competitive differentiation offered to early adopters of Enterprise Applications has been all but eliminated. True competitive advantage consists in the intelligent management of a corporation's specific problem-solving knowledge that gives it an edge. The manufacturing efficiencies of Dell, the management of cost per seat-mile of Southwest and Wal-Mart's supply chain management knowledge, empowered by technology, give them their competitive advantage — not run-of-the-mill Enterprise Applications.

Knowledge Management, which is the orderly encapsulation, perpetuation and deployment of such critical knowledge, is vital to foster a firm's competitiveness. Enterprise Management Systems (EMS) provide a management and technology framework for superior Knowledge Management.

(The author is Founder and CEO of Anantara Solutions Private Limited. TM Enterprise Management Systems (EMS) and Enterprise Planning Systems (EPS) are trademarks under application by Anantara Solutions Private Limited.)

*Source:* http://www.thehindubusinessline.in/2007/06/19/stories/2007061900710900.htm

## 2.3 Project Selection

System analysts do not start working on any projects they desire. They receive a lot of request from the management for starting different type of the projects. When projects are formally requested, the systems analysts, under the managements direction, conduct a preliminary investigation to analyze the reasons for the request and collect various facts to respond to the request in a systematic way. Some projects are feasible, while others may not be feasible for various reasons.

Developing a new information system is one kind of a planned organizational change. The introduction of a new system includes purchase of new hardware and software, appointment of skilled persons and/or training of existing staff for using the new system and changes in management and organization. So, developing a new system requires redesigning the organization. As an organization always has many projects (concerned with MIS or other business functions) to do, the management must decide the priority of a system development in order to streamline the business.

### 2.3.1 Reasons for Developing New Systems Project

System projects are initiated for different reasons. The most important reasons are:

1. ***Capability:*** Business activities are influenced by an organization's ability to process transactions quickly and efficiently. Information systems add capability in three ways:

   (i) *Improved Processing Speed:* The inherent speed with which computers process data is one reason why organisation seek the development of systems projects.

   (ii) *Increased Volume:* Provide capacity to process a greater amount of data, perhaps to take advantage of new business opportunities.

   (iii) *Faster Retrieval of Information:* Locating and retrieving information from storage. The ability in conducting complex searches.

2. ***Control:***

   (i) *Greater Accuracy and Consistency:* Carrying out computing steps, including arithmetic, correctly and consistently.

   (ii) *Better Security:* Safeguarding sensitive and important data in a form that is accessible only to authorised personnel.

3. ***Communication:***

   (i) *Enhanced Communication:* Speeding the flow of information and messages between remote locations as well as within offices. This includes the transmission of documents within offices.

   (ii) *Integration of Business Areas:* Coordinating business activities taking place in separate areas of an organisation, through capture and distribution of information.

4. ***Cost:***

   (i) *Monitor Costs:* Tracking the costs of labour, goods and overhead is essential to determine whether a firm is performing a line with expectations-within budget.

   (ii) *Reduce Costs:* Using computing capability to process data at a lower cost than possible with other methods, while maintaining accuracy and performance levels.

5. ***Competitiveness:***

   (i) *Lock in Customers:* Changing the relationship with and services provided to customers in such a way that they will not think of changing suppliers.

   (ii) *Lock out Competitors:* Reducing the chances of entering the competitors in the same market because of good information systems being used in the organization.

(iii) *Improve Arrangements with Suppliers:* Changing the pricing, service or delivery arrangements, or relationship between suppliers and the organization to benefit the firm.

(iv) *New Product Development:* Introducing new products with characteristics that use or are influenced by information technology.

## 2.3.2 Sources of Projects Requests

There are mainly four primary sources of project requests. The requesters inside the organization are:

- Department Managers
- Senior Executives
- System Analysts
- Outside Groups

In addition, government agencies outside the organization may also ask for information systems projects.

### Requests from Department Managers

Frequently, department managers who deal with day-to-day business activities, are looking for assistance with their departments. They are often not satisfied with the amount of time that the staff takes to complete the job. Sometimes, they feel that the staff members are involved in duplication of work also. In this case, the manager will discuss this problem with other administrators regarding their clerical as well as processing work and persuade higher authority to approve the development of a computer based system for office administration.

### Requests from Senior Executives

Seniors executives like Presidents, Vice-Presidents usually have more information about the organization as compared to department managers. Since these executives manage the entire organization, so naturally they have broader responsibilities.

*Did u know?* Systems project requests submitted by senior executives carry more weightage and are generally broader in scope also.

### Requests from System Analysts

Sometimes systems analysts finalize areas where it is possible to develop projects. In such cases, they may prefer either writing systems proposal themselves or encouraging a manager to allow the writing of a proposal on their behalf.

*Example:* For example, in an organization, an analyst sees that the library information system takes more time in processing and is inefficient, may prepare a project proposal for a new library information system. But the direction of the analyst, who is fully aware about the new technology that improves the existing library information system, the librarian may initiate the development of information system to the higher authority for approval.

### Requests from Outside Groups

Developments outside the organization also lead to project requests.

*Example:* Government contractors are required to use special cost accounting systems with government stipulated features.

Generally, it has been observed that new demands from external groups bring about project requests, either for new systems or changes in current ones. Project requests originated from this source are also quite important.

## Self Assessment

Fill in the blanks:

8. When projects are formally requested, the systems analysts, under the management's direction, conduct a preliminary investigation to .................................. the reasons for the request.

9. Developing a new information system is one kind of a .................................. organizational change.

## 2.4 Managing Project Review and Selection

It is true that a number of requests for systems development are generated in the organization. Someone in the organization must decide which requests to pursue and which to reject.

The management decides the priority of a system development by reviewing the answers of the following questions:

- When should the organisation go for computerisation, if it is still using manual systems?
- Are the users satisfied with the performance of existing systems (manual/computerised)? If not, what are the reasons?
- What are the major problems of the existing systems? Do they effect the normal working of the organisation? If yes, how long can the organisation tolerate such problems?
- What are the major projects that the organisation has to do? Which one of these is the most important?

*Caution* The management must ensure that the most important systems are developed first, followed by the less important ones and the least important ones in the last.

The criteria to accept or reject a request can be decided in a number of ways. One of the suitable methods commonly in use is by committee. Mainly three committee's formats are commonly used:

### 2.4.1 Steering Committee Method

This is one of the most common methods of reviewing and selecting projects for development. Such a committee, consisting of key managers from various departments of the organisation as well as members of information systems group, is responsible for supervising the review of project proposals. This committee receives requests for proposal and evaluates them. The main responsibility of the committee is to take decision, which often requires more information than the proposal provides. It is, therefore, desired to have preliminary investigation to gather more details. The steering committee approach is generally favoured because systems projects are considered as business investments. Management, not systems analysts or designers, selects projects for development. Decisions are made on the basis of the cost of the project, its benefits to the organization and the feasibility of accomplishing the development within the limits of information systems technology.

### 2.4.2 Information Systems Committee Method

In some organization, the responsibility for reviewing project requests is entrusted to a committee of managers and analysts in the information systems department. Under this method, all requests for service and development are submitted directly to a review committee within the information systems department. This committee approves or disapproves projects and sets priorities, indicating which projects are most important and should receive immediate attention. This method can be used when many requests are for routine services or maintenance of existing applications. When major equipment decisions are required or when long-term development commitments are needed to undertake a project, the decision authority decided whether a project should proceed or not. So, the major functions of this committee are:

(i) To review the systems plan and approve/disapprove them.

(ii) To integrate the systems that share the input data.

(iii) To provide alternatives to the project.

*Task* Depict major functions of Information Systems Committee Method.

### 2.4.3 User Group Committee Method

In some organization, the responsibility for project decisions is entrusted to the users themselves. Individual departments hire their own analysts and designers who handle project selection and carry out development. Although the practice of having user committees for both choose and develop systems does take some of the burden from the systems development group it can have disadvantages for the users. Some users' groups may find themselves with defective or poorly designed systems that require additional time and effort to undo any damage caused by the misinformation that such systems could generate.

*Notes* Although users groups may find the decisions of steering committees and Information Systems Committees disappointing at times, the success rate for users who undertake development job is not very encouraging.

**Project Request Contents**

The project proposals submitted by the users or the analysts to the Project Selection Committee is a critical element in launching the systems study. There is a general agreement that a project request form should contain the following:

- What is the problem?
- What are the details of the problem?

Notes

- How significant is the problem?
- What does user feel is the solution?
- How will the information systems help?
- Who else knows about this and could be contacted?

The project selection committee is responsible to review the proposals carefully and finally selects those projects which are most beneficial to the organisation. Therefore, a preliminary investigation is often requested to gather details which are asked in the project request-forms.

### Self Assessment

Fill in the blanks:

10. Someone in the organization must decide which .................................. to pursue and which to reject.
11. .................................. committee, consisting of key managers from various departments of the organization as well as members of information systems group, is responsible for supervising the review of project proposals.
12. .................................. committee approves or disapproves projects and sets priorities, indicating which projects are most important and should receive immediate attention.
13. In .................................. Committee Method, Individual departments hire their own analysts and designers who handle project selection and carry out development.
14. The project proposals submitted by the users or the analysts to the .................................. Committee is a critical element in launching the systems study.
15. A .................................. is often requested to gather details which are asked in the project request-forms.

## 2.5 Summary

- Application Portfolio Management is the procedure consumers and IT utilize to maintain the portfolio data current and to formulate investment conclusions.
- Managing application development portfolio is used to make sure you are functioning on the correct things.
- Information system planning is defined as a process for producing a strategy and plans for arranging information systems with the business approaches of an organization.
- Three broadly used strategies to planning information systems are business systems planning (BSP) method; Nolan, Norton & Co.'s computer architecture strategic planning method; and the critical success factors method.
- Information system planning objectives are preferred future situations and destinations the organizations intend to reach so as to accomplish its task.
- As an organization always has many projects (concerned with MIS or other business functions) to do, the management must decide the priority of a system development in order to streamline the business.
- Steering committee, consisting of key managers from various departments of the organization as well as members of information systems group, is responsible for supervising the review of project proposals.

- Information system committee approves or disapproves projects and sets priorities, indicating which projects are most important and should receive immediate attention.

## 2.6 Keywords

***Application Portfolio:*** An application portfolio is a collection for a specified customer, of the high-level and thorough information essential for IT and the consumer to build up an investment strategy across the customers' applications.

***Application Portfolio Management:*** It is the procedure consumers and IT utilize to maintain the portfolio data current and to formulate investment conclusions.

***Information Systems Committee:*** A committee of managers and analysts in the information systems department.

***Steering Committee:*** A committee consisting of key managers from various departments of the organization as well as members of information systems group.

## 2.7 Review Questions

1. Illustrate the process of managing the application development portfolio.
2. List the advantages of an application portfolio management process.
3. Elucidate the procedure of information system planning.
4. Why do we develop a new systems project? Illustrate.
5. Describe different committees formats used for managing projects.
6. List the questions reviewed by the management to decide the priority of a system development.
7. What are the functions of a long-term development committee?
8. Discuss some of the contents of a project request.
9. Describe the most significant reasons for a project to be initiated.
10. Enlighten various primary sources of project request.
11. Make distinction between Information system committee method and User Group Committee Method.

### Answers: Self Assessment

1. application portfolio
2. Application Portfolio Management
3. Technical Quality
4. Functional Quality
5. Information system
6. formal
7. task
8. analyze
9. planned
10. requests
11. Steering
12. Information system
13. User Group
14. Project Selection
15. preliminary investigation

Notes

## 2.8 Further Readings

*Books*

Erik W. Larson, Clifford F. Gray, *Project Management*, McGraw-Hill Professional.

Pankoj Jalote, *An Integrated Approach to Software Engineering,* Springer.

S.A. Kelkar, *Structured System Analysis and Design*, Prentice Hall of India.

*Online link*

http://www.nascio.org/awards/nominations/State-TN-InfoSystemsl.pdf

Notes

# Unit 3: Analysis

## Objectives

After studying this unit, you will be able to:

- Understand the concept of preliminary investigation
- Recognize the scope of study
- Discuss the conducting the investigation
- Understand the testing project feasibility
- Discuss the handling infeasible projects

## Introduction

After the feasibility study, the analyst takes the formal acceptance of the proposed system from the requested department. The next step is to study the current system in detail, so that the system requirements can be determined. During systems analysis, the analyst obtains actual specifications of the system by clearly understanding the needs of the users. After analysis, a document is prepared by the analyst, which is called as Software Requirement Specification (SRS) document.

Many methods and languages have been developed for requirements analysis. Structured Analysis and Design Technique (SADT) – simply called as 'Structured Analysis' – is the most commonly used method for analysis. We will discuss this method later. Problem Statement Language (PSL) and Requirements Statement Language(RSL) are some of the languages developed for specifying the requirements. PSL is fundamentally the same as the structured analysis, but it is purely textual whereas structured analysis uses both text and graphics. RSL has been specially designed for specifying the requirements of real time control systems.

Data collection is an important part of feasibility analysis and systems analysis phase of SDLC. Many techniques are used for collection of data, which are commonly known as fact finding techniques.

Notes

## 3.1 Preliminary Investigation

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgement about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

1. Clarify and Understand the Project Request. What is being done? What is required? And why? Is there an underlying reason different from the one the user identifies?
2. Determine the size of the project.
3. Access costs and benefits of alternative approaches.
4. Determine the technical and operational feasibility of alternative approaches.
5. Report the finding to management, with recommendations outlining the acceptance or rejection of the proposal.

### Self Assessment

Fill in the blanks:

1. After analysis, a document is prepared by the analyst, which is called as .................................... document.
2. .................................... Language is fundamentally the same as the structured analysis, but it is purely textual whereas structured analysis uses both text and graphics.
3. .................................... Language has been specially designed for specifying the requirements of real time control systems.
4. The first step in the system development life cycle is the preliminary investigation to determine the .................................... of the system.
5. .................................... is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgement about the feasibility of the proposed project.

## 3.2 Scope of Study

The preliminary review of the system is performed which assists in recognizing the scope of the system. Depending on the result of the original study, feasibility study occurs. The feasibility study is essentially the test of the projected system in the light of its workability, fulfilling user's needs, effective utilization of resources and .obviously, the cost usefulness.

*Did u know?* The major objective of feasibility study is not to resolve the problem but to attain the scope.

*Caution* In the process of feasibility study, the cost and benefits are predicted with greater correctness.

## Self Assessment

Fill in the blanks:

6. The preliminary review of the system assists in recognizing the ................................... of the system.

7. The major objective of ................................... study is not to resolve the problem but to attain the scope.

## 3.3 Conducting the Investigation

The data collected by the analysts during preliminary investigations are gathered through three primary methods:

1. ***Reviewing Organization Documents:*** The analysts conducting the investigation first learn about the organization involved in, or affected by the project.

   *Example:* To review an inventory systems proposal means knowing first how the department works and who are the persons directly associated with inventory system. Analysts can get some details by examining organization charts and studying written operating procedures. The procedures clearly define various important steps involved in receiving, managing and dispersing stock.

2. ***On-site Observations:*** In this method, the analysts observe the activities of the system directly. One purpose of on-site observation is to get as close as possible to the real system being studied. During on-site observation, the analyst can see the office environment, work local of the system and the users, methods of work and the facilities provided by the organization to the users.

3. ***Conducting Interviews:*** The above two methods tell the analysts how the system should operate, but they may not include enough details to allow a decision to be made about the merits of a system proposal, nor do they present user views about current operations. Analysts use interview to learn these details.

*Did u know?* Interviews allow analysts to learn more about the nature of the project request and the reason for submitting it.

*Caution* Interview should provide details that further explain the project and show whether assistance is merited economically, operationally and technically.

*Task* Discuss how to review Organization Documents.

## Self Assessment

Fill in the blanks:

8. The analysts conducting the ................................... first learn about the organization involved in, or affected by the project.

9. In  method, the analysts observe the activities of the system directly.

10. .................................. allow analysts to learn more about the nature of the project request and the reason for submitting it.

## 3.4 Testing Project Feasibility

Feasibility is the determination of whether a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible. The analyst can go ahead and prepare at the project specification which finalises project requirements. Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendations of such a study will be used as a sound bases for deciding whether to proceed, postpone or cancel the project.

*Notes* Since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgement are made.

Preliminary investigations inspect project feasibility; the probability the system will be functional to the organization. Three significant tests of feasibility are considered and described below:

1. ***Technical Feasibility:*** This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

   - The facility to produce outputs in a given time.
   - Response time under certain conditions.
   - Ability to process a certain volume of transaction at a particular speed.
   - Facility to communicate data to distant location.

   Out of all types of feasibility, technical feasibility generally is the most difficult to determine.

2. ***Operation Feasibility:*** It is mainly related to human organisational and political aspects. The points to be considered are:

   - What changes will be brought with the system?
   - What organisational structures are disturbed?
   - What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

   Generally project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations. This feasibility study is carried out by a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and are skilled in system analysis and design process.

3. ***Economic Feasibility:*** Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis; the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits out weigh costs, a decision is

taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it has a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

A number of approaches for assessing the costs of solutions have been suggested. Approaches include the following:

- *Last cost:* This is based on the observation that costs are easier to control and identify the revenues. Thus, it assumes that there is no change in income caused by the implementation of a new system. In such an evaluation, only the costs are listed and the option with the lowest cost is selected.
- *Time to Payback:* This method of economic evaluation is an attempt to answer the question. How long would it be until we get out money back on this investment in system? This requires data on both costs and benefits. This method of evaluation has two significant disadvantages:
  - It only considers the time taken to return the original investment and ignores the system's long term profitability.
  - The method does not recognize the time value of money. Benefits that accrue in the distant future are not worth as much as similar benefits that occur more quickly but this method fails to recognize this.
- *Cost-effectiveness:* Some type of cost benefit analysis is performed for each alternative. Rough projections of equipment requirements and costs, operational costs, manpower costs, maintenance cost, etc., need to be made. Projections of potential, tangible as well intangible benefits are also needed to be made.

*Example:* Tangible benefits are ability to obtain information, which was previously not available, faster or timely receipt of information, improved or better decision making, improvement in planning and control etc.

In the conduct of the feasibility study, some more interrelated types of feasibility can be considered are discussed below:

1. ***Social Feasibility:*** Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project being accepted by the group directly affected by the proposed system change.
2. ***Management Feasibility:*** It is a determination of whether a proposed project will be acceptable to management. If management does not accept a project or gives a negligible support to it, the analyst will tend to view the project as a non-feasible one.
3. ***Legal Feasibility:*** Legal feasibility is a determination of whether a proposed project infringes on known Acts, statutes, as well as any pending legislation. Although in some instances the project might appear sound, on closer investigation it may be found to infringe on several legal areas.
4. ***Time Feasibility:*** Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame. If a project takes too much time it is likely to be rejected.

*Task* Depict the disadvantages of 'time to payback' method.

Notes

*Caselet*

## Outer Harbour Development — Tuticorin Port Invites Bids for Feasibility Report

The Tuticorin Port Trust (TPT) has invited tenders from consultants for preparing a feasibility report for developing the outer harbour in the port.

The project, which is estimated to cost ₹ 2,150 crore, was recently approved by the Centre and included in the National Maritime Development Programme.

The feasibility report would include deepening of dock basin to cater to 14.50 metre draught vessels, formation of breakwater and construction of berths with other provisions such as electrical installations, port safety and fire-fighting facilities.

The report would focus on analysis of the existing situation, nature and magnitude of the problem to be addressed, need and justification for the project in the context of national private alternative strategies.

It would also take care of initial environment and social impact analysis, preliminary site investigation based on earlier reports, stakeholder commitment and risk factor, says information available on the port trust's Web site.

The report would contain traffic forecast up to 2025, vessel traffic analysis (selection of design vessels) and evaluation of optimised layout for outer harbour, breakwater system, area requirement for berth, dock basin, turning circle, channel dimension (length, width, depth) and back up area requirement.

The Tuticorin port, which is an artificial harbour, has six alongside berths, one multipurpose berth, one container terminal, two coal jetties, one oil jetty and one shallow draught berth.

The Tuticorin port, even after the ongoing construction projects and possible increase in the facilities inside the existing harbour, has to look for other developments for meeting these future needs.

Also, by virtue of its location being the closest compared to the other Indian ports to the international marine container route, the port authorities intend to plan development of an outer harbour, information available in the Web site says.

***Source:*** http://www.thehindubusinessline.in/2005/09/06/stories/2005090601480700.htm

### Self Assessment

Fill in the blanks:

11. .................................... is concerned with specifying equipment and software that will successfully satisfy the user requirement.

12. ................................... feasibility is a determination of whether a proposed project will be acceptable to the people or not.

13. ................................... feasibility is a determination of whether a proposed project infringes on known Acts, statutes, as well as any pending legislation.

14. ................................... feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame.

## 3.5 Handling Infeasible Projects

It is not essential that all projects that are submitted for assessment and examination are good enough. Generally, requests that do not pass all the feasibility tests are not pursued further, unless they are customized and re-submitted as fresh proposals. Sometimes, it so happens that a part of a recently developed system is not functional and the collection committee may choose to unite the workable portion of the project with another feasible proposal.

*Notes* In extra cases, preliminary investigations generate enough new information to propose that improvements in management and supervision, not the expansion of information systems, are the genuine solutions to reported troubles.

### Self Assessment

Fill in the blanks:

15. Generally, requests that do not pass all the feasibility .................................. are not pursued further, unless they are customized and re-submitted as fresh proposals.

*Case Study*

## Preliminary Analysis: Library Management System

**Request Clarification**

First the management of the library approached this ABC Software Ltd. for their request for the new automated system. What they stated in their request was that they needed a system for their library that could automate its various functions. And provide faster response. From this request statement, it is very difficult for the analyst to know what exactly the customer wants. So in order to get information about the system, the analyst visits the library site and meets the staff of the library. Library staff is going to be the end user of the system. Analyst asks various questions from the staff so that the exact requirements for the system become clear. From this activity, the analyst is able to identify the following requirements for the new system:

- Function for issue of books
- Function for return of books that can also calculate the fine if the book is returned after the due date
- Function for performing different queries
- Report generation functions
- Function for maintaining accounts
- Maintaining the details for members, books, and suppliers in some structured way.

Now that the requirements are known, the analyst proposes solution system.

***Solution:*** The desired system can be implemented with Oracle RDBMS in the back end with Visual Basic as the front end. It will have modules for handling issue and return

*Contd.....*

functions, generating reports, performing checks, and maintaining accounts. It will also store the data relating to books, members, and suppliers in a structures way. In our case, the data will be maintained in a relational way.

**Feasibility Study**

Now the next stage in preliminary analysis is to determine whether the proposed solution is practical enough to be implemented. For this feasibility study is done. First technical feasibility is done. Major issues in technical feasibility are to see if the required resources-trained man power, software and hardware are available or not. ABC Software Ltd. is big IT Company. It has developed similar projects using Oracle and VB. It has a special team that is formed to work in this combination of projects, that is, Oracle and Visual Basic. So manpower is readily available. The software is available with the company since it has already worked with the same software earlier also. So our solution is technically feasible. Technical feasibility doesn't guarantee if the system will be beneficial to the system if developed. For this economic feasibility is done.

First task that is done in economic analysis is to identify the cost and benefit factors in the system proposed. In our case, the analyst has identified the following costs and benefits. First task that is done in economic analysis is to identify the cost and benefit factors in the system proposed. In our case, the analyst has identified the following costs and benefits.

**Cost**

| Cost | Cost per unit | Quantity | Total Cost |
|---|---|---|---|
| Software | | | |
| Oracle | 50,000 | 1 | 50,000 |
| VisualBasic | 30,000 | 1 | 30,000 |
| Windows Server 2003 | 15,000 | 1 | 15,000 |
| Windows XP professional | 5,000 | 4 | 5,000 |
| Hardware | | | |
| Central Computer | 100,000 | 1 | 100,000 |
| Client Machine | 50,000 | 4 | 50,000 |
| Development | 50,000 | 1 | 50,000 |
| Analyst | 50,000 | 1 | 50,000 |
| Developer | 20,000 | 2 | 40,000 |
| Training | 20,000 | 1 | 20,000 |
| Data Entry | 5,000 | 1 | 5,000 |
| Warranty ( 1 month) Professional | 20,000 | 1 | 20,000 |
| **Total Cost** | | | 5,55,000 |

**Benefits**

According to new policy: A member is required to pay ₹ 500 for a half yearly membership and ₹ 1000 for a year membership. Expected increase in number of members: 75 per month 40 new members for 1 year and 35 new members for half year.

*Contd.....*

| | |
|---|---|
| Free collected from new members in one year | = 12 (40 × 1000 + 35 × 500) |
| | = ₹ 6,90,000 |
| For four years | = 4 × 6,90,000 |
| | = 27,60,000 |

Now using Net present value method for cost benefit analysis we have,

| | |
|---|---|
| Net present value (or gain) | = Benefits – Costs |
| | = 27,60,000 – 5,55,000 |
| | = 22,10,000 |
| Gain % | = Net present value/investment |
| | = 22,10,000/5,55,000 |
| | = 4.018 |
| Overall Gain | = 401.8 % in four year |
| For each year | |
| First year | |
| Investment | = 5,50,000 |
| Benefit | = 6,90,000 |
| Net present value for first year | = 6,90,000 – 5,50,000 |
| | = 1,40,000 |
| Gain % | = 1,40,000/5,50,000 |
| | = .254 = 25.4 % in first year |
| Second Year | |
| Investment | = 5,50,000 |
| Benefit | = 13,80,000 |
| Net present value for second year | = 13,80,000 – 5,50,000 |
| | = 8,30,000 |
| Gain % | = 830000/550000 |
| | = 1.50 = 150 % at the end of second year |
| Third Year | |
| Investment | = 5,50,000 |
| Benefit | = 20,70,000 |
| Net present value for third year | = 20,70,000 – 5,50,000 = 15,20,000 |
| Gain % | = 1520000/550000 |
| | = 2.76 = 276 % at the end of third year |

*Contd.....*

Notes

Fourth Year

| | |
|---|---|
| Investment | = 550,000 |
| Benefit | = 2760000 |
| Net Present Value for fourth year | = 2760000 – 550000 |
| | = 2210000 |
| Gain % | = 2210000/550000 |
| | = 4.018 |
| | = 401.8 % at end of fourth year |

From CBA we have found that it is economically feasible since it is showing great gains (about 400%). After economic feasibility, operational feasibility is done. In this, major issue is to see if the system is developed what is the likelihood that it'll be implemented and put to operation? Will there be any resistance from its user? It is very clear that the new automated system will work more efficiently and faster. So the users will certainly accept it. Also they are being actively involved in the development of the new system. Due to this fact they will know the system well and will be happy to use a new improved system. So our system is operationally feasible. After the feasibility study has been done and it is found to be feasible, the management has approved this project.

**Questions**

1. List the requirements needed for the new system.
2. What is economic analysis?

***Source:*** http://www.scribd.com/doc/50267807/35/Preliminary-Analysis-Case-study-Library-Management-System

## 3.6 Summary

- During systems analysis, the analyst obtains actual specifications of the system by clearly understanding the needs of the users.
- After analysis, a document is prepared by the analyst, which is called as Software Requirement Specification (SRS) document.
- The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system.
- Preliminary investigation is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgement about the feasibility of the proposed project.
- The analysts conducting the investigation first learn about the organisation involved in, or affected by the project.
- During on-site observation, the analyst can see the office environment, work local of the system and the users, methods of work and the facilities provided by the organisation to the users.
- Feasibility is the determination of whether a project is worth doing. The process followed in making this determination is called a feasibility study.
- Technical Feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement.

- Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system.
- Generally, requests that do not pass all the feasibility tests are not pursued further, unless they are customized and re-submitted as fresh proposals.

## 3.7 Keywords

***Economic Feasibility:*** To determine the benefits and savings that are expected from a proposed system and compare them with costs.

***Feasibility:*** The determination of whether a project is worth doing.

***Legal Feasibility:*** A determination of whether a proposed project infringes on known Acts, statutes, as well as any pending legislation.

***Management Feasibility:*** A determination of whether a proposed project will be acceptable to management.

***Preliminary investigation*****:** It is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgement about the feasibility of the proposed project.

***Social Feasibility:*** A determination of whether a proposed project will be acceptable to the people or not.

***Technical Feasibility:*** Specifying equipment and software that will successfully satisfy the user requirement.

***Time Feasibility:*** A determination of whether a proposed project can be implemented fully within a stipulated time frame.

## 3.8 Review Questions

1. What is preliminary investigation? Enlighten the objectives of preliminary investigation.
2. Explain the process of recognizing the scope of the system.
3. Explicate the various methods used for conducting the investigation.
4. What do you mean by feasibility of the project? Illustrate.
5. Describe how to test project feasibility.
6. Depict different types of feasibility.
7. Elucidate the approaches used for assessing the costs of solutions.
8. Make distinction between social feasibility and management feasibility.
9. Explicate the process of managing infeasible projects.
10. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. Comment.

### Answers: Self Assessment

1. Software Requirement Specification (SRS)

2. Problem Statement     3. Requirements Statement

**Notes**

4. feasibility
5. Preliminary investigation
6. scope
7. feasibility
8. investigation
9. On-Site Observations
10. Interviews
11. Technical Feasibility
12. Social
13. Legal
14. Time
15. tests

## 3.9 Further Readings

*Books*

Patrick McDermott, *Zen and the Art of Systems Analysis*, iUniverse.

Robert Sugden, Allan M. Williams, *The Principles of Practical Cost-Benefit Analysis*, Oxford University Press.

William N. Sweet, Alexander Kossiakoff, *Systems Engineering*, Wiley-IEEE.

*Online link*

http://newton.uor.edu/courses/sysanades/pdf/anaintro.pdf

# Unit 4: System Requirements

CONTENTS

## Objectives

After studying this unit, you will be able to:

- Understand the tools for system requirements
- Discuss the requirements determination
- Recognize the activities in requirement analysis
- Understand the types of system requirements

## Introduction

Since systems analysts do not work as managers or employees in user departments (such as marketing, purchasing, manufacturing or accounting), they do not have the same base of facts and details as the managers and users in those areas. Therefore, an early step in the analyst's investigation is to understand the situation. Certain types of requirements are so fundamental as to be common in most situations. Developing answers to a specific group of questions will help you understand these basic requirements.

There are also special kinds of requirements that arise, depending on whether the system is transaction or decision oriented and whether the system cuts across several departments. For instance, the need to inform the inventory manager of an unusually large order that is forthcoming underscores the importance of linking the sales, purchasing and warehouse departments.

Notes

## 4.1 Tools for System Requirements

Conceptually, requirements analysis includes three types of activity:

- *Eliciting requirements:* The task of communicating with customers and users to determine what their requirements are.
- *Analyzing requirements:* Determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.
- *Recording requirements:* Requirements may be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. Historically, this has included such things as holding interviews, or holding focus groups and creating requirements lists. More modern techniques include prototyping, and use cases.

*Notes* Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced.

### 4.1.1 Stakeholder Interviews

Stakeholder interviews are a common method used in requirement analysis. Some selection is usually necessary, cost being one factor in deciding whom to interview. These interviews may reveal requirements not previously envisaged as being within the scope of the project, and requirements may be contradictory. However, each stakeholder will have an idea of their expectation or will have visualized their requirements.

### 4.1.2 Joint Requirements Development Sessions

Requirements often have cross-functional implications that are unknown to individual stakeholders and often missed or incompletely defined during stakeholder interviews. These cross-functional implications can be elicited by conducting JRD sessions in a controlled environment, facilitated by a Business Analyst, wherein stakeholders participate in discussions to elicit requirements, analyze their details and uncover cross-functional implications. A dedicated scribe to document the discussion is often useful, freeing the Business Analyst to focus on the requirements definition process and guide the discussion.

### 4.1.3 Contract-style Requirement Lists

One traditional way of documenting requirements has been contract style requirement lists. In a complex system such requirements lists can run to hundreds of pages.

#### Measurable Goals

Best practices take the composed list of requirements merely as clues and repeatedly ask “why?” Until the actual business purposes are discovered. Stakeholders and developers can then devise tests to measure what level of each goal has been achieved thus far. Such goals change more slowly than the long list of specific but unmeasured requirements. Once a small set of critical, measured goals has been established, rapid prototyping and short iterative development phases may proceed to deliver actual stakeholder value long before the project is half over.

## Prototypes

In the mid-1980s, prototyping was seen as the solution to the requirements analysis problem. Prototypes are mock-ups of an application. Mock-ups allow users to visualize an application that hasn't yet been constructed. Prototypes help users get an idea of what the system will look like, and make it easier for users to make design decisions without waiting for the system to be built. Major improvements in communication between users and developers were often seen with the introduction of prototypes. Early views of applications led to fewer changes later and hence reduced overall costs considerably.

However, over the next decade, while proving a useful technique, prototyping did not solve the requirements problem:

- Managers, once they see a prototype, may have a hard time understanding that the finished design will not be produced for some time.
- Designers often feel compelled to use patched together prototype code in the real system, because they are afraid to 'waste time' starting again.
- Prototypes principally help with design decisions and user interface design. However, they can't tell you what the requirements originally were.
- Designers and end users can focus too much on user interface design and too little on producing a system that serves the business process.

Prototypes can be flat diagrams (referred to as 'wireframes') or working applications using synthesized functionality. Wireframes are made in a variety of graphic design documents, and often remove all color from the software design (i.e. use a greyscale color palette) in instances where the final software is expected to have graphic design applied to it. This helps to prevent confusion over the final visual look and feel of the application.

***Uses of System Prototypes***

- The major utilization of system prototypes is to assist consumers and developers recognize the requirements for the system
    - *Requirements elicitation:* Users can research with a prototype to observe how the system assists their work
    - *Requirements validation:* The prototype can disclose errors and blunders in the requirements
- Prototyping can be regarded as a risk lessening activity which decreases requirements risks.

***Prototyping Benefits***

The benefits of prototyping are given as below:

- Misinterpretations among software users and developers are depicted
- Missing services may be identified and perplexing services may be recognized
- A working system is obtainable early in the procedure
- The prototype may provide as a foundation for obtaining a system specification
- The system can sustain user training and system testing

Notes

Prototyping leads to the following:

- Enhanced system usability
- Closer match to the system required
- Enhanced design quality
- Enhanced maintainability
- Abridged on the whole development attempt

***Prototyping in the software process:*** There are two types of prototyping used in the software process.

- ***Evolutionary prototyping:*** A method to system development where an initial prototype is generated and developed via a number of stages to the concluding system.
- ***Throw-away prototyping:*** A prototype which is generally a practical implementation of the system is generated to aid find out requirements problems and then not needed. The system is then produced by means of some other enlargement process.

The purpose of evolutionary prototyping is to provide a working system to end-users. The development begins with those requirements which are best recognized. Evolutionary prototyping must be utilized for systems where the specification cannot be produced in advance such as AI systems and user interface systems. Evolutionary prototyping is based on techniques which permit rapid system iterations. Verification is impracticable as there is no specification. Validation specifies illustrating the competence of the system

In case of Evolutionary prototyping:

- Specification, design and implementation are entangled
- The system is produced as a series of increment that are provided to the customer
- Methods for rapid system development are used like CASE tools and 4GLs
- User interfaces are generally produced by means of a GUI development tool kit.

The purpose of throw-away prototyping is to authenticate or obtain the system requirements. The prototyping process begins with those requirements which are unsuccessfully recognized. Throw-away prototyping is used to decrease requirements risk. The prototype is produced from an initial specification, provided for experiment then discarded.

The throw-away prototype should not be regarded as a final system because:

- Some system traits may have been left out
- There is no specification for enduring preservation
- The system will be badly structured and complicated to preserve.

### *Use Cases*

A use case is a technique for documenting the potential requirements of a new system or software change. Each use case provides one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific business goal. Use cases typically avoid technical jargon, preferring instead the language of the end user or domain expert. Use cases are often co-authored by requirements engineers and stake holders.

Use cases are deceptively simple tools for describing the behavior of software or systems. A use case contains a textual description of all of the ways which the intended users could work with the software or system. Use cases do not describe any internal workings of the system, nor do they explain how that system will be implemented. They simply show the steps that a user follows to perform a task. All the ways that users interact with a system can be described in this manner.

During the 1990s, use cases rapidly became the most common practice for capturing functional requirements. This is especially the case within the object-oriented community, where they originated, but their applicability is not restricted to object-oriented systems, because use cases are not object-oriented in nature.

From a traditional software engineering perspective, a use case describes just one feature of the system. For most software projects, this means that perhaps tens or sometimes hundreds of use cases are needed to fully specify the new system. The degree of formality of a particular software project and the stage of the project will influence the level of detail required in each use case.

A use case defines interactions between external actors and the system under consideration, to accomplish a business goal. Actors are parties outside the system that interact with the system; an actor can be a class of users, a role users can play, or another system.

Use cases treat the system as a black box, and the interactions with the system, including system responses, are perceived as from outside the system. This is deliberate policy, because it simplifies the description of requirements and avoids the trap of making assumptions about how this functionality will be accomplished.

A use case should:

- describe a business task to serve a business goal
- be at an appropriate level of detail
- be short enough to implement by one software developer in a single release

*Caution* Use cases can be very good for establishing functional requirements, but they are not suited to capturing non-functional requirements.

*Did u know?* Each use case focuses on describing how to achieve a single business goal or task.

*Task* Depict the functions of use cases.

## Self Assessment

Fill in the blanks:

1. ........................................ determines whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.

2. ........................................ may reveal requirements not previously envisaged as being within the scope of the project, and requirements may be contradictory.

3. ........................................ help users get an idea of what the system will look like, and make it easier for users to make design decisions without waiting for the system to be built.

Notes

4. A ........................................ contains a textual description of all of the ways which the intended users could work with the software or system.

5. Use cases treat the system as a........................................, and the interactions with the system, including system responses, are perceived as from outside the system.

6. Use cases can be very good for establishing functional requirements, but they are not suited to capturing ........................................ requirements.

## 4.2 Requirements Determination

It is helpful to view requirements determination through the three major activities of requirements which are discussed below:

### 4.2.1 Activities in Requirements Analysis

Activities involved in requirement analysis which assists in viewing requirements determination are:

- Requirements anticipation
- Requirements investigation
- Requirements specification

**Requirements Anticipation**

This activity is basically foreseeing system characteristics based on previous experiences. The system analyst, due to prior experience of studying a similar system, may foresee the requirements. He may also foresee certain problems or features and requirements for a new system. As a result, the features they investigate for the current system, questions that they raise, or methods employed may be based on this familiarity.

Requirements anticipation can be a mixed blessing. On the one hand, experience from previous studies can lead to investigation of areas that would otherwise go unnoticed by an unexperienced analyst. Having the background can help one know what to ask or which aspects of investigation can be beneficial to the organization. On the other hand, if a bias is introduced or short cuts are taken in investigations, requirements anticipation is a problem.

**Requirements Investigation**

This activity is at the heart of system analysis. Using a variety of tools and skills, analysts study the current system and document its features for further analysis.

Requirements investigation relies on fact finding techniques discussed later and includes methods for documenting and describing the system features.

**Requirements Specification**

The data produced during the fact finding study are analyzed to determine requirements specifications – the description of features for a new system. This activity has three interrelated parts:

(i) *Analysis of Factual Data:* The data collected during the fact finding study and included in data flow and decision analysis documentation are examined to determine how well the system is performing and whether it will meet the organization's demands.

(ii) *Identification of Essential Requirements:* Features that must be included in a new system, ranging from operation details to performance criteria, are specified.

(iii) ***Selection of Requirements Strategies:*** The methods that will be used to achieve the stated requirements are selected. These form the basis for system design, which follows requirements specifications.

*Did u know?* Requirements specification places a great deal of responsibility on the system analyst for the quality of work performed.

*Caution* All three activities are important and must be performed correctly.

## 4.2.2 Basic Requirements

Analysis structures their investigation by seeking answer to these four major questions:

- What is the basic business process?
- What data are used or produced during the process?
- What are the limits imposed by time and volume of work?
- What performance controls are used?

### Understand the Process

Begin with the basic. Analysts must raise questions that, when answered, will provide a background of fundamental details about the system and describe it. Asking the following questions will help acquire the necessary understanding:

- What is the purpose of this business activity? (Objective)
- What steps are performed?
- Where are they performed?
- Who performs them?
- How long does this take?
- How often is it done?
- Who uses the resulting information?

Suppose you are investing an inventory recording system, something about which you know very little. You ask all the above questions with reference to the inventory recording system. Answers to these questions provide a broad understanding of what inventory recording is all about and show that the objective of inventory recording is more than just buying stock. But there is not yet enough information to fully understand the system.

### *Identify Data Used and Information Produced*

Analysis next needs to find out what data are used to perform each activity.

*Example:* To reorder inventory, the buyer might require data about quantity in hand, expected demand for the item, supplier name and item cost.

Most business transactions also produce the information that is useful to managers when they evaluate employee, business and system performance and that may be useful in another context to both manager and analyst.

*Example:* Inquiring analysts will find out data about inventory reordering and stocking also provide information about warehouse demands, purchasing practices, sales and cash flow.

### *Determining Process Timing and Volume*

The frequency of business activities varies greatly.

*Example:* Some activities, such as paying taxes, occur only few times in a year, paying salaries to employees is monthly and paying to supplier occurs whenever purchases are made. Therefore, analysts must learn how often an activity is repeated.

Knowing whether an activity occurs frequently may lead an analyst to raise many additional and important questions to determine the reason for the frequency and its effect on business activities.

Many times the easiest way to get this information is to identify the reason for the activity: what causes the activity to be performed? Analysts refer to direct cause of activities as the trigger function. Activities can be triggered by customers (through call, letters, orders, etc.) by events (completion of an application to open a new bank account) and by the passage of time.

Some activities, such as completing a purchase requisition, take only a few seconds. Others, such as deciding whether to accept a merger offer, occur infrequently but take lot of time. Time alone does not determine the importance of an activity, but it does affect the way analysts evaluate certain steps in carrying out the performance.

*Notes* The volume of items to be handled may increase the amount of time needed to complete the activity. The sheer quantity of items making up an activity can produce special problems for the analyst to study, even though the activity occurs infrequently.

### *Identify Controls*

In business situations there are well controlled either by management or process monitoring, determining whether an activity has been performed properly may be no problem. But during the analysis stage, the analyst must examine control methods: are there specific performance standards? Who compares performance against standards? How are mistakes caught? How are errors handled? Are the errors excessive? Weak or missing controls are an important discovery in any systems investigation.

## Self Assessment

Fill in the blanks:

7. ........................................ activity is basically foreseeing system characteristics based on previous experiences.

8. In case of ........................................ activity, using a variety of tools and skills, analysts study the current system and document its features for further analysis.

9. ........................................ places a great deal of responsibility on the system analyst for the quality of work performed.

10. Knowing whether an activity occurs frequently may lead an ........................................ to raise many additional and important questions to determine the reason for the frequency and its effect on business activities.

## 4.3 Types of Requirements

1. ***Functional requirement:*** It points to the statements of services that the system should offer, how the system should respond to specific inputs and how the system should perform in specific situation.

   Functional requirements essentially:

   - Illustrate functionality or system services
   - Depend on the category of software, predictable users and the sort of system where the software is used
   - Functional user requirements may be elevated statements of what the system should do; functional system requirements should illustrate the system services detail.

*Example:*

- The consumer shall be able to investigate either all of the preliminary set of databases or choose a subset from it
- The system shall offer suitable viewers for the user to read documents in the document store
- Each order shall be assigned a unique identifier (ORDER ID) which the user shall be able to copy to the account's permanent storage region.

2. ***Nonfunctional requirement:*** A property or eminence the system must have

   ❖ Performance

   ❖ Security

   ❖ Costs

   Restrictions on the services or functions provided by the system like timing constraints, restrictions on the development process, standards, etc.

   A non-functional requirement includes:

   ❖ Product requirements

   ➢ Requirements which state that the delivered product must perform in a specific manner, e.g. execution speed, reliability etc.

   ❖ Organizational requirements

   ➢ Requirements which are a result of organizational plans and procedures, e.g. process standards used implementation requirements etc.

   ❖ External requirements

   ➢ Requirements which happen from factors which are external to the system and its growth process, e.g. interoperability requirements, governmental requirements etc.

Notes

*Task* Make distinction between functional requirements and non-functional requirements.

## Self Assessment

Fill in the blanks:

11. ........................................ requirement points to the statements of services that the system should offer, how the system should respond to specific inputs and how the system should perform in specific situation.

12. Functional requirement depend on the category of software, predictable users and the sort of system where the ........................................ is used.

13. ........................................ requirements are the requirements which state that the delivered product must perform in a specific manner.

14. ........................................ requirements are the requirements which are a result of organizational plans and procedures.

15. ........................................ requirements are the requirements which happen from factors which are external to the system and its growth process.

## 4.4 Summary

- There are also special kinds of requirements that arise, depending on whether the system is transaction or decision oriented and whether the system cuts across several departments.
- Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved.
- Stakeholder interviews are a common method used in requirement analysis. These interviews may reveal requirements not previously envisaged as being within the scope of the project, and requirements may be contradictory.
- Requirements often have cross-functional implications that are unknown to individual stakeholders and often missed or incompletely defined during stakeholder interviews.
- Requirements Anticipation activity is basically foreseeing system characteristics based on previous experiences.
- Requirements Investigation activity is at the heart of system analysis. Using a variety of tools and skills, analysts study the current system and document its features for further analysis.
- Functional requirement points to the statements of services that the system should offer, how the system should respond to specific inputs and how the system should perform in specific situation.
- Product requirements are the requirements which state that the delivered product must perform in a specific manner.
- Organizational requirements are the requirements which are a result of organizational plans and procedures.
- External requirements are the requirements which happen from factors which are external to the system and its growth process.

## 4.5 Keywords

***Analyzing Requirements:*** Analyzing requirements determines whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues.

***Eliciting Requirements:*** Eliciting requirements are the task of communicating with customers and users to determine what their requirements are.

***External Requirements:*** These are the requirements which happen from factors which are external to the system and its growth process.

***Functional Requirement:*** It points to the statements of services that the system should offer, how the system should respond to specific inputs and how the system should perform in specific situation.

***Organizational Requirements:*** These are the requirements which are a result of organizational plans and procedures.

***Product Requirements:*** These are the requirements which state that the delivered product must perform in a specific manner.

***Prototype:*** A working system that is developed to test ideas and assumptions about the new system.

***Recording Requirements:*** Recording Requirements may be documented in various forms, such as natural-language documents, use cases, user stories, or process specifications.

***Use Case:*** A use case is a technique for documenting the potential requirements of a new system or software change.

## 4.6 Review Questions

1. How will you analyze the requirements of an Information System?
2. What do you mean by requirement analysis? Illustrate.
3. List the activities associated with the requirement analysis.
4. Describe various tools and techniques of requirement analysis.
5. How has prototyping become the solution to the requirement analysis?
6. Explain the concept of documenting requirements in contract style requirement lists.
7. Use cases are deceptively simple tools for describing the behavior of software or system. Comment.
8. Make distinction between Requirements anticipation and Requirements investigation.
9. Discuss how to determine Process Timing and Volume.
10. Enlighten the different types of requirements. Give examples.

### Answers: Self Assessment

1. Analyzing requirements
2. Stakeholder interviews
3. Prototypes
4. use case
5. black box
6. non-functional
7. Requirements Anticipation
8. Requirements Investigation

**Notes**

9. Requirements specification
10. analyst
11. Functional
12. software
13. Product
14. Organizational
15. External

## 4.7 Further Readings

*Books*

Charles S. Wasson, *System Analysis, Design, and Development: Concepts, Principles and Practices*, 2006, Wiley-Interscience.

Dorothy J. Tudor; Ian J. Tudor, *System Analysis and Design: A Comparison of Structured Methods*, 1995, NCC Blackwell.

Elias M. Awad, *Business Data Processing,* 1980, Prentice Hall (Higher Education Division, Pearson Education).

Elias M. Awad, *Systems Analysis and Design*, 1985, R.D. Irwin.

Gary B. Shelly; Thomas J. Cashman, Harry J. Rosenblatt, *System Analysis and Design*, 2001, Course Technology.

Jeffrey L. Whitten; Lonnie D. Bentley; Kevin C. Dittman, *Systems Analysis and Design Methods*, 2003, McGraw-Hill.

John W. Satzinger, Jackson, Burd, *System Analysis & Design*, 2006, Thomson Learning.

Joseph S. Valacich; Joey F. George; Jeffrey A. Hoffer, *Essentials of System Analysis and Design*, 2005, Prentice Hall.

Leonard Fertuck, *System Analysis and Design*, 1994, McGraw-Hill Higher Education.

Penny A. Kendall, *Introduction to System Analysis and Design: A Structured Approach*, 1996, Irwin.

Preeti Gupta, *System Analysis and Design*, 2005, Firewall Media.

Ronald G Noice, *System Analysis & Design*, 1984, Oxford University Press Canadian Branch.

*Online links*

https://www.alibre.com/assets/systemrequirements.pdf

www.irritrol.com/pccontrol/docs/pccontrol_sysreqs.pdf

# Unit 5: Fact Finding Techniques

Notes

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the concept of fact finding
- Discuss the various fact finding techniques
- Explain the interview, questionnaire, record review and observation

## Introduction

Fact finding means learning as much as possible about the present system. This is carrying out a highly detailed review of the work being done at the moment and the system (methods) involved.

The Philosophy of fact finding includes:

(a) Finding out the hierarchical structure of the client firm and methods of working within it;

(b) Moving an open mind at all times (People's view of their work may not be correct).

In this unit, you will understand various fact finding techniques such as Interview, Questionnaire, record review, and observation.

## 5.1 Fact Finding

Fact finding occurs during all the following phases:

1. ***Survey Phase:*** The analyst collects general facts about the problems, opportunities, directives, environment, end users and so forth.

2. ***Study Phase:*** The analyst collects general fact about how the current system functions, and about problems and opportunities. This phase normally requires the most fact finding technique.

3. ***Definition Phase:*** The analyst collects facts about the end-users's requirements and expectations. This phase involves the second most extensive fact finding process.

4. ***Selection Phase:*** Analyst collects facts about candidate solutions, end user's opinions about candidates and costs and benefits.

5. ***Fact Recording:*** Unless the investigator has formulated a plan for recording the notes of the facts obtained during the fact finding stage, he or she would end up with a mass of notes on all areas, which would be difficult to examine.

A good practice is to sectionalize notes into areas of investigation (e.g., by department or the operation) or by type of information (e.g., organisation charts, interviews, forms etc.). Standard forms of presenting information about the system must be used. Data flow diagrams can be used to represent information about an existing system as well as a proposed new system. EAR data models can often be drafted at the fact finding stage too. Later they can be amended and redrawn as the results are analyzed. Some traditional charting techniques are also vital here.

*Example:* Decision tables and system flow charts are good examples of techniques, which could capture the details of how the existing system works.

*Caution* At this stage, no attempt is made for analysis or design since the existing system must be fully understood first.

A further difficulty in fact finding is that it is essentially looking for two kinds of fact. The first type exists in the current system and typically is discovered on present document and in existing procedures. In contrast, the second type of fact covers such areas as "Information needs beyond the present system, management reporting requirements and management information."

*Did u know?* It may be difficult to uncover all the intricacies of the current system but at least, they actually exist and can be recognized by the users and operators.

*Notes* In some instances, these may be easy to identify but in most of the circumstances, it is impossible for users to envisage what might be useful or feasible before receiving the information.

*Task* Make distinction between survey phase and study phase.

## Self Assessment

Fill in the blanks:

1. In .............................. Phase, the analyst collects general fact about how the current system functions, and about problems and opportunities.

2. Analyst collects facts about candidate solutions, end user's opinions about candidates and costs and benefits during ............................... phase.

## 5.2 Fact Finding Techniques

It is essential to gather all the facts about a current system to ensure that all the strengths and weaknesses are discovered. Thus, when a new system is designed, as many of the weaknesses as possible are eliminated while retaining the strengths.

Effective fact finding techniques are crucial to the application of systems analysis and design methods during system project. Fact finding is performed during the study, definition, evaluation, design and implementation phases of the system development life cycle. To support development activities, the analyst must collect facts about end-users, the business, data and information resources, and the components of information system in vague. The common fact finding techniques are as follows:

1. Sampling
2. Research
3. Interview
4. Questionnaire
5. Record Review
6. Observation:
   (a) Background Reading
   (b) Measuring

### 5.2.1 Sampling

The sampling of existing documents and files can provide many facts and details with little or no direct personal communication being necessary. The analyst should collect historical documents, business operations manuals and forms, and information systems documents. Sampling techniques are used by the analyst to ensure that an adequate number of documents has been studied. These techniques make it possible to collect a representative sub-set of the documents and minimize the change in the identification of exceptional events.

### 5.2.2 Research

Research is an often over looked technique based on the study of other similar applications. Site visits are a special form of research.

The data that the analyst collects during preliminary investigations are gathered though two primary methods:

- Reviewing documents.
- Interviewing selected company personnel.

**Reviewing Organization Document**

The analyst conducting the investigation first term about the organization involved in, or affected by, the project.

Notes

*Example:* To review an inventory system proposal means knowing first how the inventory department operates and who the managers and supervisors are. Analyst can usually learn these details by examining organization charts and studying written operating procedures. The procedures describe how the inventory process should operate and identify the most important steps involved in receiving, managing and dispensing stocks.

### Examination of Current System

We ought to understand the existing system, manual or computerized, before we design and build a new system. Therefore, we now study and analyze the current system. Many times, the initial problems were mere symptoms, frequently of more serious or subtle problems. During the study phase, were would like to address the causes and effects of problems, opportunities, and directives.

The finding of the study phase are documented as some form of problem statement for the next phase of the life cycle. This problem system may either be a formal report or, an updated feasibility assessment. Based on the findings, the end-users can either cancel the project or approve the movement towards the next phase.

### Ideas for Improvement

There is always a room for the consideration of people's ideas. Provided these are clearly recognized as such and separated from mere opinions good ideas make a valuable contribution towards the design of a new system. The analyst should, therefore, be on the lookout for well founded ideas and encourage the staff to put forward their suggestions during the investigation stage.

### Checklists

It would be convenient to have a comprehensive checklist showing all facts to be collected and questions to be asked during the course of a systems investigation and to merely fill in the answers. This method requires the analyst to have in depth knowledge of the applications in questions. Within a given situation a sample checklist acts as a useful aide-memory, especially for the less experienced system analyst.

Other techniques are discussed in the sections below.

## 5.2.3 Interviews

Interviews are formal meetings with current users of the existing system and potential users of the proposed system. The users may be managers or other employees of the organization, who can provide information about the working of the existing system and can explain the drawbacks of the current system. Although, interviews is a time consuming method, it is the best technique for getting the qualitative information (i.e., problems of existing system, different opinions about expectations from the new system, etc.).

Interviews must be conducted in a planned way. They are carried out by the systems analyst in following steps:

(i) First of all, the analyst decides the objective or purpose of the interview.

(ii) The analyst identifies the users to be interviewed.

(iii) He/she establishes a mutually convenient time, duration and place for meeting with each user in advance.

(iv) The analyst obtains the appropriate manager's consent before interviewing staff.

*Example:* In the case study on 'Stock Monitoring System', the users that are to be interviewed are:

(a) Store Clerk

(b) Production Supervisor

(c) Store Manager

(d) Production Manager

During interviews, the analyst asks various questions to the different users. Some of them are:

(a) Store Clerk

  (i) Which of the system are you using for maintaining the stock - Manual or Computerised?

  (ii) Which documents do you maintain for keeping information regarding various parts?

  (iii) Which documents do you maintain for issuing, receiving and returning various parts?

(b) Production Supervisor

  (i) How do you get parts from the store? Which document do you use to request the store clerk to issue the part?

  (ii) Which document do you use to return the unused part to the store?

(c) Store Manager

  (i) How do you know the current stock status of various items in the store?

  (ii) Do you maintain any documents for keeping optimum level of stock of various items? If yes, do they provide you the required information?

(d) Production Manager

  (i) Which documents do you maintain for keeping records of the parts used in various stages of production?

  (ii) How many parts do you require daily for production?

Interviews could be structured or unstructured. In structured interviews, the analyst asks standard questions with open response (questions are answered in user's own words) or closed response (a set of answers are provided with questions) format. In unstructured interviews, any relevant question about the system may be asked by the analyst.

### 5.2.4 Questionnaire

Questionnaire is a structured and formal method of collecting data. It is mainly used, when there is a scattered group of users and it is not possible to conduct interviews of all the users. It allows the analyst to obtain information from a large number of persons without conducting interviews.

Questionnaire could be open-ended or closed. In open-ended questionnaire, questions are given without any choice of answers and the user has to write the answers in own words on the space given. In closed questionnaire, a choice of answers is given with each question and the user has just to mark the correct response. The analyst may also makes a questionnaire with both open-ended and closed questions as illustrated in Exhibit 5.1.

Questionnaire offers the following advantages:

- Less expensive than conducting interviews (if users are scattered in many branches);
- Time saving as compared to other techniques;
- Provides qualitative information like detailed factual data;
- More useful to ask personal and controversial questions;
- Answers can be compared with documents.

Although, questionnaires have many advantages, still they are less used by analysts due to their following drawbacks:

- Incorrect conclusions may be drawn if response rate is low;
- Questions may be interpreted differently by different users;
- No discussion can be made on controversial answers;
- Real work situations cannot be observed.

**Exhibit 5.1: An Example of Questionnaire with Both Open-ended and Close-ended Questions**

**B.R. Auto Limited**

Questionnaire for Stock Monitoring System

Name of the user _____________ Department _____________

Designation_____________ Working under _____________

Nature of work _____________ Date _____________

Answer all the questions in detail. Use extra sheets if required:

Q 1. Which system are you using- (a) Manual [ ] (b) Computerised [ ] (c) Both [ ]

Q 2. Are you satisfied with the system (a) Yes [ ] (b) No [ ] (c) Partially Satisfied [ ]

Q 3. What is your major problem?

_______________________________________________

_______________________________________________________________________

_______________________________________________________________________

_______________________________________________________________________

Q 4. Name the documents you maintain for various purposes.

| S. No. | Document | Purpose | Frequency (Daily/Monthly) |
|---|---|---|---|
| | | | |

Contd.....

Q 5. Volume of Data for various transactions.

| S. No. | Transaction Name | No. of transaction per day |
|---|---|---|
| 1. | Item Received | |
| 2. | Items Issued | |
| 3. | Items Returned | |

Q 6. Do you get the stock status report immediately when required? If no, why it is delayed?

*Task* Make distinction between open-ended questionnaire and closed questionnaire.

### 5.2.5 Record Review

Organizations usually have a large volume of documents in the form of organization charts, administrative procedure manuals, standard operation procedures, policy manuals, job description documents, account books, sales analysis reports, etc., that can provide a valuable information to the analyst. By inspecting these documents, the analyst can easily understand the working of the current system. Record inspection is the best source of quantitative information. It is carried out by the systems analyst in the following steps:

(i) First, the analyst visits the concerned department and meet the managers and users.

(ii) He enquires various users about the documents they maintain.

(iii) He identifies and prepares the complete list of documents that are relevant to the current and proposed system.

(iv) He examines each and every document to understand the working of the system.

(v) In the last, the analyst collects the copies of all the relevant documents.

Each document would have its own cycle of creation, amendment use to deletion, relevant question might include the following:

- What event initially triggers the generation of the document?
- What generates the document?
- How is it prepared?
- Where is the data derived from?
- Who use the documents?
- For what purpose, is it used?
- How is it stored?
- How long is it kept?

*Example:* Consider the case study on ‘Stock Monitoring System’. In this system, the following documents are identified by the analyst:

(a) Part Register

(b) Requisition Slips

(c) Daily Transaction Register

The analyst studies these documents and understands the complete procedure to prepare the stock status report. He also receives information about the number of transactions occurred in a period (day/month/year) during last year.

> *Notes* The study of the organizational charts, procedure manuals and statics can reveal much usual information about a procedure. However, a close study of the forms currently being used should give the best guide to current practice, which may, or may not, be in tune with original requirements.

### 5.2.6 Observation

Observation is a fact finding technique in which the analyst studies people during their jobs. To minimize the chance that the observation time is not representative of normal work loads, the analyst can use work sampling to collect observation data randomly. This is best employed in conjunction with other techniques and carried out after the observer has an understanding of the procedures involved. Only then would he or she be able to spot irregularities.

Observing entails watching the department staff carrying out their various tasks in normal ways. It is a time consuming activity and therefore not to be indulged in without a definite purpose. As a rule, people do not take kindly to being observed at their work, so this is one of the system analyst's more delicate tasks.

The aspects of a department's work revealed by observation are as follows:

a. ***Interruptions to the normal flow of work:*** This are caused by callers from other departments, telephone calls received and made and visitors from outside the company.

b. ***Informal communication of information:*** This is among the members of the department, callers and visitors and over the telephone. Since no paperwork is involved, this type of information flow might go undetected if not observed. Informal communication is more usual in small companies because it is easier to communicate orally in more compact environment.

c. ***The usage of flies of documents:*** We should also include the non-routine reference to file documents in order to handle queries, often received by telephone. As aspects of this activity is the disruption caused by staff borrowing file documents and replacing them immediately.

d. ***The balance of the work load:*** This applies to the different times of the day or week and between the various members of the department.

e. ***Operational Inefficiencies:*** The observable factors here are bad working condition, machines and equipment in poor condition, absence of authority or leadership and insufficient understanding of the procedures by the staff members.

#### Background Reading

Organization usually aims as a considerable of documentary evidence and this can provide the analyst with an important insight into current organizational norms. This evidence may be available in many forms, though it must be recognized that smaller companies may be less formalized or documented because of time and resource pressures. Reports of previous marked surveys or feasibility studies may be out dated but worth reading to understanding the

Notes

background of the current study. Company information may be available in the following forms:

- Organizational charts.
- Administrative procedure manuals.
- Job descriptions and specifications.
- Training manuals and memoranda.
- Sales and promotional literature.

The strategic plan and its associated corporate appraisal would be significant source of relevant information.

While conducting interviews, the staff of the study group should seek to discover the information and data processing requirement not satisfied by the current system and to find out what information is needed. In this phase, it should be recognized that many persons do not know what they need or can use. Some analysis is then required to help the user perceive what information would be useful.

Documents are very important because they represent the formal information flows of the present system. During discussions, the documents used by participants would be referred to, examined and explained. The analyst should collect specimens of all these relevant documents input forms, output analyses, reports, invoices etc. In an attempt to understand how date are parsed and used in the present system. Bank forms and copies of completed forms used in the organization should be collected.

### Measurement

When facts are unobtainable through other methods or when their accuracy is suspect, measurement or estimation is employed. It is not a method recommended for general use since it absorbs considerable time and demands great care. Measured factors include quantities, times, intervals and rates. The tools of measure are counters (mechanical or electronic), rules, clocks, stop watches, pocket calculators, scale and last but not least, the human brain. In most, cases, measurement yields an approximate figure but accuracy is acceptable for its purpose.

Observation of a system in real work environment is necessary for collecting valuable information about how different processes are being carried out.

*Did u know?* Even though, observation is a time-consuming method, it is the best technique to understand the system as a real world problem.

*Example:* For observing the 'Stock Monitoring System' of B. R. Auto Limited, the systems analyst visits various departments and observes the different processes. He prepares the observation report as illustrated in Table 5.1.

*Caution* The analyst must observe the normal levels of supervision and control, the different stages and place of work and bottlenecks in the workflow.

Notes

Table 5.1: Observation Chart of 'Stock Monitoring System'

| Department | Processes |
|---|---|
| Store | 1. Various parts required for manufacturing are stored in store which are received from different vendors. |
| | 2. The material received from vendors is accompanied with a receipt slip. |
| | 3. Store clerk issue the parts to production department an receiving the requisition slip. |
| | 4. Store clerk maintains the part register. |
| | 5. Store clerk prepares various reports. |
| Production | 1. The production supervisor requests store clerk to issue the parts required for manufacturing. |
| | 2. At the end of day, the production supervisor returns the unused parts to the store by filling a return slip. |

## Self Assessment

Fill in the blanks:

3. Effective fact finding techniques are crucial to the application of systems analysis and design methods during ............................... .

4. The............................... of existing documents and files can provide many facts and details with little or no direct personal communication being necessary.

5. ............................... are formal meetings with current users of the existing system and potential users of the proposed system.

6. Although, interview is a time consuming method, it is the best technique for getting the ............................... information.

7. In ............................... interviews, the analyst asks standard questions with open response format.

8. ............................... technique is mainly used, when there is a scattered group of users and it is not possible to conduct interviews of all the users.

9. In ............................... questionnaire, questions are given without any choice of answers and the user has to write the answers in own words on the space given.

10. In ............................... questionnaire, a choice of answers is given with each question and the user has just to mark the correct response.

11. In record review, organizations usually have a large volume of documents in the form of organization charts, administrative procedure manuals, standard operation procedures, etc., that can provide a ............................... information to the analyst.

12. Record inspection is the best source of ............................... information.

13. ............................... is a fact finding technique in which the analyst studies people during their jobs.

14. When facts are unobtainable through other methods or when their accuracy is suspect, ............................... is employed.

15. Observation is the best technique to understand the system as a ............................... problem.

Notes

*Case Study*

## Fact Finding Techniques: Library Management 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 talk to you regarding the functioning of your library.

Librarian: Hello, do come in. I was expecting 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 whole contribution

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 to a particular member.

Analyst: How often you get new members?

Librarian: Very often. At about 50 to 100 members in a month. But for two months we have freezed the membership because it is already very difficult to manage the existing 250

Contd.....

Notes

members. But if this whole system gets computerized 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 50-100 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 400 to 500 for half yearly and 1000 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 categorization for members. All are treated at par.

Analyst: How many books are there?

Librarian: About 5000 books

Analyst: Do you people keep records for them?

Librarian: 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?

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

Analyst: Thanks for your co-operation. 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.

*Contd.....*

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 400 to 500 for half year and 1000 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 4. 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 30 people.

*Contd.....*

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

Member: 25 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 for Library Staff**

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. ______________________________________________________________

    ______________________________________________________________

**Questionnaire for Library Members**

In order to get the views of the existing members, the analyst also distributed questionnaires to the member also.

*Contd.....*

Notes

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 : ________________

Now the analyst has a good understanding of the requirements for the new system, we can move to the designing.

**Question**

How does analyst perform interviews?

***Source:*** http://www.scribd.com/doc/50267807/57/Fact-Finding-Techniques-Case-Study-Library Management-System

## 5.3 Summary

- Fact finding occurs during survey study, definition, selection and facts recording of an organization when a new system is designed, errors are made to eliminate its weaknesses and retain its strengths.
- Fact finding is performed during the study, definition, evaluation, design and implementation phases of the system development life cycle.
- The five common fact finding techniques are sampling, research, observation, questionnaire and Interview.
- By inspecting these documents, the analyst can easily understand the working of the current system.

- The sampling of existing documents and files can provide many facts and details with little or no direct personal communication being necessary.
- Research is an often over looked technique based on the study of other similar applications. Site visits are a special form of research.
- Interviews are formal meetings with current users of the existing system and potential users of the proposed system.
- Questionnaire is a structured and formal method of collecting data. Questionnaire could be open-ended or closed.
- Observation is a fact finding technique in which the analyst studies people during their jobs.
- Observation of a system in real work environment is necessary for collecting valuable information about how different processes are being carried out.

## 5.4 Keywords

***Fact Finding:*** Learning as much as possible about the present system.

***Interviews:*** Formal meetings with current users of the existing system and potential users of the proposed system.

***Observation:*** A fact finding technique in which the analyst studies people during their jobs.

***Questionnaire:*** A structured and formal method of collecting data, used when there is a scattered group of users.

***Research:*** Research is an often over looked technique based on the study of other similar applications. Site visits are a special form of research.

## 5.5 Review Questions

1. What do you mean by fact finding? Depict the various phases of fact finding.
2. Discuss various phases in which fact finding appears.
3. What is the difference between structured and unstructured interviews? Illustrate.
4. Make distinction between open-ended and closed questionnaire
5. Illustrate the observation technique of a system with an example.
6. Prepare a case study for fact findings of any department of a particular organization.
7. Elucidate in detail various techniques of fact finding.
8. Enlighten the steps involved in record inspection of an organization.
9. What are Interviews? How are they conducted?
10. Define questionnaire. What are the advantages and disadvantages of questionnaire?
11. Which is the best technique to understand the system as a real world? Illustrate.

### Answers: Self Assessment

1. study
2. selection
3. system project
4. sampling
5. Interviews
6. qualitative

7. structured
8. Questionnaire
9. open-ended
10. closed
11. valuable
12. quantitative
13. Observation
14. measurement
15. real world

## 5.6 Further Readings

*Books*

D R Jeffery;  M J Lawrence, [Sydney, N.S.W. ; Englewood Cliffs, N.J.], *Systems Analysis and Design,* Prentice-Hall of Australia, 1985, ©1984.

James C Wetherbe, St. Paul, *Systems Analysis and Design*, West Pub. Co., ©1988.

S.A. Kelkar, *Structured System Analysis and Design*, Prentice Hall of India

*Online links*

http://www.itworld2.com/frmfft1.aspx

http://repository.binus.ac.id/content/M0094/M009435141.pdf

Notes

# Unit 6: Tools for Documenting Procedures and Decisions

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Discuss the concept of decision trees and decision tables.
- Explain the structured English

## Introduction

Process descriptions are the tools for documenting the procedures and describing the system logic. They contain the logic used to process the input data for getting the output. The commonly used process description tools are:

(a) ***Structured English:*** used to describe a procedure in simple English statements.

(b) ***Decision Tree:*** used to describe a set of conditions and actions diagrammatically.

(c) ***Decision Table:*** used to describe a set of conditions and actions in a tabular form.

## 6.1 Decision Trees and Decision Tables

Decision tables and trees were developed long before the widespread use of computers. They not only isolate many conditions and possible actions, but also help ensure that nothing has been overlooked.

### 6.1.1 Decision Tables

The decision table is a chart with four sections listing all the logical conditions and actions. In addition the top section permits space for title, date, author, system and comment.

Notes

The condition stub displays all the necessary tests or conditions. Like the diamond in a flowchart or the IF in pseudocode, these tests require yes or no answers. The condition stub always appears in the upper left-hand corner of the decision table, with each condition numbered to allow easy identification.

Thus, condition stub is a list of all the necessary tests in a decision table. In the lower left-hand corner of the decision table we find the action stub where one may note all the processes desired in a given module. Actions, like conditions, receive numbers for identification purposes.

*Did u know?* Action Stub is a list of all the processes involved in a decision table.

The upper right corner provides space for the condition entry – all possible permutations of yes and no responses related to the condition stub. The yes or no possibilities are arranged as a vertical column called rules. Rules are numbered 1, 2, 3 and so on. We can determine the number of rules in a decision table by the formula:

Number of rules = 2^N = 2N

where N represents the number of conditions and ^ means exponent.

Thus, a decision table with four conditions has 16 (2^4 = 2 × 2 × 2 × 2=16) rules one with six conditions has 64 rules and eight conditions yield 256 rules.

Thus, condition entry is a list of all the yes/no permutations in a decision table. The lower right corner holds the action entry. X's or dots indicate whether an action should occur as a consequence of the yes/no entries under condition entry. X's indication action; dots indicate no action.

Thus, Action Entry indicates via dot or X whether something should happen in a decision table.

**Table 6.1**

| | |
|---|---|
| Title:<br>Author:<br>Comments: | Date:<br>System: |
| Condition Stub | Condition Entry |
| Action Stub | Action Entry |

When we build the yes or no rules for the condition entry, we must construct all possible patterns of y's and n's. An arrangement that guarantees thoroughness is to place two y's in succession followed by two n's. In the second row, we place alternating pairs of y's and n's.

*Example:* Decision table for bicycle assembly is shown in Table 6.2

A decision table with four conditions ($2^4$ =16) would have 16 different sets of y's and n's and would result in the following pattern of yes and no responses.

The first row, therefore, will have eight y's followed by eight n's. The second row (corresponding to the second entry in the condition stub) has four y's, four y's and four n's.

Notes

**Table 6.2: Decision Table for Bicycle Assembly**

| TITLE: Bicycle Assembly<br>Author:<br>Comments: More than one carton of parts needs to be assembed | | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 1. Last Carton? | y | y | n | n |
| 2. Hand brakes? | y | n | y | n |
| 1. Open container | - | - | x | x |
| 2. Stack parts | - | - | x | x |
| 3. Assemble whells | - | - | x | x |
| 4. finish assembly | - | - | x | x |
| 5. End of assembly operations | x | x | - | - |

The complete four condition entry would read:

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| y | y | y | y | y | y | y | n | n | n | n | n | n | n |
| y | y | y | y | n | n | n | y | y | y | n | n | n | n |
| y | y | n | n | y | y | n | y | n | n | y | y | n | n |
| y | n | y | n | y | n | n | y | y | n | y | n | y | n |

This form ensures that the analyst includes all combinations with duplication.

If large number of conditions exist (four conditions result in 16 condition entries, six conditions in 64), decision tables can become unwieldy. On occasion, two or more rules may be combined to reduce or eliminate redundancy. In figures 6.3 and 6.4, rules 1 and 2 cause the last action in the action stub to occur.

Therefore, these two rules could be combined to eliminate redundancy. To indicate redundancy, we put a dash (-) in the condition entry to show that this condition stub is irrelevant and can be ignored.

*Caution* To avoid lengthy decision tables, analysts must remove redundancies and yet still take precautions not overlook anything.

The decision table in Table 6.3 depicts the AP cheque module. Although this format is fairly typical, in practice you will encounter several different kinds of decision tables. Table 6.3 called limited entry, because the condition entry contains yes or no responses for each rule.

***Limited Entry:*** A type of decision table listing a y or n response for each condition.

AP cheque decision table:\

**Table 6.3: A Limited Entry Decision Table**

Title : AP Cheque    Date: Sept. 25, 1994

Author :    System: Accounts Payable System

Comments: Two files are to be read until the end of file.

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1. | End of vendor master file? | y | y | y | y | n | n | n | n |
| 2. | End of sorted invoice file? | y | y | n | n | y | y | n | n |
| 3. | Do vendor numbers match? | y | n | y | n | y | n | y | n |
| 1. | Read a vendor master record | . | . | . | . | . | . | . | x |
| 2. | Read an invoice record | . | . | . | . | . | . | x | . |
| 3. | Add amount to total | . | . | . | . | . | . | x | . |
| 4. | Print invoice detail line | . | . | . | . | . | . | x | . |
| 5. | Print data line | . | . | . | . | . | . | . | x |
| 6. | Print amount in words | . | . | . | . | . | . | . | x |
| 7. | Print vendor name/address | . | . | . | . | . | . | . | x |
| 8. | End of module | x | x | . | . | . | . | . | . |

***Extended Entry:*** Type of decision table displaying values to be tested in the condition entry.

AP cheque written as an extended-entry decision table:

**Table 6.4: An Extended-entry Decision Table**

Title : AP Cheque    Date: Sept. 25, 1994

Author :    System: Accounts Payable System

Comments: Two files are to be read until the end of file.

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Vendor master file? | End | End | End | More | More | More | More | More |
| 2. | Sorted invoice file? | End | End | More | End | End | End | More | More |
| 3. | Vendor numbers? | End | More | More | End | End | More | End | More |
| 1. | Read a vendor master record | . | . | . | . | . | . | . | x |
| 2. | Read an invoice record | . | . | . | . | . | . | x | . |
| 3. | Add amount to total | . | . | . | . | . | . | x | . |
| 4. | Print invoice detail line | . | . | . | . | . | . | x | . |
| 5. | Print data line | . | . | . | . | . | . | . | x |
| 6. | Print amount in words | . | . | . | . | . | . | . | x |
| 7. | Print vendor name/address | . | . | . | . | . | . | . | x |
| 8. | End of module | x | x | . | . | . | . | . | . |

***Mixed Entry:*** A type of decision table mixing values in the condition and action entries.

AP cheque written as a mixed-entry decision table: Shown below in Table 6.5

**Table 6.5: A Mixed-entry Decision Table**

Title : AP Cheque                                         Date: Sept. 25, 1994
Author :                                         System: Accounts Payable System
Comments: Two files are to be read until the end of file.

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1. | Is vendor master file at end? | y | y | y | y | n | n | n | n |
| 2. | Sorted invoice file? | End | End | More | End | End | End | More | More |
| 3. | Vendor numbers? | End | More | More | End | End | More | End | More |
| 1. | Read a vendor master record | . | . | . | . | . | . | . | x |
| 2. | Read an invoice record | . | . | . | . | . | . | x | . |
| 3. | Add amount to total | . | . | . | . | . | . | x | . |
| 4. | Print invoice detail line | . | . | . | . | . | . | x | . |
| 5. | Print data line | . | . | . | . | . | . | . | x |
| 6. | Print amount in words | . | . | . | . | . | . | . | x |
| 7. | Print vendor name/address | . | . | . | . | . | . | . | x |
| 8. | End of module | x | x | . | . | . | . | . | . |

**Open ended** (1) A type of decision table that permits access to another decision table. (2) Questionnaire items that respondents must answer in their own words.

A mixed-entry decision table combines the values and yes or no (Table 6.5), while an open-ended one allows an action entry specifying an additional decision table (Table 6.6).

**Table 6.6**

Title : AP Cheque                                         Date: Sept. 25, 1994
Author :                                         System: Accounts Payable System
Comments: Two files are to be read until the end of file.

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 1. | End of vendor master file? | Y | y | y | y | n | n | n | n |
| 2. | End of sorted invoice file? | Y | y | n | n | y | y | n | n |
| 3. | Do vendor numbers match? | Y | n | y | n | y | n | n | n |
| 1. | Read a vendor master record | - | - | - | - | - | - | - | x |
| 2. | Read an invoice record | - | - | - | - | - | - | x | - |
| 3. | Add amount to total | - | - | - | - | - | - | x | - |
| 4. | Print invoice detail line | - | - | - | - | - | - | x | - |
| 5. | Print data line | - | - | - | - | - | - | - | x |
| 6. | Print amount in words | - | - | - | - | - | - | - | x |
| 7. | Print vendor name/address | - | - | - | - | - | - | - | x |
| 8. | End of module | x | x | - | - | - | - | - | - |

An analyst may want to use one of these other types of decision tables to make the table more readable for a user or manager or to decompose a large (seven conditions leading to 128 rules) table into a series of smaller ones.

***AP cheque decision table (open-ended):*** Besides designing screen layout formats and determining screen specification, the design must develop input controls for interactive dialogue and illustrate the way in which screens and menus are linked together. Three tools which help the design team in doing this are dialogue trees, decision trees, and picture-frame analysis.

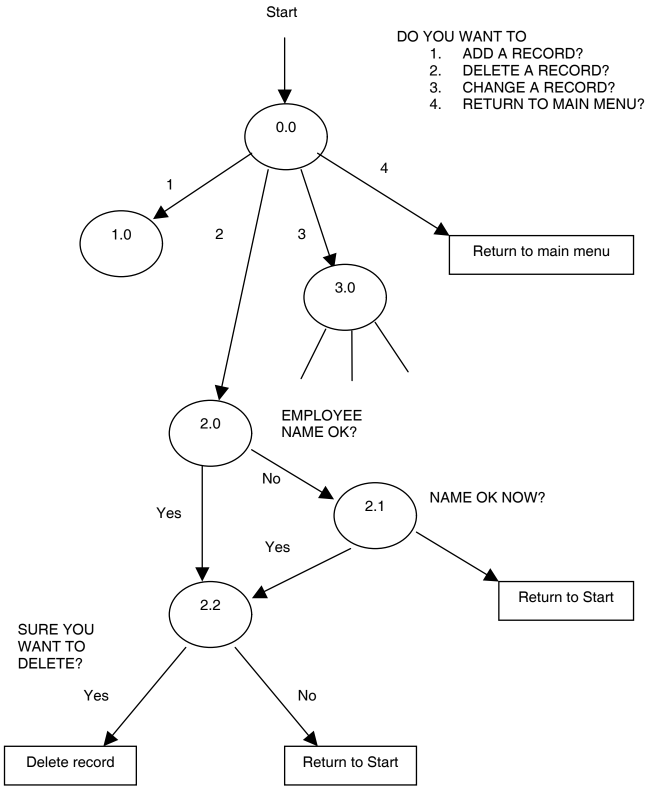
*Notes* With dialogue and decision trees, the team is able to show the flow of control in processing, including the actions users can take to halt or stop an input procedure. With picture-frame analysis, the design team is able to provide a walk-through of how screens will appear once a design becomes operational.

*Task* Make distinction between limited entry and extended entry.

### 6.1.2 Constructing a Dialogue Tree

A dialogue tree maps the static and dynamic messages that take place between the computer and the user. Figure 6.1 shows the design of a tree for a simple file processing menu.

**Figure 6.1: Dialogue Tree Showing Branches from Prompting Menus**

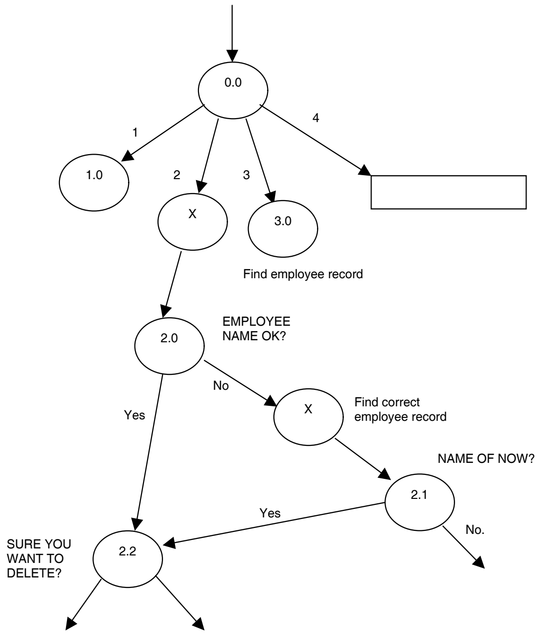


As shown, a dialogue tree has multiple branch points when menus are used, and forks at yes or no points. The dialogue tree should lead you to conclude the following:

1. When an initial response of 2 is received, the program branches to a procedure to DELETE A RECORD from the employee master file.
2. Before a record is deleted, the user is asked to verify that the employee name is correct. The message reads: EMPLOYEE NAME OK?
3. If the name is correct, the tree forks and asks: SURE YOU WANT TO DELETE?
4. If the name is incorrect, the tree forks and asks: NAME OK?
5. If the name is correct (is OK now) and the user responds yes to the question SURE YOU WANT TO DELETE, the record is removed from the file.
6. If the name is not correct, or if the name is correct but the user decides not to delete control is shown as "return to start" -namely, a loop back to the start of the tree.

Isn't this tree incomplete? If the employee name is not correct at node 2.0, how could it be correct at node 2.1? An expanded dialogue tree, like the one shown in Figure 6.2 helps fill in the missing messages. The more detailed tree shown a node with an X. This is a nonrestricted node, meaning that it is not restricted to a prescribed number of choices. The first non restricted node indicates that it is necessary to find an employee record before testing to determine whether the name is correct. Moreover, if a record is found but the name is incorrect, a second attempt (as noted by a second unrestricted node) is made to find the correct employee record. If this second search is successful, the user is asked: NAME OF NOW?

**Figure 6.2: An Expanded Dialogue Tree**



## 6.1.3 Decision Tree

At times, a dialogue tree is too specific for design teams to work with. What they prefer is an easier-to-follow mapping of a complex design. This mapping should show branch points and forks, but not the details of the user dialogue.

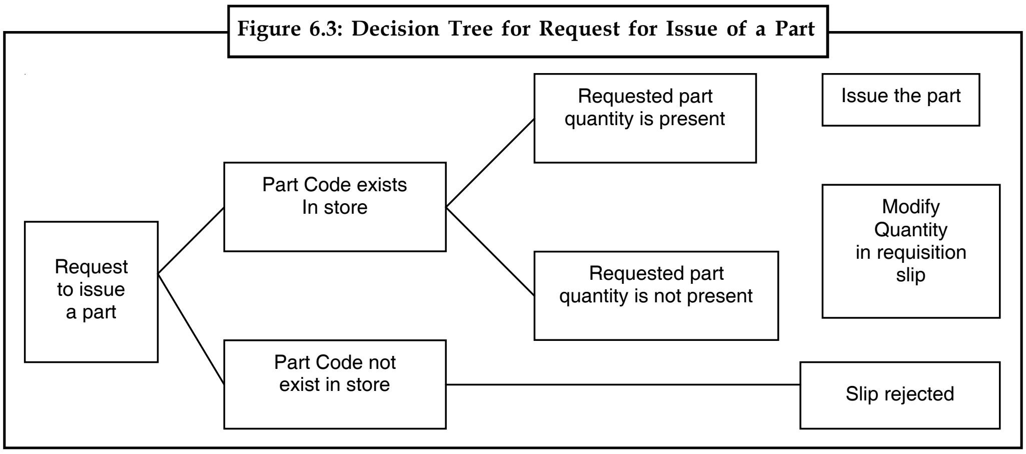
Decision tree is a diagram for showing the alternative actions that can be performed in a process depending upon different set of conditions. The root is on the left side and the actions are shown at the right side. This tree is very easy to construct as illustrated in Figure 6.3.

*Notes* Although, decision tree is an excellent tool for sketching the logical conditions in a process, it is not suitable for complex conditions, due to a large number of branches.

*Caution* For complex conditional statements, decision tables are preferred.

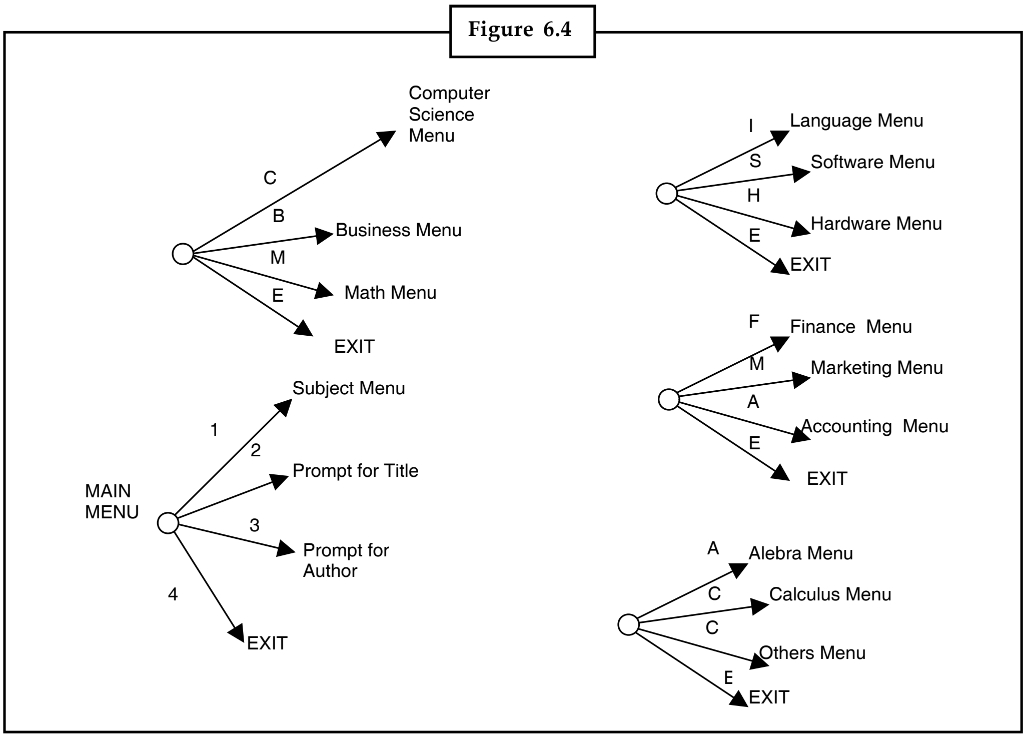
*Did u know?* The decision tree resembles a fallen tree, having a root and many branches.

**Figure 6.3: Decision Tree for Request for Issue of a Part**


Request to issue a part
Part Code exists In store
Part Code not exist in store
Requested part quantity is present
Requested part quantity is not present
Issue the part
Modify Quantity in requisition slip
Slip rejected

A decision tree helps to show the paths that are possible in a design following an action or decision by the user. Figure 6.4 illustrates this type of tree. As indicated, if the user selects 1, followed my M and A, the algebra menu would be displayed.

**Figure 6.4**



**Notes**

*Example:* Figure 6.5 depicts a decision tree for a software-based system, X. In this case, the software engineering organization can

1. Build system X from scratch;
2. Reuse existing "partial experience" components to construct the system;
3. Buy an available software product and modify it to meet local needs; or
4. Contract the software development to an outside vendor.

If the system is to be built from scratch, there is a 70% probability that the job will be difficult. Using the estimation techniques, the project planner projects that a difficult development effort will cost $450,000. A "simple" development effort is estimated to cost $380,000. The expected value for cost, computed along any branch of the decision tree, is:

expected cost = $\sum$(path probability)$_i$ × (estimated path cost)i

where i is the decision tree path. For the build path,

expected cost build = 0.39($380K) + 0.70($450K) = $429K

Following other paths of the decision tree, the projected costs for reuse, purchase, and contract under a variety of circumstances, are also shown. The expected costs for these paths are:

expected cost reuse= 0.40($275K) + 0.60[0.20 ($310K) + 0.80 ($490K)] = $382K

expected cost buy = 0.70($210K) + 0.30 ($400K) = $267K

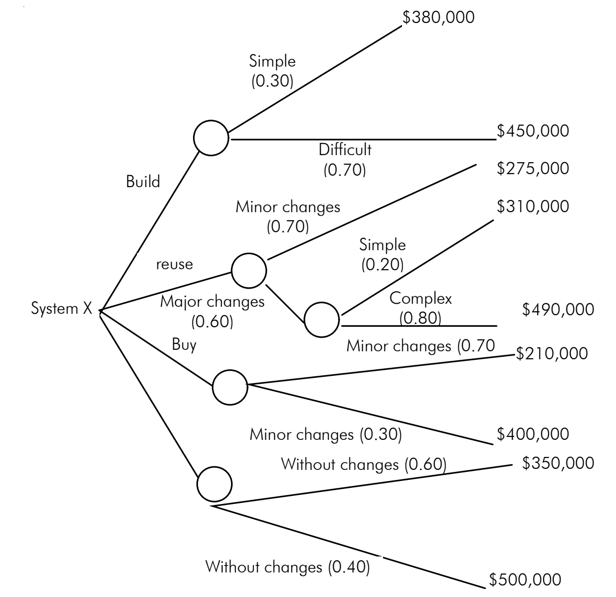
expect cost contract = 0.60($350K) + 0.40 ($ 500K) = $410K

Based on the probability and projected costs have been noted in Figure 6.5, the lowest expected cost is the buy option.

It is important to note, however, that many criteria – not just cost – must be considered during the decision making process.

Availability, experience of the developer/vendor/contractor, conformance to requirements, local "politics", and the likelihood of change are but a few of the criteria that may affect the ultimate decision to build, reuse, buy or contract.

**Figure 6.5: A Decision Tree to Support the Make-buy Decision**



## Self Assessment

Fill in the blanks:

1. The .......................... is a chart with four sections listing all the logical conditions and actions.
2. The ............................ is used to display all the necessary tests or conditions.
3. ............................ is a list of all the yes/no permutations in a decision table.
4. ............................ indicates via dot or X whether something should happen in a decision table.
5. A ............................ decision table combines the values and yes or no, while an open-ended one allows an action entry specifying an additional decision table.
6. A ............................ maps the static and dynamic messages that take place between the computer and the user.
7. An ............................ dialogue tree helps to fill in the missing messages.
8. ............................ is a diagram for showing the alternative actions that can be performed in a process depending upon different set of conditions.
9. A decision tree helps to show the paths that are possible in a design following an action or decision by the............................. .

## 6.2 Structured English

Structured English uses narrative statements to describe a procedure. It uses three basic types of statements:

(a) ***Sequential Structures:*** They include a set of instructions that are carried out one after another and do not depend on any condition.

(b) ***Decision Structures:*** They include one or more sets of instructions that are carried out depending upon one or more conditions. They generally use the phrase IF THEN ELSE to carry out different actions.

(c) ***Iteration Structures:*** They include a set of instructions that are repeated until a particular condition occurs. They generally use the phrase DO WHILE ...ENDDO to repeat a set of instructions.

*Example:* The examples of these three types of statements are illustrated in Table 6.7.

**Table 6.7: Examples of Three Types of Statements**

| Sequential Structure | Decision Structure | Iteration Structure |
|---|---|---|
| Accept employee code<br>Accept employee name<br>Accept other details<br>Store data | If Basic_Pay <=1000<br>HRA = 500<br>else<br>If Basic_Pay <= 3000<br>HRA = 1000<br>else<br>HRA = 1500<br>endif<br>endif | Ans = "Y"<br>Do while Ans = "Y"<br>Accept employee code<br>Accept employee name<br>Accept other details<br>Display "Continue (Y/N)?"<br>Accept Ans<br>enddo |

Notes

*Task* Make distinction between Sequential Structures and Iteration Structures.

## Self Assessment

Fill in the blanks:

10. ............................ uses narrative statements to describe a procedure.

11. ............................ Structures include a set of instructions that are carried out one after another and do not depend on any condition.

12. ............................ Structures include one or more sets of instructions that are carried out depending upon one or more conditions.

13. ............................ Structures include a set of instructions that are repeated until a particular condition occurs.

14. Decision Structures generally use the phrase ............................ to carry out different actions.

15. Iterative structures generally use the phrase ............................ to repeat a set of instructions.

## 6.3 Summary

- Process descriptions are the tools for documenting the procedures and describing the system logic. They contain the logic used to process the input data for getting the output.
- The decision table is a chart with four sections listing all the logical conditions and actions. In addition the top section permits space for title, date, author, system and comment.
- A mixed-entry decision table combines the values and yes or no, while an open-ended one allows an action entry specifying an additional decision table.
- A dialogue tree maps the static and dynamic messages that take place between the computer and the user.
- Decision tree is a diagram for showing the alternative actions that can be performed in a process depending upon different set of conditions.
- The decision tree resembles a fallen tree, having a root and many branches. The root is on the left side and the actions are shown at the right side.
- With dialogue and decision trees, the team is able to show the flow of control in processing, including the actions users can take to halt or stop an input procedure.
- Structured English uses narrative statements to describe a procedure.

## 6.4 Keywords

***Data Dictionary:*** A catalogue of all data elements, data structures and processes described in logical DFDs.

***Data Element:*** The smallest unit of data that has some meaning.

***Data Flow:*** A data structure that shows a unit of data in motion.

***Data Store:*** A data structure for collecting data input during processing.

***Data Structure:*** A group of data elements that describe a unit in the system.

***Decision Table*:** A chart with four sections, listing all the logical conditions and actions.

Notes

***Decision Tree:*** A diagram for showing the alternative actions that can be performed in a process depending upon different set of conditions.

***Dialogue Tree:*** A diagram which maps the static and dynamic messages that take place between the computer and the user.

***Process:*** Actions performed on input data to produce the output data.

***Structured English:*** Structured English uses narrative statements to describe a procedure.

## 6.5 Review Questions

1. What do you mean by process description?
2. What are decision tables? Illustrate the concept.
3. Make distinction between limited, extended, and mixed entry.
4. What is a dialogue tree? Explain the design of a dialogue tree for a simple file processing menu.
5. Illustrate the concept of decision trees with the help of examples.
6. How are decision trees better than decision tables?
7. A mixed-entry decision table combines the values and yes or no, while an open-ended one allows an action entry specifying an additional decision table. Comment.
8. Enlighten the concept of structured English with example.
9. Describe the various types of statements used in structured English.
10. Depict the utility of decision trees.

### Answers: Self Assessment

1. decision table
2. condition stub
3. Condition entry
4. Action Entry
5. mixed-entry
6. dialogue tree
7. expanded
8. Decision tree
9. user
10. Structured English
11. Sequential
12. Decision
13. Iteration
14. IF THEN ELSE

15 DO WHILE ...ENDDO

## 6.6 Further Readings

*Books*

D R Jeffery; M J Lawrence, [Sydney, N.S.W. ; Englewood Cliffs, N.J.]: *Systems Analysis and Design*, Prentice-Hall of Australia, 1985, ©1984.

James C Wetherbe, St. Paul, *Systems Analysis and Design*, West Pub. Co., ©1988.

S.A. Kelkar, Structured *System Analysis and Design*, Prentice Hall of India.

*Online link*

http://www.hit.ac.il/staff/leonidm/information-systems/ch60.html

Notes

# Unit 7: Structured Analysis Development Strategy

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the concept of structured analysis development strategy
- Discuss the features, structures and techniques of structured analysis development strategy
- Identify dataflow tools

## Introduction

Structured Analysis and Structured Design are process centered methodologies. Structured analysis builds process models, called data flow diagrams, for a system. Structured Design transforms the process model for a system into a top-down, process model for the programs that comprise the system.

Structured analysis method focused on what the system or application does rather than on how it is done. It uses graphical symbols to describe movement and processing of data. Important components include data flow diagrams. It is suitable for all type of applications and is useful as a supplement to other development methods.

## 7.1 Structured Analysis Development Strategy

Structured Systems Analysis and Design Method (SSADM) is a systems approach to the analysis and design of information systems. It is a discipline within the software development industry which seek to provide a framework for activity and the capture, storage, transformation and dissemination of information so as to enable the economic development of computer systems that are fit for purpose.

SSADM can be thought to represent a pinnacle of the rigorous document-led approach to system design, and contrasts with more contemporary Rapid Application Development methods such as DSDM.

*Did u know?* SSADM is a waterfall strategy by which an Information System design can be arrived at.

### 7.1.1 Waterfall Model

The waterfall model is a sequential software development model (a process for the creation of software) in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

In the unmodified "waterfall model", progress flows from the top to the bottom, like a waterfall.

In Royce's original waterfall model, the following phases are followed in order:

1. Requirements specification
2. Design
3. Construction (AKA implementation or coding)
4. Integration
5. Testing and debugging (AKA validation)
6. Installation
7. Maintenance

To follow the waterfall model, one proceeds from one phase to the next in a purely sequential manner.

*Example:* One first completes "requirements specification" – they set in stone the requirements of the software.

When the requirements are fully completed, one proceeds to design. The software in question is designed and a "blueprint" is drawn for implementers (coders) to follow – this design should be a plan for implementing the requirements given. When the design is fully completed, an implementation of that design is made by coders. Towards the later stages of this implementation phase, disparate software components produced by different teams are integrated. After the implementation and integration phases are complete, the software product is tested and debugged; any faults introduced in earlier phases are removed here. Then the software product is installed, and later maintained to introduce new functionality and remove bugs.

*Notes* The waterfall model maintains that one should move to a phase only when its preceding phase is completed and perfected. Phases of development in the waterfall model are discrete, and there is no jumping back and forth or overlap between them.

Notes

*Caution* There are various modified waterfall models (including Royce's final model) that may include slight or major variations upon this process.

*Task* What are the different phases of waterfall model?

### 7.1.2 Usage

The waterfall model is widely used, including by such large software development houses as those employed by the US Department of Defense and NASA and upon many large government projects. Those who use such methods do not always formally distinguish between the "pure" waterfall model and the various modified waterfall models, so it can be difficult to discern exactly which models are being used to what extent.

### 7.1.3 Arguments for the Waterfall Model

Time spent early on in software production can lead to greater economy later on in the software lifecycle; that is, it has been shown many times that a bug found in the early stages of the production lifecycle (such as requirements specification or design) is cheaper, in terms of money, effort and time, to fix than the same bug found later on in the process. ([McConnell 1996], p. 72, estimates that "a requirements defect that is left undetected until construction or maintenance will cost 50 to 200 times as much to fix as it would have cost to fix at requirements time.").

*Example:* To take an extreme example, if a program design turns out to be impossible to implement, it is easier to fix the design at the design stage than to realize months later, when program components are being integrated, that all the work done so far has to be scrapped because of a broken design.

This is the central idea behind Big Design Up Front (BDUF) and the waterfall model - time spent early on making sure that requirements and design are absolutely correct will save you much time and effort later. Thus, the thinking of those who follow the waterfall process goes, one should make sure that each phase is 100% complete and absolutely correct before proceeding to the next phase of program creation. Program requirements should be set in stone before design is started (otherwise work put into a design based on "incorrect" requirements is wasted); the program's design should be perfect before people begin work on implementing the design (otherwise they are implementing the "wrong" design and their work is wasted), etc.

A further argument for the waterfall model is that it places emphasis on documentation (such as requirements documents and design documents) as well as source code. In less designed and documented methodologies, should team members leave, much knowledge is lost and may be difficult for a project to recover from. Should a fully working design document be present (as is the intent of Big Design Up Front and the waterfall model) new team members or even entirely new teams should be able to familiarize themselves by reading the documents.

As well as discussed just now, some prefer the waterfall model for its simple and arguably more disciplined approach. Rather than what the waterfall adherent sees as "chaos", the waterfall model provides a structured approach; the model itself progresses linearly through discrete, easily understandable and explainable "phases" and is thus easy to understand; it also provides easily markable "milestones" in the development process. It is perhaps for this reason that the waterfall model is used as a beginning example of a development model in many software engineering texts and courses.

It is argued that the waterfall model and Big Design Up Front in general can be suited to software projects which are stable (especially those projects with unchanging requirements, such as with "shrink wrap" software) and where it is possible and likely that designers will be able to fully predict problem areas of the system and produce a correct design before implementation is started. The waterfall model also requires that implementers follow the well made, complete design accurately, ensuring that the integration of the system proceeds smoothly. Sunil sees the two big advantages of the pure waterfall model as producing a "highly reliable system" and one with a "large growth envelope", but rates it as poor on all other fronts. On the other hand, he views any of several modified waterfall models as preserving these advantages while also rating as "fair to excellent" on "working with poorly understood requirements" or "poorly understood architecture" and "providing management with progress visibility", and rating as "fair" on "managing risks", being able to "be constrained to a predefined schedule", allowing for midcourse corrections", and providing customer with progress visibility". The only criterion on which he rates a modified waterfall as poor is that it requires sophistication from management and developers.

### 7.1.4 Modified Waterfall Models

In response to the perceived problems with the "pure" waterfall model, many modified waterfall models have been introduced. These models may address some or all of the criticisms of the "pure" waterfall model.

While all software development models will bear at least some similarity to the waterfall model, as all software development models will incorporate at least some phases similar to those used within the waterfall model, this section will deal with those closest to the waterfall model.

*Caution* For models which apply further differences to the waterfall model, or for radically different models seek general information on the software development process.

### Self Assessment

Fill in the blanks:

1. .............................. method focused on what the system or application does rather than on how it is done.
2. Structured Systems Analysis and Design Method (SSADM) is a .............................. approach to the analysis and design of information systems.
3. The .............................. model is a sequential software development model in which development is seen as flowing steadily downwards through the phases of requirements analysis, design, implementation, testing, integration, and maintenance.
4. In response to the perceived problems with the "pure" waterfall model, many .............................. waterfall models have been introduced.

## 7.2 Features of Structured Analysis Development Strategy

Structured Analysis Development Strategy comprises three significant features:

- ***Structures:*** It Define the frameworks of steps and levels and their inputs and outputs.
- ***Techniques:*** It define how the steps and tasks are executed.
- ***Documentation:*** It define how the products of the steps are displayed.

Notes

### 7.2.1 Structures of SSADM

SSADM application development projects are alienated into five modules that are further divided into a hierarchy of levels, steps and tasks:

1. ***Feasibility Study:*** The business area is examined to identify whether a system can cost effectively assist the business requirements.
2. ***Requirements Analysis:*** The requirements of the system to be generated are recognized and the current business environment is modeled regarding the processes executed and the data structures involved.
3. ***Requirements Specification:*** Detailed functional and non-functional requirements are identified and new methods are introduced to define the needed processing and data structures.
4. ***Logical System Specification:*** Technical systems choices are produced and the logical design of update and enquiry processing and system dialogues.
5. ***Physical Design:*** A physical database design and a set of program stipulations are generated by means of the logical system specification and technical system specification.

*Task* Make distinction between Requirements Specification and Logical System Specification.

### 7.2.2 Techniques of Structured Analysis

The three most important techniques that are used in SSADM are:

- ***Logical Data Modelling:*** This is the process of identifying, modelling and documenting the data requirements of the system being designed. The data are separated into entities (things about which a business needs to record information) and relationships (the associations between the entities).
- ***Data Flow Modelling:*** This is the process of identifying, modelling and documenting how data moves around an information system. Data Flow Modeling examines processes (activities that transform data from one form to another), data stores (the holding areas for data), external entities (what sends data into a system or receives data from a system), and data flows (routes by which data can flow).
- ***Entity Behaviour Modelling:*** This is the process of identifying, modelling and documenting the events that affect each entity and the sequence in which these events occur.

### 7.2.3 Documentation

Several different kinds of document will be produced during a project such as diagrams, forms, matrices, and narrative reports. Working documents are produced as a means of developing some of the diagrams. Other documents are part of the formal documentation standards of SSADM and are carried forward into later steps of the project.

In addition to the diagrams there is a huge amount of highly structured information that needs to be developed. To help record this structured information a number of standard forms are suggested.

In addition to forms, matrices are used to help start some of the diagrams. An entity matrix is used to identify relationships between entities as an initial step in the logical data structuring technique and an entity/event matrix is used as a basis for Entity Life histories.

*Did u know?* Matrices are also sometimes useful in cross-checking one technique with another.

*Notes* In many cases formal reports such as feasibility study reports or full study reports will be required. Although SSADM does not provide detailed recommendations on the content or format of such reports as they are often set by the organization, much of the SSADM documentation including diagrams and forms are needed to be integrated into these formal reports.

## Self Assessment

Fill in the blanks:

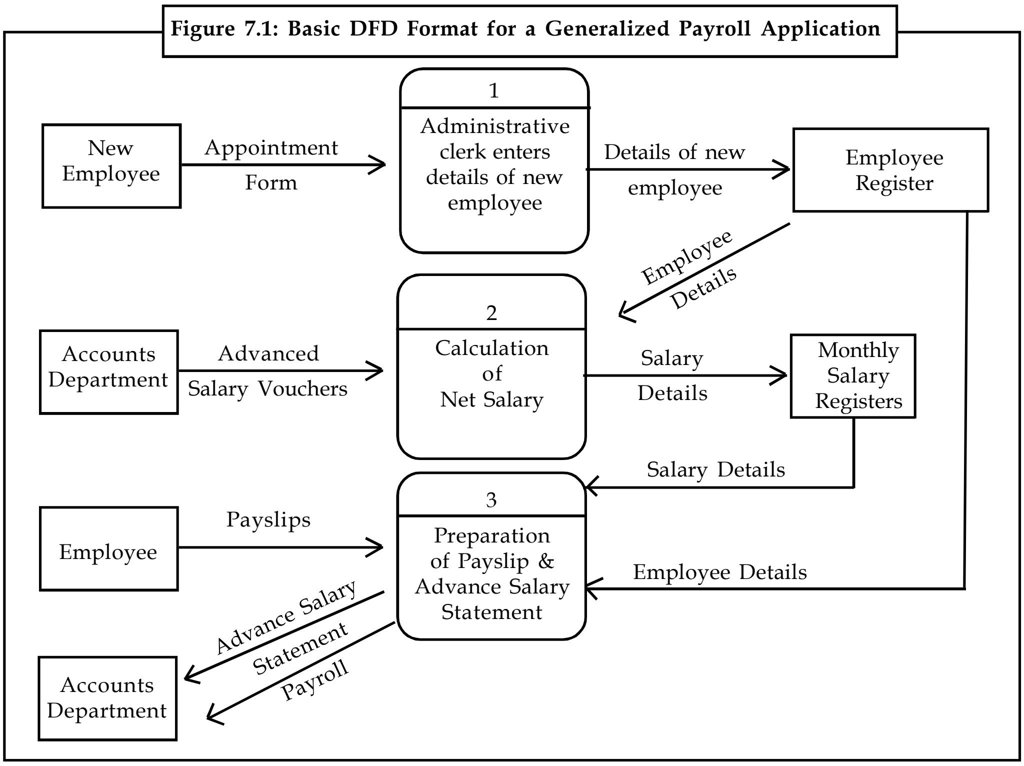
5. .............................. define the frameworks of steps and levels and their inputs and outputs.
6. In .............................. the business area is examined to identify whether a system can cost effectively assist the business requirements.
7. .............................. define how the steps and tasks are executed.
8. .............................. define how the products of the steps are displayed.
9. In .............................. Specification, technical systems choices are produced and the logical design of update and enquiry processing and system dialogues.
10. .............................. Modelling is the process of identifying, modelling and documenting the data requirements of the system being designed.
11. .............................. Modelling is the process of identifying, modelling and documenting the events that affect each entity and the sequence in which these events occur.
12. In.............................., the requirements of the system to be generated are recognized and the current business environment is modeled regarding the processes executed and the data structures involved.

## 7.3 Data Flow Tools

1. ***Data Flow Diagrams (DFDs):*** Data flow diagrams are widely used graphic tools for describing the movement of data within or outside the system. As a DFD consists of a series of bubbles joined by lines, it is also known as a 'bubble chart'. The basic DFD format is shown in Figure 7.1.
2. ***Data Dictionary:*** Data dictionary is an organized list of terms and their definitions for all the data elements and data structures that are pertinent to the system. It stores the names along with their descriptions of all data used in a system.
3. ***Process Descriptions:*** Process descriptions are another major tool of structured analysis, that describe the sequence of different processes in the system. Structured English (pseudo code), decision tree and decision table are commonly used process descriptions.

Notes

Figure 7.1: Basic DFD Format for a Generalized Payroll Application



## Self Assessment

Fill in the blanks:

13. As a DFD consists of a series of bubbles joined by lines, it is also known as a '...............................'.

14. ............................... is an organized list of terms and their definitions for all the data elements and data structures that are pertinent to the system.

15. ............................... are the major tool of structured analysis, that describe the sequence of different processes in the system.

## 7.4 Summary

- Structured Systems Analysis and Design Method (SSADM) is a discipline within the software development industry which seek to provide a framework for activity and the capture, storage, transformation and dissemination of information so as to enable the economic development of computer systems that are fit for purpose.
- SSADM is a waterfall strategy by which an Information System design can be arrived at.
- The waterfall model is a sequential software development model in which development is seen as flowing steadily downwards through the phases of requirements analysis, design, implementation, testing, integration, and maintenance.
- The waterfall model maintains that one should move to a phase only when its preceding phase is completed and perfected.
- Structures define the frameworks of steps and levels and their inputs and outputs.

- Techniques define how the steps and tasks are executed.
- Documentation define how the products of the steps are displayed.
- Data flow diagrams are widely used graphic tools for describing the movement of data within or outside the system.
- Data dictionary is an organized list of terms and their definitions for all the data elements and data structures that are pertinent to the system.
- Process descriptions are another major tool of structured analysis, that describe the sequence of different processes in the system.

## 7.5 Keywords

***Data Flow Modelling:*** The process of identifying, modelling and documenting how data moves around an information system.

***Entity Behaviour Modelling:*** The process of identifying, modelling and documenting the events that affect each entity and the sequence in which these events occur.

***Logical Data Modelling:*** The process of identifying, modelling and documenting the data requirements of the system being designed.

***Requirements Analysis:*** The requirements of the system to be generated are recognized and the current business environment is modeled regarding the processes executed and the data structures involved.

***Structured System Analysis:*** A systems approach to the analysis and design of information systems. It is a discipline within the software development industry which seek to provide a framework for activity and the capture, storage, transformation and dissemination of information so as to enable the economic development of computer systems that are fit for purpose.

## 7.6 Review Questions

1. What are the various methodologies of system analysis and design? Illustrate.
2. Discuss the concept of Structured System Analysis.
3. Elucidate the strategy by which an Information System design can be arrived.
4. Illustrate various features of Structured Analysis Development Strategy.
5. To follow the waterfall model, one proceeds from one phase to the next in a purely sequential manner. Comment.
6. Describe various techniques used for structured system analysis.
7. Discuss the concept of documentation in Structured Analysis Development.
8. What are the different data flow tools? Illustrate.
9. Make distinction between Requirements Analysis and Requirements Specification.
10. Make distinction between Logical Data Modelling and Data Flow Modelling.

### Answers: Self Assessment

1. Structured analysis
2. systems
3. waterfall
4. waterfall

Notes

5. Structures
6. Feasibility Study
7. Techniques
8. Documentation
9. Logical System
10. Logical Data
11. Entity Behaviour
12. Requirements Analysis
13. bubble chart
14. Data dictionary
15. Process descriptions

## 7.7 Further Readings

*Books*

Charles S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*, 2006, Wiley-Interscience.

Dorothy J. Tudor; Ian J. Tudor, *System Analysis and Design: A Comparison of Structured Methods*, 1995, NCC Blackwell.

Elias M. Awad, *Business Data Processing*, 1980, Prentice Hall (Higher Education Division, Pearson Education).

Elias M. Awad, *Systems Analysis and Design*, 1985, R.D. Irwin.

Gary B. Shelly; Thomas J. Cashman, Harry J. Rosenblatt, *System Analysis and Design*, 2001, Course Technology.

Jeffrey L. Whitten; Lonnie D. Bentley; Kevin C. Dittman, *Systems Analysis and Design Methods*, 2003, McGraw-Hill.

John W. Satzinger, Jackson, Burd, *System Analysis & Design,* 2006, Thomson Learning.

Joseph S. Valacich; Joey F. George; Jeffrey A. Hoffer, *Essentials Of System Analysis And Design*, 2005, Prentice Hall.

Leonard Fertuck, *System Analysis and Design,* 1994, McGraw-Hill Higher Education.

Penny A. Kendall, *Introduction to System Analysis and Design: A Structured Approach*, 1996, Irwin.

Preeti Gupta, *System Analysis and Design*, 2005, Firewall Media.

Ronald G Noice, *System Analysis & Design*, 1984, Oxford University Press Canadian Branch.

*Online link*

http://searchsoftwarequality.techtarget.com/definition/SSADM

# Unit 8: Tools for Structured Design

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the various tools for structured design
- Explain the data flow diagrams
- Discuss the concept of data dictionaries

## Introduction

Structured design is the most popular methodology for developing system designs. In this methodology, the software is considered as a transformation function that converts the given inputs into desired outputs. Structured design divides the system into different modules that are arranged in a hierarchy and organized in top-down manner. The main objective of structured design is to reduce the coupling and enhance the cohesion between different modules.

In the structured design methodology, the system is designed through the following steps:

(i) Identification of inputs and outputs from SRS;

(ii) Description of functional aspects of the system;

(iii) Representation of system in DFD;

Notes

(iv) Formation of Data dictionary;

(v) Documentation of structured design by structure charts.

Although, structured design is the most popular methodology, it has a major drawback. The main disadvantage of structured design is that it does not model the real life system.

## 8.1 Tools for Structured Design

Tools for structured design include:

- ***Data Flow Diagram:*** Graphical description of a systems data and how the processes transform the data is known as Data Flow Diagram (DFD).
- ***Data Dictionary:*** Data dictionary is a catalogue of all data elements, data structures and processes described in logical DFDs.

These tools are discussed in detail in the sections below.

### Self Assessment

Fill in the blanks:

1. Graphical description of a systems data and how the processes transform the data is known as ............................... .
2. .............................. is a catalogue of all data elements, data structures and processes described in logical DFDs.

## 8.2 Data Flow Diagrams

During analysis phase of SDLC, the systems analyst or other members of the project team draw many diagrams to show how data move within an organization. These diagrams, popularly called as DFD (Data Flow Diagram), quickly convey to both the software developers and users how the current system is working and how the proposed system will work. The main advantage of DFD is that they are easily understood by the users, and hence users can suggest modifications in the proposed system. We will discuss now the different types of DFD and see how these DFD can be drawn.

### 8.2.1 Types of DFD

There are two types of DFD - Physical and Logical DFD.

- ***Physical DFD:*** The data flow diagrams which represent the model of the current system (manual or computerised), are known as physical DFD. These diagrams are drawn, when the analyst studies the current working system in detail.
- ***Logical DFD:*** The data flow diagrams which represent the model of the proposed system, are known as logical DFD. These diagrams are drawn from the physical DFD.

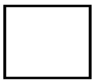
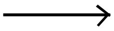
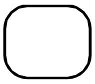
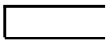
Each of these DFD can be further categorized into different levels like zero level (context diagram), first level, second level and third level.

*Did u know?* Each higher level DFD is drawn by adding more details to each process of lower level, by a technique called 'Exploding DFD'.

### 8.2.2 DFD Modeling Notation

The four notations that are widely used in DFD, are shown in Table 8.1. The description of these notations is explained below:

Table 8.1: DFD Modeling

| Symbol | Descriptions |
|---|---|
|  | External Entity |
|  | Data Flow |
|  | Process |
|  | Data Store |

(a) ***External Entity:*** External Entity represents any entity that supplies data or receives information from the system.

*Example:* Customer, sales department, employee, etc., are external entities.

(b) ***Data Flow:*** The data flow indicates the movement of data either from input to process or from process to output. Data flow is labeled to show what data is flowing.

*Example:* Customer details, sale reports, etc., are data flows.

(c) ***Process:*** Processes are the actions performed on input data to produce the output data. They are given some meaningful names.

*Example:* Prepare Bill, Calculate Sales, Compute Pay, etc., are the processes.

(d) ***Data Store:*** Data store indicates the data file or register where data is accumulated.

*Example:* Customer Master File, Employee Register, Sales Transaction File, etc., are data stores.

### 8.2.3 Steps to draw DFD

The different level physical and logical DFD are generally drawn in the following steps:

1. Identify external entities and data flows of the current system and draw physical context diagram.
2. Identify data stores and processes of the system and draw first level physical DFD.
3. Explore the processes of first level and draw second level DFD.
4. Explore the processes of second level and draw third level DFD.
5. Derive the logical view of each physical DFD by the following ways:
   (a) Remove documents and show actual data in data flow.
   (b) Remove registers and use files as data stores.
   (c) Remove unnecessary processes.
   (d) Remove data flow between external entities (if any) and show data flow through processes.

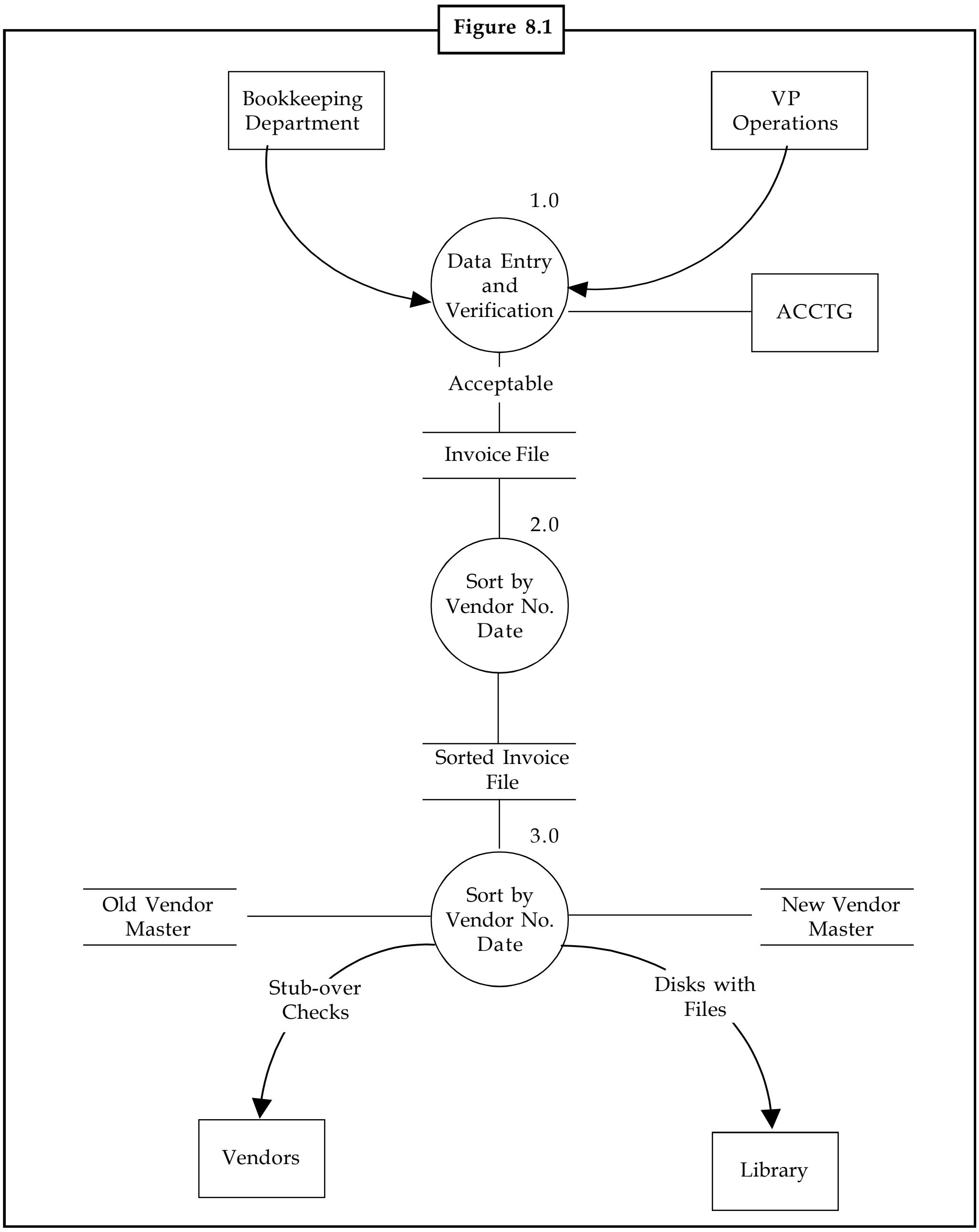
Notes

### 8.2.4 Rules to draw DFD

The following seven rules govern construction of data flow diagrams (DFD):

1. Arrows should not cross each other.
2. Squares, circles, and files must bear names.
3. Decomposed data flows must be balanced (all data flows on the decomposed diagram must reflect flows in the original diagram).
4. No two data flows, squares, or circles can have the same name.
5. Draw all data flows around the outside of the diagram.

**Figure 8.1**

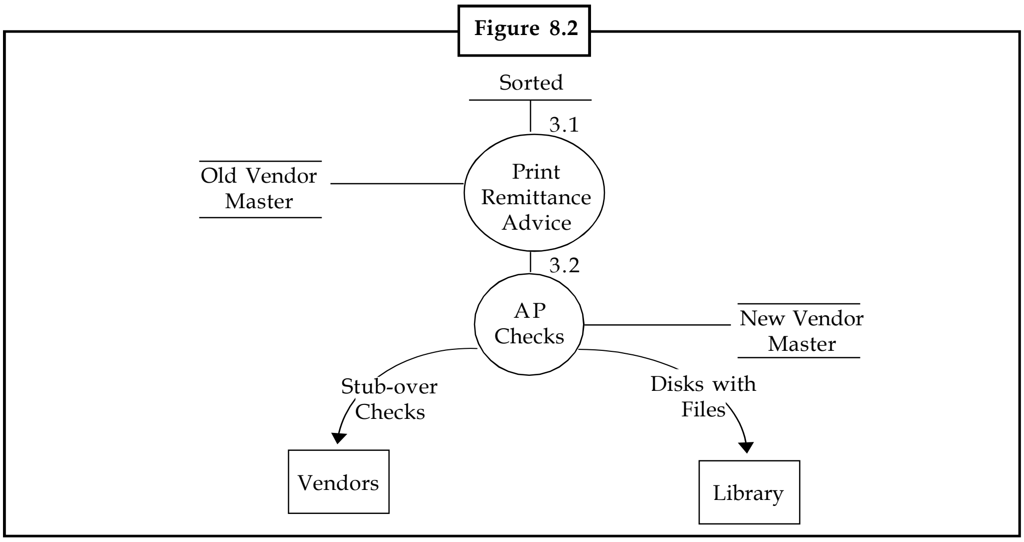


6. Choose meaningful names for data flows, processes, and data stores. Use strong verbs followed by nouns.

7. Control information such as record counts, passwords, and validations requirements are not pertinent to a data-flow diagram.

If too many events seem to be occurring at a given point, an analyst can decompose a data conversion (circle). The new data conversions form a parent-child relationship with the original data conversion the child circle in Figure 8.1 belongs to the parent in Figure 8.2.

Figure 8.2



Devotees of data-flow diagrams insist that no other analyst's tool expresses so fully the flow of data. After all, don't computer people begin with data flow rather than the processing of the data? Another strong advantage is the balancing feature that builds in an error-detection system other tools lack.

*Example:* If a parent data-flow diagram shows three inputs and two outputs, the leveled child diagrams taken together must have three inputs and two outputs. If there is an imbalance between parent and child data-flow diagrams, an error exists in either the parent or child diagram.

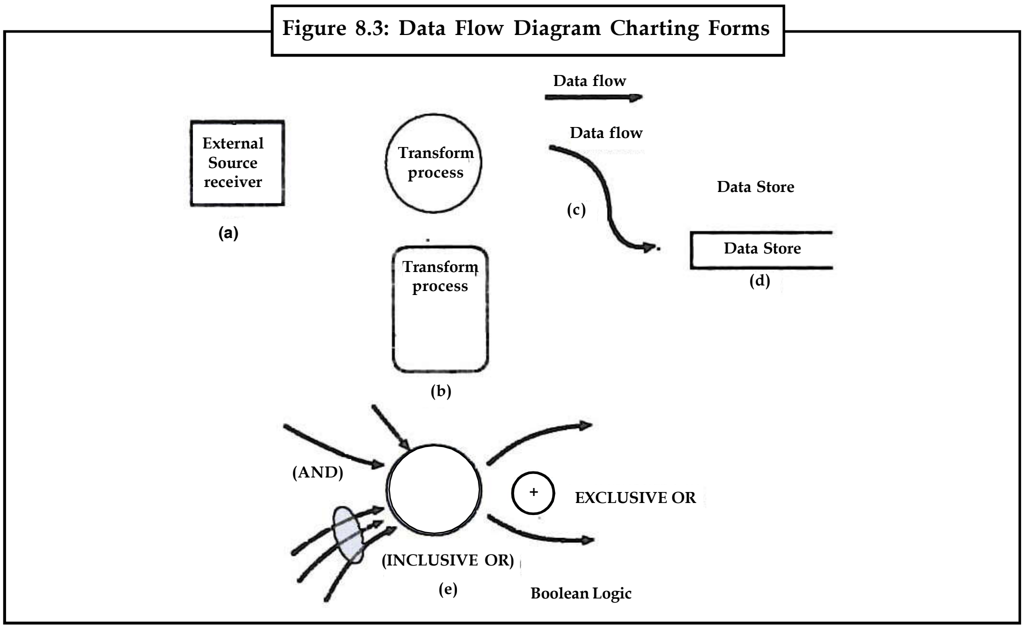
## 8.2.5 Charting Tools used for DFDs

The Data Flow Diagram (DFD) is the core specification in this method. Figure 8.3 shows the very few charting forms that are necessary. It is shown in Figure 8.3(a) as a square box is the external source or receiver of data. Shown in Figure 8.3(b) is the transform bubble. Two variations of this have been put forth. The circle of real bubble is the better-known and is used by both Victor Weinberg and Tom DeMarco. The rectangular bubble shown beneath the circle is the form used by Chris Gane and Trish Sarson. The reason for the rectangular bubble is the perceived need to enter more information than can be contained in the bubble. Tom DeMarco, who is the purist in this group of four writers and educators in the structured method, holds that the data content of the bubble must be just the bare bones of one verb and a noun, since our objective is not to explain the process but to partition into leveled transforms.

*Caution* In fact the rectangular bubble is hard to draw freehand and the template containing this special form is not one you are likely to have around the shop, it must be specially ordered.

Notes

Figure 8.3: Data Flow Diagram Charting Forms



The line arrow is much more important in this method because it carries the data flow: the data into the transform and the data out of the transform. Figure 8.3(c) shows the line arrows. The variations on the use of the line arrow are significant, because again the answer needs that different proponents of this method have perceived. There different points of view are advanced by the practitioners mentioned above regarding line arrows.

1. When multiple lines go into or leave a transform Weinberg offers the ability to use Boolean logic describing symbols to represent AND, INCLUSIVE OR, and EXCLUSIVE OR. This clearly indicates a felt need for decision logic above the base level. Demarco advises against using Boolean decision logic. Our examples will not use Boolean logic beyond showing examples in Figure 8.3(e), since this seems to the author to present a consensus view to those who use the DFD approach.

2. DeMarco Shows the line arrow as a curved line giving a different "feeling" than the straight lines and right angles of Weinberg and Gane-Sarson. In the DeMarco approach there is more of a sense a flow -a sense of data in motion. Our examples will use the curved line.

3. Regarding the data flow along the line arrow, this would be a serious problem, like the problem of expressing the process within the confines of the bubble. The answer here is to take advantage of the fact that the method allows multiple lines in and out of a bubble and to break up the wordy data flow into several briefly named data flows. It is either that or lengthen the line. This concentration on detail of form, worrying about whether to use a circle or a square or a curved or a straight line, may read as petty to the non-chartist. To those who mean to use this method to specify systems it is just as serious mater as the concern of the professional tennis player with the type of racket to be used in a tournament. What we are trying to do with these forms is to invent solutions to problems as we move from the fluid to the concrete and from the tentative to the certain. We need a method for all seasons, but especially to communicate the fluid and the tentative new idea.

   The final diagramming element shown in Figure 8.3(d) is the open rectangle or two parallel lines, which indicates the data store (such as a database, file, Kardex or phone book). Gane and Sarson, unlike the others, show the data store as the two parallel lines joined at one side to make an open rectangle.

*Notes* These are all the charting forms we need to use this methodology. Again, as in the flowcharting forms much can be developed out of a few tools. Essentially a system of any complexity whatever is shown with the bubble, line, data store rectangle, and external box. It can be seen that although some practitioners prefer to use variation in their notation, the broad style is similar.

*Caution* All lines must be identified by their data.

*Task* Illustrate various charting tools used for DFD's.

### Self Assessment

Fill in the blanks:

3. The data flow diagrams which represent the model of the current system (manual or computerized), are known as............................... .
4. The data flow diagrams which represent the model of the proposed system, are known as ............................... .
5. The ............................... indicates the movement of data either from input to process or from process to output.
6. ............................... are the actions performed on input data to produce the output data.
7. ............................... Entity represents any entity that supplies data or receives information from the system.
8. Each higher level DFD is drawn by adding more details to each process of lower level, by a technique called '...............................' .

## 8.3 Data Dictionary

Before we discuss the importance and contents of a data dictionary, let us understand the meaning of the following terminology:

- ***Data Element:*** Data element is the smallest unit of data that has some meaning.

  *Example:* Part code, part name, date of transaction, etc., are data elements.

- ***Data Structure:*** Data structure is a group of data elements that describe a unit in the system.

  *Example:* Part Details is a data structure that contains part code, part name and date of transaction as data elements.

- ***Data Store:*** Data store is a data structure for collecting data input during processing.

  *Example:* Part Register is a data store.

- ***Data Flow:*** Data flow is a data structure, that shows a unit of data in motion.

  *Example:* New Part Details is a data flow that moves from an external entity to a process.

A data dictionary defines each term (called a data element) encountered during the analysis and design of a new system. Data elements can describe files, data flows, or processes.

*Example:* Suppose you want to print the vendor's name and address at the bottom of a cheque. The data dictionary might define vendor's name and address as follows:

Vendor name and address = Vendor name+<br>
Street+<br>
City+<br>
State+<br>
Pin +<br>
Phone+<br>
Fax+<br>
e-mail

This definition becomes a part of the data dictionary that ultimately will list all key terms used to describe various data flows and files.

### 8.3.1 Major Symbols

A data dictionary uses the following major symbols:

| | | |
|---|---|---|
| (i) | = | Equivalent to |
| (ii) | + | And |
| (iii) | [ ] | Either/or |
| (iv) | ( ) | Optional entry |

### 8.3.2 Rules to draw Data Dictionaries

Four rules govern the construction of data dictionary entries:

1. Words should be defined to stand for what they mean and not the variable names by which they may be described in the program: use CLIENT_NAME not ABCPQ or CODE06. Capitalization of words helps them to stand out and may be of assistance.
2. Each word must be unique; we cannot have two definitions of the same client name.
3. Aliases, or synonyms, are allowed when two or more entries show the same meaning; a vendor number may also be called a customer number. However, aliases should be used only when absolutely necessary.
4. Self-defining words should not be decomposed.

*Example:* Examples of Data dictionary

The data elements, data stores, data flows and processes are described in a data dictionary as illustrated in Exhibit 8.1, 8.2, 8.3, and 8.4.

Notes

**Exhibit 8.1: An Example of Data Element in a Data Dictionary**

| | | |
|---|---|---|
| PART CODE<br>Short Description: This element describes the code of a part at or subassembly. | | Data Element |
| Organisation: BR. Auto Limited | | |
| Date: | | |
| Aliases (contents): Part Number | | |
| If Discrete | | If Continuous |
| Value | Meaning | Range of |
| | | Value: A001 to Z999 |
| | | Typical Value: A001 |
| | | Length: 4 |
| | | Internal Representation: Character |

**Exhibit 8.2: An Example of Data Store in a Data Dictionary**

| | |
|---|---|
| TRASACTIONS FILE<br>Description: Day's Transaction | Data Store |
| Data Flow in: | Data flow out: |
| Issue, Return, Receipt | Consolidated transaction of the day |
| | |
| Contents: Part No., Part Name, Date of Transaction, Issue, Receipt, Name of Concerned Person and posting Status | |
| | Physical Organisation: Store |

**Exhibit 8.3: An Example of Data Flow in a Data Dictionary**

| | |
|---|---|
| New Part Details | Data Flow |
| Source Ref. Description: | Part |
| Destn. Ref. Description: | Part Register |
| Expanded Description: | Part Details like: Part code, Part name, Date of Transaction etc. are entered into the Part register. |
| Included data structure: | Volume Information |
| Part register | Volume increases when new part is entered but do not decrease when old part is discarded. |

**Exhibit 8.4: An Example of Process in a Data Dictionary**

| Description: Maintain part Register | | Process |
|---|---|---|
| Input | Logic Summary | Output |
| New part details | All part details with a unique part code is accepted.<br>Duplicate part code is not accepted | Updated part |
| Daily Transaction Details | Daily Transactions<br>Details of those part codes are accepted that exist in part-Registers. | |

### 8.3.3 Importance of Data Dictionary

Data dictionary is an important tool for structured analysis as it offers following advantages:

- It is a valuable reference for designing the system. It is used to build the database and write programs during design phase.
- It assists in communicating meanings of different elements, terms and procedures.
- It facilitates analysis in determining additions and changes in the system.
- It helps the analyst to record the details of each element and data structure.
- It is used to locate errors in the system descriptions.
- It is also a useful reference document during implementation of the system.

### 8.3.4 Types of Data Dictionary

Table 8.2 illustrates the different types of data dictionaries and the functions of each addresses.

**Table 8.2: Data-Dictionary Types and Functions**

| Type<br>Function | Stand-alone<br>Global:<br>manual or automated | Integrated with one database management system (DBMS) |
|---|---|---|
| PASSIVE<br>Documenting function only | Full organisation documentation possible | |
| ACTIVE<br>Active in program preparation but not during execution | Full organisation documentation possible plus:<br>Supports program and operations development with data structure (like program data definitions at even editing and validation code) | Full documentation possible<br>• Supports program and operations development with data structure<br>• Supports database definitions language, database definition, and program specification blocks. |
| IN-LINE<br>Also active during program execution<br>May have only limited documentation function | | Full documentation not possible in most cases<br>• Checks transaction and report syntax during job execution.<br>• Can edit and validate input transactions at the dictionary level in-line rather than per application. |

There are two kinds of data dictionaries:

i. Integrated and

ii. Stand-alone.

The integrated dictionary is related to one database management system. To the extent the organization data is under this DBMS it is global or organisation wide. However, very few enterprises have all their data eggs in one basket, so the dictionary documentation (metadata) can be considered as local and fragmented.

The stand-alone dictionary is not tied to any one DBMS, although it may have special advantages for one DBMS, such as the IBM DB-DC Data Dictionary, which has special features related to the IBM IMS DBMS but is still a stand-alone variety of dictionary.

*Task* Make distinction between integrated dictionary and stand-alone dictionary.

### 8.3.5 Data Dictionary Functions

Both these types of dictionaries can be identified by functions as either passive, active or in-line. Viewed either way, by type or function, the differences are striking. Passive, active, and in-line dictionaries differ functionally as follows:

- ***Passive Data Dictionaries:*** The functionally passive dictionary performs documentation only. This variety of dictionary could be maintained as a manual rather than an automated database. For more than limited documentation use, the automated passive dictionary has clear advantages.

*Notes* From the organizational view the documentation function is the most important dictionary service with the most potential benefits, so the passive dictionary should not be thought of negatively. It has more limited functionality but may perform its critical function of global documentation best of all.

- ***Active Data Dictionaries:*** Besides supporting documentation to one degree or another, the active data dictionary supports program and operations development by exporting database definitions and program data storage definitions for languages such as COBOL and Job Control Language (JCL) for execution-time performance. The IBM DB/DC Data Dictionary already mentioned is such a stand-alone, active data dictionary. A dictionary such as this is not an in-line by a determined effort of major proportions.

- ***In-line Data Dictionaries:*** An in-line data dictionary is active during program execution, performing such feats as transaction validation and editing. Such a dictionary would always have some documentation value, but documentation across the organisation about the organisation functions and activities and all the organisation information data stores is not likely.

*Did u know?* In-line dictionaries are associated with DBMS products such as Cullinet Software Corporation's IDMS-R or Cincom System's TOTAL, to name just two.

## Self Assessment

Fill in the blanks:

9. .............................. is the smallest unit of data that has some meaning.

10. .............................. is a data structure for collecting data input during processing.

11. .............................. is a group of data elements that describe a unit in the system.

12. The .............................. dictionary is related to one database management system.

13. The .............................. dictionary is not tied to any one DBMS, although it may have special advantages for one DBMS.

14. The functionally passive dictionary performs .............................. only.

15. An .............................. data dictionary is active during program execution, performing such feats as transaction validation and editing.

## 8.4 Summary

- The main objective of structured design is to reduce the coupling and enhance the cohesion between different modules.
- Graphical description of a systems data and how the processes transform the data is known as Data Flow Diagram (DFD).
- The data flow diagrams which represent the model of the current system (manual or computerised), are known as physical DFD.
- The data flow diagrams which represent the model of the proposed system, are known as logical DFD.
- Data dictionary is a catalogue of all data elements, data structures and processes described in logical DFDs.
- A data dictionary defines each term (called a data element) encountered during the analysis and design of a new system.
- The functionally passive dictionary performs documentation only and this variety of dictionary could be maintained as a manual rather than an automated database.
- The active data dictionary supports program and operations development by exporting database definitions and program data storage definitions for languages such as COBOL and Job Control Language (JCL) for execution-time performance.
- An in-line data dictionary is active during program execution, performing such feats as transaction validation and editing.

## 8.5 Keywords

***Data Dictionary:*** A catalogue of all data elements, data structures and processes described in logical DFDs.

***Data Element:*** The smallest unit of data that has some meaning.

***Data Flow:*** A data structure that shows a unit of data in motion.

***Data Flow Diagram:*** Graphical description of a System's data and how the processes transform the data.

***Data Store:*** A data structure for collecting data input during processing.

***Data Structure:*** A group of data elements that describe a unit in the system.

***External Entity:*** An entity that supplies data or receives information from the system.

***Logical DFD:*** The data flow diagrams which represent the model of the proposed system.

***Physical DFD:*** The data flow diagrams which represent the model (manual or computerized) of the current system.

***Process:*** Actions performed on input data to produce the output data.

## 8.6 Review Questions

1. What are data flow diagrams? Illustrate various symbols used in DFDs.
2. Describe the concept of data dictionary with example.
3. Enlighten the various types of data flow diagrams.
4. Illustrate stepwise procedure to draw DFD.
5. Enlighten the rules to draw DFD.
6. What are the various charting tools of DFD? Illustrate.
7. Write the rules to draw data dictionaries.
8. Elucidate the advantages of data dictionary.
9. Depict various types of data dictionary.
10. Make distinction between active and passive data dictionary.

### Answers: Self Assessment

1. Data Flow Diagram (DFD).
2. Data dictionary
3. physical DFD
4. logical DFD
5. data flow
6. Processes
7. External
8. Exploding DFD
9. Data element
10. Data store
11. Data structure
12. integrated
13. stand-alone
14. documentation
15. in-line

## 8.7 Further Readings

*Books*

D R Jeffery; M J Lawrence, [Sydney, N.S.W. ; Englewood Cliffs, N.J.]: *Systems Analysis and Design*, Prentice-Hall of Australia, 1985, ©1984.

James C Wetherbe, St. Paul, *Systems Analysis and Design*, West Pub. Co., ©1988

S.A. Kelkar, *Structured System Analysis and Design*, Prentice Hall of India.

*Online link*

http://www.csus.edu/indiv/m/martinm/116-DeveTools/index.htm

# Unit 9: Application Prototypes

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the concept of application prototypes
- Identify the purpose of prototypes
- Discuss the steps used for prototypes
- Illustrate the prototyping tools
- Discuss the prototype example

## Introduction

Prototyping is the process of creating, developing and refining a working model of the final operational system. The prototype is a live working system and not just a paper-based design. Users can test its operations and explore its facilities and so, do not have to rely upon written descriptions. Prototyping is an engineering discipline that has found its way into computer systems development. The idea, at least in the engineering parlance is to build small-scale working models of a product or its components. When it applies to computer systems development, it means that the analyst builds a small scale working model of the system or a sub-system. In this unit you will understand some concepts of application prototypes such as purpose, steps tool, etc.

## 9.1 Prototyping

Prototyping stresses the early delivery of a working system that could be used to refine user's requirements and system characteristics. Prototyping 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. The user's experience with this experimental system leads to suggestions and tips for modifications to be incorporated into the prototype, thus leading to a new series of user experimentation.

Hence, the prototype is successively refined in a series of iterations until it eventually becomes an acceptable reflection of the user's requirements.

A prototyping strategy signifies the construction model of a system. Designing and constructing a scaled-down but functional edition of a preferred system is the process called Prototyping. A prototype is a functioning system that is produced to check ideas and assumptions concerning the new system. It includes working software that accepts input, performs computations, generates printed or exhibited information or carry out other significant activities. It is the first edition or iteration of an information system i.e. an original model. Customer assesses this model. This can be successfully performed only if the data are real and the conditions are live. Changes are predictable as the system is accessed.

A prototype is a working system – not just an idea on paper that is developed to test ideas and assumptions about the new system. Like any computer based system, it consists of working software that accepts inputs, performs calculations, produces printed or displayed information or performs other meaningful activities. It is the first version or iteration of an information system in original model.

The design and the information produced by the system are evaluated by users. This can be effectively done only if the data are real and the situations alive. Changes are expected as the system is used.

Although it is not a structured technique yet prototyping has become a very popular complementary methodology. Based on an engineering approach, prototyping builds small scale working models of a system. These models, called prototypes, help end-users more quickly define requirements, feasibility and designs. Prototyping would detrimental when used to take shortcuts around systems analysis.

Methodologies are gaining powerful new support from CASE, Computer Assisted software Engineering. CASE provides automated tools to improve both productivity and quality when the structured techniques and prototyping are used.

Whole application systems are incorporated with the prototype in order that their functionality can be shared.

*Example:* If text preparation is needed, a standard word processor can be utilized.

The prototype is actually a pilot or test model; the design evolves through usage. If use of the sales prototype system reveals that entering customers' names and addresses through a portable terminal creates too many errors, designers may modify the system so that only customer's names are fed; the addresses can be automatically retrieved from files stored in the system.

*Notes* System prototyping is an interactive and iterative or evolutionary development in which the user is directly involved in the process. It may begin with only a few functions and be expanded to include others that would be identified later. It may also start with what both analyst and users believe is a complete set of functions that may expand or contract through usage and experience.

*Caution* The development of a prototype application proceeds in an orderly fashion, regardless of the particular tools used.

Prototyping is considered as a methodology. As a methodology is a compilation of methods, you may systematize the methods into steps. You may write them down in the sequence in which they should be performed. Prototyping is performed in an organized manner, and that manner can be illustrated, i.e., can be skilled, planned, calculated, compared, and customized - which are the components of any methodology.

Prototyping as a methodology is significant in constructing fast, improved, more dependable, improved quality systems. Now we will discuss why and how that can be attained.

Prototyping depends on constructing a model of a system to be developed. Furthermore, the initial model should comprise the chief program modules, the database, screens, reports, and the inputs and outputs that the system will utilize for conversing with other systems.

Prototyping utilizes the model for designing the system. The preliminary version of the prototype is not full system, but it does enclose its designer's perceptive of the database, screens, and reports. As the user and Information Systems people start to function with the prototype, it will vary. The preliminary version of the prototype is a skeletal version of the system; it does not enclose all the processing and validation policies that the system will eventually have. As those functioning with the prototype change it and add to it, they will be finishing the design of the system. Thus, the prototype will be a vehicle for designing the concluding version of the system.

Prototyping utilizes the model to execute the system. In current average organization, you write programs in some problem solution language, like COBOL, to execute a system. The preliminary version of the prototype will include programs specified in some language (probably a fourth generation language) to shift data back and forth among the screens, the database, reports, and the inputs and outputs used to converse with interfacing systems. In the beginning, these programs may do slight processing; they may in fact dummy it. As prototyping process persists, newer versions of programs that execute more closely to those of the eventual system, will substitute the unique versions.

*Example:* A program that in fact takes out data for a report from a database may restore one that dummied out data.

Prototyping make use of the model to execute both the system and the acceptance testing of the system. The preliminary version of the prototype, in addition to all succeeding versions, will converse with system test versions of feeding systems and systems to be fed. Thus, the prototyping will at all times run in "system test" mode. And, as the user will be functioning with the prototype from the commencement, the user will be performing "acceptance testing" of the prototype from the beginning.

Prototyping does not go away from a mock-up if, after building and researching with the preliminary model of the system, and perhaps making some modifications to it, you discard it when you have attained a good understanding of the needs. A prototype as a mock-up is precious. With the Prototyping method, although a prototype may be slight more than a mock-up when it is first constructed, it turned out to be the first one of its sort by the time it is completed. So, when the prototyping process finishes, the prototype has develop into the system.

At present Prototyping Methodology is having a larger and larger utilization in more and more organization. Part of the cause is the formation of software tools that permit the implementation of such methodology or strategy to build up software.

In regard of an information system, prototypes are engaged to assist system designers construct an information system that make it perceptive and simple to operate for end users. Prototyping is an iterative procedure that is part of the analysis stage of the systems development life cycle.

During the requirements determination segment of the systems analysis phase, system analysts collect information regarding the organization’s present procedures and business processes connected the projected information system. Additionally, they study the existing information system, if there is one, and carry out user interviews and amass documentation. This assists the analysts produce a preliminary set of system requirements.

Prototyping can expand this process since it converts these basic, yet at times intangible, conditions into a tangible but restricted functioning model of the preferred information system. The user reaction attained from producing a physical system that the users can touch and see assist an evaluative reaction that the analyst can utilize to vary the current requirements in addition to developing new ones.

Prototyping occurs in many forms - from low tech sketches or paper screens from which clients and developers can paste controls and objects, to high tech operational systems by means of CASE (computer-aided software engineering) or fourth generation languages and all over in between. Many organizations make use of numerous prototyping tools.

*Example:* Some will make use of paper in the preliminary analysis to aid tangible user feedback and then afterwards develop an operational prototype by means of fourth generation languages, like Visual Basic, throughout the design stage.

The above illustrated Prototyping Methodology comprises the following major advantages:

- Satisfy users
- Decreases development cost
- Reduces communication problems
- Lowers operations expenditure
- Slashes calendar time needed
- Generates the right system the first time
- Cuts manpower required.

Prototyping Methodology also includes some disadvantages that are defined as below:

- Visible utilization of computer resources
- Object system may be less competent
- Needs cooperation among user and Information Systems
- Some observe prototyping as an art not a methodology.

Since prototypes intrinsically augment the eminence and amount of conversation among the developer/analyst and the end user, its' use has turn out to be extensive. In the early 1980's, organizations utilized prototyping about thirty percent (30%) of the time in development projects. By the early 1990's, its utilization had doubled to sixty percent (60%).

### *CASE Tools*

CASE tools and procedures are intended to mechanize application development. Using CASE tools business analysts and developers explain the software application and keep the depictions in a storage area. The CASE tools make use of the repository as a basis of information regarding the application to then produce the source code.

This technique of producing software accumulates information regarding the application and postpones the coding. It emphasizes RAD by eradicating the requirement for developers to write all of the source code. Rather, the CASE tools produce the source code. Programmers write only the code that cannot be articulated declaratively and/or the components that are away from the potentials of the CASE tools.

CASE tools are assorted and assist a broad choice of development errands. Some simple CASE tools have restricted functionality like model diagrams to produce database schemas. Other CASE tools will produce a whole application.

CASE contains intense and supple development methods. Business users function with developers to portray the application by focusing on particular components. The application needs go directly from the user into the CASE tool. This method generates completed components at ordinary short intervals which merge to provide the whole application.

CASE occurs into the lean development methodology. Users can choose the most significant parts of the application and these parts are the first components accomplished and executed for use.

CASE and joint development are perfect partners. CASE tools, being declarative in nature, facilitate users to be included in the development of the application right from the beginning They perform as advisors to business analysts and developers or they can contribute in stating the application. The learning curve for users is small as they do not require to know how to write applications. They only require to recognize how to utilize the CASE tools to depict the application.

CASE function with iterative and prototyping rapid application development methods. Using CASE tools to produce the source code for a component that has been illustrated is a rapid manner of prototyping the application. The prototype is reviewed, the descriptions are changed to make any needed changes and the source code is then restored. The application is constructed by repeated iterations.

*Did u know?* If prototyping is to be successful, then it is essential that the prototype is created quickly to give timely feedback to the development life cycle.

*Task* What does prototype strategy signify?

### Self Assessment

Fill in the blanks:

1. .................................... is the process of creating, developing and refining a working model of the final operational system.
2. Prototyping strategy signifies the .................................... model of a system.
3. System prototyping is an interactive and iterative or evolutionary development in which the.................................... is directly involved in the process.
4. The development of a prototype application proceeds in an orderly fashion, regardless of the particular .................................... used.

## 9.2 Purpose of Prototypes

A prototype is a functioning physical model of a system or a subsystem. In general, the analyst's goal is to collect information regarding the user's needs from the bottom up by permitting the user to cooperate with the prototype. Effectively, the prototype serves as a beginning edition of the system or constituent from which needs are extracted and on which succeeding editions are based.

Thus, the purpose of prototyping is to test out assumption made by analyst and users about required system features.

## Self Assessment

Fill in the blanks:

5. A prototype is a ................................... physical model of a system or a subsystem.

6. In general, the analyst's goal is to collect ................................... regarding the user's needs from the bottom up by permitting the user to cooperate with the prototype.

7. The purpose of prototyping is to ................................... out assumption made by analyst and users about required system features.

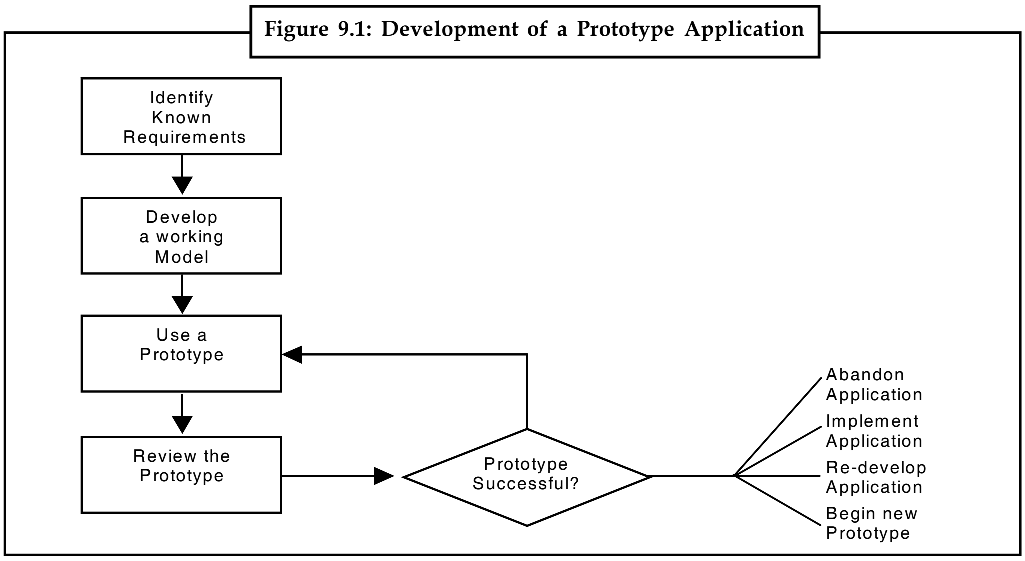
## 9.3 Steps used for Prototypes

Before starting the system design work, user and system analyst sit together and discuss to identify the requirements. These discussions form the basis of the construction of the prototype system analyst is fully responsible for the development of the working prototype.

The steps of Prototyping process are: They are:

- Recognize the user's well-known information requirements and features required in the system.
- Generate a working prototype.
- Revise the prototype depending on feedback obtained from customer.
- Replicate these steps as required to attain an acceptable system.

**Figure 9.1: Development of a Prototype Application**



*Caution* Actual growth of a functioning prototype is the liability of a systems analyst.

*Notes* The dissimilarity between a prototype model and an definite information system is that, a prototype will not comprise the error inspection, input data validation, security and processing totality of a completed application. It will not propose user assistance as in the final system.

Notes

### Self Assessment

Fill in the blanks:

8. Before starting the system design work, user and system analyst sit together and discuss to identify the .................................... .

9. Actual growth of a functioning prototype is the liability of a ................................... .

## 9.4 Use of Prototyping

A Prototype is the representation of software to be created. A Prototype assists in the institution of requirements more evidently. They may be created to reproduce the business functionality of the system, its extent of coverage, ease of use and appropriateness to the organization's manner of working, etc. Once the preliminary vagueness regarding the users' expectations and functionality of the systems has been cleared by prototyping, a more formal systems analysis and design can filter the prototype further. Therefore, by using prototyping as a balance to the use of a methodology, the benefits of both strategies can be preserved.

The prototype model consists of the following features:

(i) It assists in determining user requirements more deeply.

(ii) At the time of genuine product development, the customer feedback is obtainable.

(iii) It does consider any types of risks at the first level.

Prototyping is functional for identifying requirements for the software to be generated in the following circumstances:

1. Requirements are not clear.
2. For any intricate systems, prototypes are more functional.
3. In the cases where communication troubles occur between customer and analysts, this model is functional.
4. Tools and data are readily available for building the working system.

A prototype is typically a test model. It is an interactive process. It may start with only new functions and be stretched to comprise others that are recognized later.

*Did u know?* Prototyping is useful when the requirements are not well defined.

*Task* Illustrate the features of prototype model.

### Self Assessment

Fill in the blanks:

10. A Prototype is the representation of .................................... to be created.

11. Prototyping is useful when the requirements are not .................................... .

## 9.5 Prototype Tools and Prototype Example

After deciding on the prototyping strategy the prototyping tool must be chosen. The objective is to fit the tool to the needs of the system under development, the abilities of the developers, and the requirements of the users.

Here are a set of traits to be considered of when choosing a prototyping tool. They occurred from a set of requirements for user-interface prototyping, but can simply be extended to other portions of a prototyping project.

- ***Ease of use:*** Good prototyping tools should permit all members to contribute in development and improvement of the prototype. Steep learning curves are intolerable since many of the contributors do not have time to study the tools.
- ***Fast turnaround:*** In prototyping many small improvements must be made. Tools should permit developers to rapidly make variations and swiftly see the results.
- ***Extensive control over prototype features:*** Prototyping tools should be very supple so developers can attempt out new ideas.
- ***Data collection capabilities:*** For more multifaceted projects a perfect prototyping tool would be able to capture information regarding task specifications, system functionality, interface functionality, screen layouts and behavior, design rationale, user feedback, benchmarks, reusable code.
- ***Executability:*** The prototype should be as authentic to the real system as developers require to enlarge reliability of information collected.
- ***Lifecycle support:*** Prototyping tools can aid with all phases of development even away from systems analysis.
- ***Team design:*** Should sustain groups of people functioning together either concurrently or asynchronously, or distantly if these are necessary by the project.
- ***Version control:*** Developers may desire to discover and assess unusual designs. The prototyping tool should permit for edition control if it is required by the situation.

The prototype can be simply developed with tools of Fourth Generation Languages (4GL's) and with the aid of Computer Aided Software Engineering (CASE) tools. Prototyping strategy is a form of Rapid Application Development (RAD).

Many organizations utilize multiple prototyping tools.

*Example:* For example, some will make use of paper in the initial analysis to make possible concrete user feedback and then afterward build up an operational prototype by means of fourth generation languages, like Visual Basic, throughout the design stage.

Prototypes can be utilized to assist the thinking, scheduling, experimenting and knowledge processes at the same time as designing the product to the product development team. Questions and suspicions concerning some issues of the design can be addressed by constructing and learning the prototype.

*Example:* In scheming the suitable elbow-support of an organization chair, numerous physical prototypes of such elbow supports can be constructed to learn concerning the "feel" of the elbow support when performing usual tasks on the organization chair.

Notes

## Self Assessment

Fill in the blanks:

12. After deciding on the prototyping strategy the .................................... must be chosen.
13. In prototyping, tools should permit .................................... to rapidly make variations and swiftly see the results.
14. The prototype can be simply developed with tools of fourth generation languages (4GL's) and with the aid of .................................... tools.
15. Prototyping strategy is a form of.................................... .

## 9.6 Summary

- Prototyping is the process of creating, developing and refining a working model of the final operational system.
- Prototyping 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.
- Designing and constructing a scaled-down but functional edition of a preferred system is the process called Prototyping.
- A prototype is a working system – not just an idea on paper that is developed to test ideas and assumptions about the new system.
- The purpose of prototyping is to test out assumption made by analyst and users about required system features.
- The dissimilarity between a prototype model and an definite information system is that, a prototype will not comprise the error inspection, input data validation, security and processing totality of a completed application.
- Prototyping is useful when the requirements are not well defined.
- After deciding on the prototyping strategy the prototyping tool must be choosen. The objective is to fit the tool to the needs of the system under development, the abilities of the developers, and the requirements of the users.

## 9.7 Keywords

***Executability***: The prototype should be as authentic to the real system as developers require to enlarge reliability of information collected.

***Lifecycle Support***: Prototyping tools can aid with all phases of development even away from systems analysis.

***Prototyping:*** The process of creating, developing and refining a working model of the final operational system.

***Team Design***: Team design should sustain groups of people functioning together either concurrently or asynchronously, or distantly if these are necessary by the project.

## 9.8 Review Questions

1. What is prototyping? Illustrate the working of prototype application.
2. Enlighten the purpose of prototyping.
3. Discuss the characteristics of prototyping.
4. Illustrate the steps required for the development of a prototype application.
5. Explain the various uses of prototyping.
6. Before starting the system design work, user and system analyst sit together and discuss to identify the requirements. Comment.
7. Illustrate the objective used for selecting prototyping tools.
8. Explicate the concept of system prototype method.
9. Depict the set of features considered when choosing a prototyping tool.
10. Elucidate the circumstances in which prototyping is used for identifying requirements for the software.

### Answers: Self Assessment

1. Prototyping
2. construction
3. user
4. tools
5. functioning
6. information
7. test
8. requirements
9. systems analyst
10. software
11. well defined
12. prototyping tool
13. developers
14. Computer Aided Software Engineering (CASE)
15. Rapid Application Development (RAD)

## 9.9 Further Readings

*Books*

D R Jeffery; M J Lawrence, [Sydney, N.S.W. ; Englewood Cliffs, N.J.], *Systems Analysis and Design*, Prentice-Hall of Australia, 1985, ©1984.

James C Wetherbe, St. Paul , *Systems Analysis and Design*, West Pub. Co., ©1988.

S.A. Kelkar, *Structured System Analysis and Design*, Prentice Hall of India.

*Online link*

http://www.cs.virginia.edu/~sullivan/publications/options-prototyping.pdf

# Unit 10: Computer Aided System Tools

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the concept of CASE Tools
- Illustrate the role of case tools
- Discuss the categories of case tools

## Introduction

Some Tools used in System Development encloses Project management application, Drawing/graphics application, Word processor/text editor, Computer-Aided System Engineering (CASE) tools, Integrated Development Environment (IDE), Database management application, Reverse-engineering tool, and Code generator tool.

In this unit, you will understand the concept of CASE Tools, role of case tools, and categories of case tools.

## 10.1 CASE Tools

Computer Aided Software Engineering (CASE) tools are steadily becoming well-liked for the growth of software as they are civilizing in the potentials and functionalities and are establishing to be advantageous for the expansion of quality software. But, what are the CASE tools? And how do they assist the process of growth of software?

CASE tools are the software engineering tools that allow mutual software development and preservation. CASE tools assist almost all the stages of the software development life cycle like analysis, design, etc., involving umbrella activities like project management, configuration management etc.

There are two generations of CASE tools. The first generation of CASE tools can be widely categorized into three groups: information generations or 4GLs, front-end design/analysis tools.

The various 4GL products comprise the following: Report generators, Query languages, DBMS front ends and modeling languages. Most undergo from deficiencies: they are tied too closely to a proprietary database system thus providing a very limited solution; they are functionally too scrawny to be more than a building block in a bigger application solution or they are not simply incorporated with current production system and data.

Design tools assist a user to sketch blue prints or design diagrams depending on some prearranged methodology. Usually, high-level design documentation is offered automatically.

*Notes* The understandable flow in design tools is that they are separate and their results are not simply incorporated into the consequent phases of the life cycle.

*Caution* A most important inadequacy of the first generation CASE products is their incapability to ridge the gap among design and application generation.

The second generation CASE tools are developed into two major categories: life cycle automation and solution software.

The first category of tools aimed at data processing work to offer common solutions to their troubles. The second category of tools is intended at analyst or application expert, in a restricted field of application, to offer quick solution to the end user. Electronic spread sheets display such tools in the very limited field of financial analysis.

The CASE tools follow a usual process for the growth of the system.

*Example:* For generating data base application, CASE tools may assist the following development steps:

- Creation of data flow and entity models
- Instituting a relationship between requirements and models
- Development of top-level design
- Development of functional and process description
- Development of test cases.

The CASE tools on the foundation of the above stipulations can help in automatically producing data base tables, forms and reports, and user documentation. Therefore, the CASE tools:

- Assist contemporary development of software systems, they may recover the quality of the software
- Aid in automating the software development life cycles by use of certain standard methods
- Produce an organization wide environment that minimizes cyclical work
- Assist developers to focus more on top level and more imaginative problem solving tasks
- Support and perk up the quality of documentation (Completeness and non- ambiguity), testing process offers automated checking), project management and software maintenance.

Notes

Most of the CASE tools comprise one or more of the following types of tools:

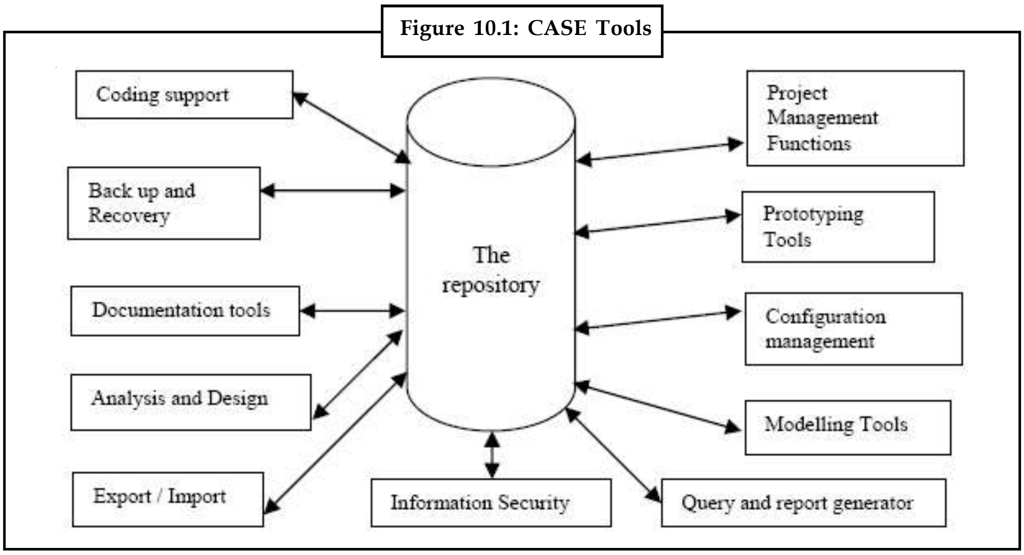
- Repository to amass all diagrams, forms, models and report definitions etc.
- Diagramming tools
- Screen and report generators
- Code generators
- Documentation generators
- Reverse Engineering tools (that take source code as input and generate graphical and textual representations of program design-level information)
- Re-engineering tools (that take source y and/or performance).

Some necessary traits that must be assisted by CASE tools in addition to the above are:

- It should have Security of information. The information may be visible/ changeable by authorized customers only.
- Version Control for various products.
- A value to Import/Export information from different external resources in a compatible fashion.
- The process of Backup and Recovery as it encloses very valuable data.

Figure 10.1 Case Tools shows an environment having CASE.

Figure 10.1: CASE Tools



*Did u know?* The CASE tools usually, assist standard software development methods like Jackson Structure programming or structured system analysis and design method.

*Task* Make distinction between first generation CASE tools and second generation CASE tools.

## Self Assessment

Fill in the blanks:

1. .............................. are the software engineering tools that allow mutual software development and preservation.
2. .............................. tools assist a user to sketch blue prints or design diagrams depending on some prearranged methodology.
3. .............................. generation of CASE tools can be widely categorized into three groups: information generations or 4GLs, front-end design/analysis tools.

## 10.2 Role of CASE Tools

CASE tools play a chief role in the following activities:

- Project management
- Data dictionary
- Code generation
- User interface design
- Schema generation
- Creation of metadata for data warehouse
- Reverse engineering
- Re-engineering
- Document generation
- Version control
- OO analysis and design
- Software testing
- Data modeling
- Project scheduling
- Cost estimation

CASE technology has effected in important improvements in quality and efficiency. All characteristics of software engineering process are not assisted by today's CASE tool. Most of the CASE tools offer good support for data modeling, object oriented design and programming.

Additionally, they more or less, support testing and maintenance.

*Caution* An ultimate CASE tool must assist all features of system development such as analysis, design, implementation, testing and maintenance.

## Self Assessment

Fill in the blanks:

4. An ultimate CASE tool should assist all features of .............................. such as analysis, design, implementation, testing and maintenance.

Notes

5. Most of the CASE tools offer good support for data modeling, .............................. design and programming.

## 10.3 Categories of CASE Tools

Depending on their activities, sometimes CASE tools are classified into the following categories:

1. Upper CASE tools
2. Lower CASE tools
3. Integrated CASE tools.

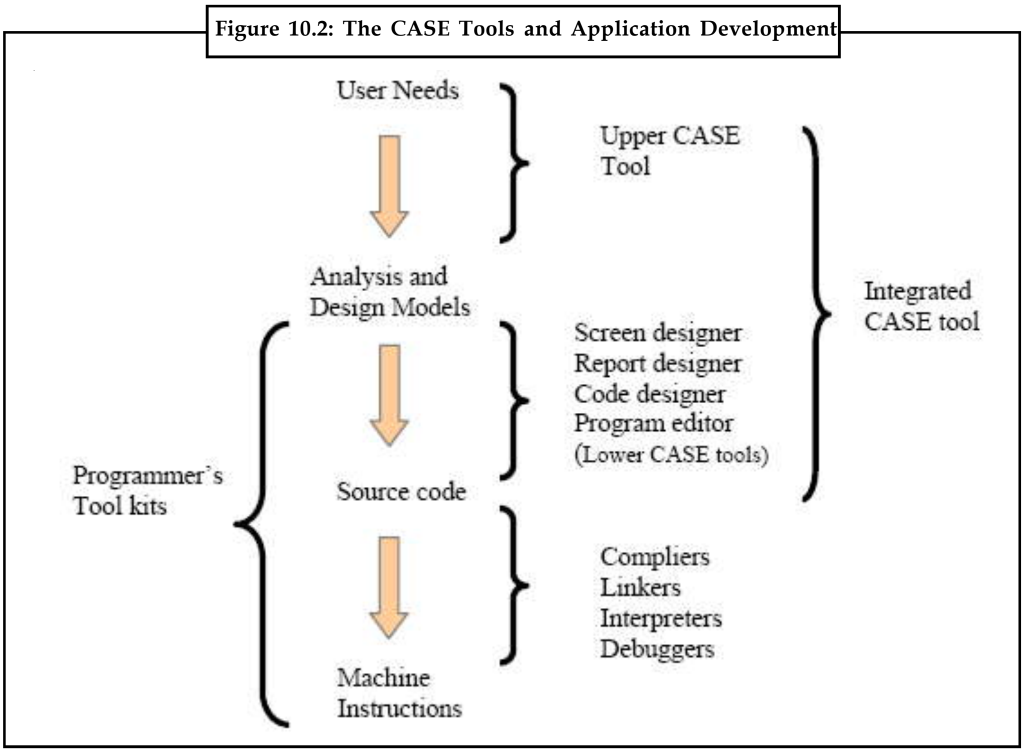
***Upper CASE:*** Upper CASE tools chiefly concentrate on the analysis and design phases of software development. They comprise tools for analysis modeling, reports and forms generation.

***Lower CASE:*** Lower CASE tools support execution of system development. They comprise tools for coding, configuration management, etc.

***Integrated CASE Tools:*** Integrated CASE tools help in offering linkages among the lower and upper CASE tools. Hence creating a consistent environment for software development where programming by lower CASE tools may robotically be produced for the design that has been generated in an upper CASE tool.

Figure 10.2 displays the positioning of CASE tools in a Software Application development.

**Figure 10.2: The CASE Tools and Application Development**



*Task* How does an integrated CASE tool offers linkages between the lower and upper CASE tools.

### Self Assessment

Fill in the blanks:

6. .............................. tools chiefly concentrate on the analysis and design phases of software development.

7. .............................. tools support execution of system development.

8. .............................. tools help in offering linkages among the lower and upper CASE tools.

## 10.4 Need of CASE Tools

The software development process is luxurious and as the projects turn out to be more multifaceted in nature, the project implementations develop into more challenging and expensive.

The CASE tools offer the incorporated homogenous environment for the expansion of complex projects. They permit creating a shared repository of information that can be accessed to diminish the software development time. The CASE tools also supply the environment for monitoring and handling projects such that team leaders are able to administer the complex projects.

Particularly, the CASE tools are usually organized to:

- Reduce the cost as they computerize many recurring manual tasks.
- Reduce development time of the project as they assist standardization and evade repetition and reuse.
- Develop better quality complex projects as they supply greater consistency and harmonization.
- Create good quality documentation.
- Create systems that are maintainable due to proper control of configuration item that support traceability needs.

There are some disadvantages of CASE tools. These are:

- Complex functionality
- Many project management troubles are not amenable to automation. Thus, CASE tools cannot be used in such cases.

*Notes* Please note that CASE tools cannot do the following:

1. Automatic growth of functionally relevant system
2. Force system analysts to follow a prearranged methodology
3. Transform the system analysis and design process.

### Self Assessment

Fill in the blanks:

9. The CASE tools offer the .............................. homogenous environment for the expansion of complex projects.

10. The CASE tools supply the environment for .............................. and handling projects.

Notes

## 10.5 Characteristics of Successful CASE Tools

A CASE tool must contain the following characteristics in order to be used competently:

- ***A standard methodology:*** A CASE tool must assist a standard software development methodology and standard modeling methods. In the present situation most of the CASE tools are moving towards UML
- ***Flexibility:*** Flexibility in utilization of editors and other tools. The CASE tool must propose flexibility and the option for the user of editors' development environments.
- ***Strong Integration:*** The CASE tools should be incorporated to support all the levels.

*Example:* If a variation is made at any stage, for instance, in the model, it should get reflected in the code documentation and all associated design and other documents, thus offering a consistent environment for software development.

- ***Integration with testing software:*** The CASE tools must offer interfaces for automatic testing tools that take care of regression and other types of testing software under the modifying requirements.
- ***Support for reverse engineering:*** A CASE tools must be able to produce complex models from already developed code.
- ***On-line help:*** The CASE tools offer an online tutorial.

### Self Assessment

Fill in the blanks:

11. The CASE tools must offer .............................. for automatic testing tools that take care of regression and other types of testing software under the modifying requirements.
12. A CASE tool must be able to produce complex .............................. from already developed code.

## 10.6 Use of CASE Tools by Organizations

The following are some of the manners in which CASE tools are utilized:

- ***To allow single design methodology:*** CASE tools assist organization to regulate the development process. It also allows coordinated development. Integration turns out to be easy as common methodology is adopted.
- ***Rapid Application Development:*** Organizations utilize CASE tools to perk up the speed and quality of system development.
- ***Testing:*** CASE tools aid to ease and advanced testing process via automated checking and simplify program maintenance.
- ***Documentation:*** In traditional software development procedure, the quality of documentation at various stages is based on the individual. CASE tools perk up the quality and consistency of documentation at a variety of stages of SDLC.

*Did u know?* CASE Tools makes sure the wholeness of the documentation.

- ***Project Management:*** It enhances project management activity and to some degree automates different activities concerned in project management.
- ***Productivity and reduction of cost:*** Use of CASE tools makes the software simple to preserve and thus decrease the preservation costs. Automation of different activities of system development and management processes enhances efficiency of the development team.

## Self Assessment

Fill in the blanks:

13. CASE tools aid to ease and advanced testing process via .............................. and simplify program maintenance.
14. CASE tools enhance .............................. activity and to some degree automate different activities concerned in project management.
15. Use of CASE tools makes the software simple to preserve and thus decrease the .............................. costs.

## 10.7 Summary

- CASE tools are the software engineering tools that allow mutual software development and preservation.
- The CASE tools usually, assist standard software development methods like Jackson Structure programming or structured system analysis and design method.
- The various 4GL products comprise the following: report generators, query languages, DBMS front ends and modeling languages.
- Design tools assist a user to sketch blue prints or design diagrams depending on some prearranged methodology.
- The CASE tools can help in automatically producing data base tables, forms and reports, and user documentation.
- An ultimate CASE tool should assist all features of system development such as analysis, design, implementation, testing and maintenance.
- Upper CASE tools chiefly concentrate on the analysis and design phases of software development.
- Lower CASE tools support execution of system development.
- Integrated CASE tools help in offering linkages among the lower and upper CASE tools.
- The CASE tools offer the incorporated homogenous environment for the expansion of complex projects.

## 10.8 Keywords

***CASE Tools:*** CASE tools are the software engineering tools that allow mutual software development and preservation.

***Integrated CASE Tools:*** Integrated CASE tools help in offering linkages among the lower and upper CASE tools.

***Lower CASE Tools:*** Lower CASE tools support execution of system development.

***Upper CASE Tools:*** Upper CASE tools chiefly concentrate on the analysis and design phases of software development.

## 10.9 Review Questions

1. What are CASE Tools? Enlighten the concept of CASE Tools.
2. Illustrate the concept of two generations of CASE tools.
3. Depict the development steps used in CASE Tools.
4. List various types of tools enclosed under CASE Tools.
5. Explicate the different types of roles played by CASE Tools.
6. Make distinction between Upper CASE tools and Lower CASE tools.
7. Illustrate the need of CASE tools.
8. Elucidate the disadvantages of CASE tools.
9. Describe various characteristics of successful CASE Tools in order to be used competently.
10. Explicate the methods used in which CASE tools are utilized.

### Answers: Self Assessment

1. CASE tools
2. Design
3. First
4. system development
5. object oriented
6. Upper CASE
7. Lower CASE
8. Integrated CASE
9. incorporated
10. monitoring
11. interfaces
12. models
13. automated checking
14. project management
15. preservation

## 10.10 Further Readings

*Books*

D R Jeffery; M J Lawrence, [Sydney, N.S.W. ; Englewood Cliffs, N.J.], *Systems Analysis and Design,* Prentice-Hall of Australia, 1985, ©1984.

S.A. Kelkar, *Structured System Analysis and Design*, Prentice Hall of India.

James C Wetherbe, St. Paul, *Systems Analysis and Design*, West Pub. Co., ©1988

*Online link*

http://www.selectbs.com/analysis-and-design/computer-aided-software-engineering-case-tool

Notes

# Unit 11: Analysis to Design Transition

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the objectives of analysis to design transition
- Discuss the features of analysis to design transition
- Explain the elements of design

## Introduction

The objective of systems design is to intend a system that is effective, dependable, and maintainable, while fulfilling the requirements and restrictions defined throughout the systems analysis phase. In this unit, you will understand the objectives and features of analysis to design transition. We will also discuss elements of design such as inputs, outputs, files, etc.

## 11.1 Transition from Analysis to Design

### 11.1.1 Objectives

A reasonable transition from systems analysis to systems design is mainly reliant on an accurate and explicable system requirements document, which is the preliminary point for the systems design stage. At this point, you have a logical design of the information system, which describes the functions and traits of the system, in addition to the relationship between its components. The logical design defines *what* must occur, and the physical design of the information system illustrates *how* it will be fulfilled physically. The physical design is a plan for the actual accomplishment of the system, and is formed throughout the systems design stage of the SDLC. The objective of systems design is to intend a system that is effective, dependable, and maintainable, while fulfilling the requirements and restrictions defined throughout the systems analysis phase.

### 11.1.2 Features

#### Controlling the Transition

Review the scope of analysis in opposition to the scope of design. The transition from analysis to design needs that you comprehend the dissimilarity between what was modeled in analysis and what you will represent in design. During project initiation and problem analysis you continued under the supposition that the "system" you were modeling had not anything at all to do with software and hardware.

#### Problem Domain Definition

The functionality (*use case model* ), the assets (*object model* ), and the communication of the resources to sustain the functionality (*sequence* and *collaboration* diagrams) would survive whether or not you ever offered automation.

*Example:* In the ticket sales system, you recognized the requirement to set up the seating, set up shows, price the seating in the shows, and sell tickets.

There is not anything technological regarding these functions. Actually, they have been performed physically for centuries.

***Use case model :*** The UML model used to symbolize clients' expectations regarding how they will cooperate with the system.

***Object model:*** For many people this expression is identical with class diagram. Though, the model in a larger sense also involves the object diagram and package diagram.

***Sequence diagram:*** The UML diagram intended to model the communication between objects overtime. The scope of the diagram is normally a single scenario.

***Collaboration diagram:*** One of the two UML standard communication diagrams intended to model the interaction between objects.

#### Preserve the Problem Domain Model

In reality, the analysis level object model will be the foundation for your database design. Few, if any, of the new objects added throughout design will turn out to be part of the database. Those that do will be added during object design to recover performance, not functionality.

*Caution* Everything you specified in analysis must remain integral as you shift into design.

### Design Layer

Design adds a layer of functionality above the analysis models. This layer is the software that emphasis the utilization of the problem domain resources by means of interfaces, databases, transaction control, and communication that conforms to the use case model. This layer of technology will probably change frequently, but the fundamental problem domain will remain comparatively stable.

*Task* Make distinction between sequence diagram and collaboration diagram.

## Self Assessment

Fill in the blanks:

1. The ........................... design is a plan for the actual accomplishment of the system, and is formed throughout the systems design stage of the SDLC.
2. The objective of systems design is to intend a system that is effective, dependable, and maintainable, while fulfilling the requirements and restrictions defined throughout the ........................... phase.
3. ........................... diagram isone of the two UML standard communication diagrams intended to model the interaction between objects.
4. ........................... layer is the software that emphasis the utilization of the problem domain resources by means of interfaces, databases, transaction control, and communication that conforms to the use case model.
5. ........................... model is the UML model used to symbolize clients' expectations regarding how they will cooperate with the system.
6. Design adds a layer of ........................... above the analysis models.

## 11.2 Elements of Design

The computer system design process is an exercise of specifying "how" the system will work. It is an iterative process which is based on "what" the system will do.

Mainly, following parts have been included in the system design process:

### 11.2.1 Output

The starting point of the design process is the proper knowledge of system requirements which will normally be converted in terms of output.

A major objective of a system is to produce an output that has value to its user. Whatever the nature of the output (goods, services or information), it must be in line with the expectations of the intended user. Inputs are the elements (material, human resources, information) that enter the system for processing. Output is the outcome of processing. A system feeds on input to produce output in much the same way in which a business brings in human, financial and material resources to produce goods and services.

Defining aim is very vital in system work. If we do not know where we want to go, we will not know when we have reached there. We shall be unnecessarily wasting our time and energy in the process. Once we know our aim, we can try to achieve it in the best possible way. The user department has to define these objectives in terms of their needs. These becomes the outputs which the systems analyst keeps into mind.

### 11.2.2 Files

Once the input data is captured in the system, these may have to be preserved either for a short or long period. These data will generally be stored in files in a logical manner. The designer will have to devise the techniques of storing and retrieving data from these files. Files are used to store data. Inputs necessary for the system are stored in files either in terms of isolated facts or in large volumes.

After designing the input and output, the designer begins to pay his attention on the work of file designing or how data should be organized around user requirement. How data are organized depends on the data and response requirements that determine hardware configurations. Selecting from options available for organizing the data. File organization may be sequential, Index sequential, inverted lost or random.

*Did u know?* System analyst is responsible for designing the files and selecting their contents.

Files are the heart of a computer application. The basic terms used to describe a file hierarchy are:

1. ***Data Item:*** A basic or individual element of data is called data item. Each data item is identified by a name and is assigned a value.
2. ***Record:*** The collection of related data items is called a record. It is necessary to distinguish one specific record from another. System analyst select one data item in the record that is likely to be unique in all the records of a file which is used to identify the record for further processing. This item is called the key field or: record key.
3. ***File:*** A collection of related records. Each record in a file is included because it pertains to the same entity.
4. ***Database:*** The highest level in the hierarchy is the database. It is a set of interrelated files for real time processing. It contains the necessary data for problem solving and can be used for several users who are accessing data concurrently.

#### *Types of Files*

There are various types of files in which the records are collected and maintained. They are categorised as:

1. ***Master File:*** Master files are the most important type of file. Most file design activities concentrate here. In a business application these are considered to be very significant because file that contain the essential records for maintenance of the organisation's business.
2. ***Transaction File:*** A transaction file is a temporary file used for two purposes. First of all, it is used to accumulate data about events as they occur. Secondly, it helps in updating Master file to reflect the result of current translations.
3. ***Table File:*** A special type of master file is included in many systems to meet specific requirements where data must be referenced repeatedly. Table files are permanent files containing reference data used in processing transactions, updating master file or producing

output. As the name implies, these files store reference data in tabular form. Table files conserve memory space by storing data in a file, that otherwise would be included in programs or master file records.

4. ***Report File:*** Report files are collected contents of individual output reports or documents produced by the system. They are created by the system where many reports are produced by the system but printer may not be available for all reports. The process of creating it is known as spooling which means that output that cannot be printed when it is produced, is spooled into a report file. Then, depending on the availability of printer, the system will be instructed to read the report file and print the output on the printer.

5. ***Backup File:*** It is a copy of master, transaction or table file made to ensure a copy is available if anything happens to the original.

6. ***Archival File:*** These files are copies of files made for long term storage of data that may be required at a much later date. Usually, these files are stored far away from the computer centre so that they cannot be easily retrieved for use.

7. ***Dump File:*** This is a copy of computer field data at a particular point of time. This may be a copy of master file to be retained to help recovery in the event of a possible corruption of the master file or it may be part of a program in which error is being traced.

8. ***Library Files:*** Library file generally contains application programs, utility programs and system software packages.

## File Organization

A file is organised to ensure that records are available for processing. Before a file is created the application to which the file will be used must be carefully examined. Clearly, a fundamental consideration will concern the data to be recorded on the file. But an equally important and less obvious consideration concerns how the data are to be placed on the file. There are four common methods of organising files.

(a) Serial Organisation

(b) Sequential Organisation

(c) Random file Organisation

(d) Indexed Sequential Organisation

**Table 11.1: An Example of Database File Showing Columns and Fields**

| | FIELDS | | | | | |
|---|---|---|---|---|---|---|
| | **ID. No.** | **Name** | **Address** | **Class** | **Section** | **Roll No.** |
| **RECORDS** | 001 | Suchitra | 218, MIG, Rajouri Garden | 12th | A | 10 |
| | 002 | Komal | E-18, Tarogre Garden Ext. | 6th | B | 23 |
| | 003 | Adhish | F-74, Naraina Vihar | 11th | A | 31 |
| | 004 | Mohit | F-280, Vikaspuri | 10th | C | 21 |
| | 005 | Vicky | WZ – 143/4C, New Mahavir Nagar | 8th | B | 12 |

We are discussing below the above methods of organizing files.

### Serial Organization

In Serial Organization, the records are stored one after another without any logical order. For example, in table 11.1 with serial organization, the Id. No 002 can be stored after Name Konal as shown in Table 11.1. So, the records need not be stored in ascending or descending order of any field. The records can be added only at the end of a file. The retrieval of a particular record can be done only by reading its preceding records from the beginning of file.

*Advantages of Serial Organization*

The Serial Organization is the simplest method of file organization. Serial Organization is commonly used for storing the transaction data of those applications where records are accessed in the order of their storage.

*Disadvantages of Serial Organization*

The main disadvantages of serially organised files are given below:

- The retrieval of records is very slow in these files.
- The deletion and modification of a particular record is difficult and time-consuming.

### Sequential Organization

In sequential file organisation, the records are arranged sequentially in a predetermined order. For example, in table 11.1, the record with Id. No 003 will follow the record with Id. No. 002. In sequential file organisation the records are stored and sorted in continuous blocks on tape or disk. Within each block, the records are sorted in ascending or descending order to simplify the logic for accessing the data.

*Advantages of Sequential Organization:* The sequential file organisation is a traditional and very common way of data storage method due to following two main reasons:

- It is best suited for applications using batch processing; and
- The sequential files can be stored on magnetic tapes which are the least expensive storage devices.

*Disadvantages of Sequential Organization:* The sequential files have following main disadvantages:

- The sequential files are very inefficient when the number of records is high because for searching a particular record, on average, one-half of the records must be processed.
- When a less number of records are required to be modified or deleted, then the processing becomes time-consuming. This is due to the fact that the magnetic tapes do not permit a record to be updated at same location. The entire file must be read first and then rewritten on another tape.

### Random File Organization

As we have discussed, the sequential files have many disadvantages. Their major disadvantage is that it is necessary to read records sequentially. However, it is desired that records can be accessed directly or randomly. The file organisation in which records are organised randomly and need not be in sequence is known as Random or Direct File Organisation. In random files, the records are stored at the storage location numbers (address) which can be determined by the record key values.

*Advantages of Random File Organization*

The random file organisation has the following advantages:

- It provides the fastest direct access to the records.
- It is best suited for applications requiring online inquiry and updating such as Railway Ticket Bookings and Bank Account Statements Program.

*Disadvantages of Random File Organisation*

Although, random file organisation is considered best for online application, it has certain drawbacks. A major problem with hashing technique in random file organisation is the occurrence of synonymous as SRA. Hashing can produce identical storage location (SRAs) for two different records. These record keys are called synonyms.

### Indexed Sequential Organization

Indexed sequential files are based on the concept of an index. For instance, this book contains different topics which are organised in certain sequence and it also contains an index at the end. The page number of any topic can be searched by just looking through this index. Similarly, in Indexed Sequential Organisation, the records are stored sequentially on a data file, but a second file (Index File) also exists which makes it possible to access any record directly. So, the records of indexed sequential files can be accessed sequentially as well as directly. The files for indexed sequential organisation must be stored on a Direct Access Storage Device (DASD). The floppy and hard disks are commonly used as direct access storage devices.

To understand the indexed sequential organisation, let us first discuss the different areas of disk storage. The data is stored in following three areas of storage:

(i) ***Prime Area:*** When an indexed sequential file is originally created, the records are written on an area, called prime area. This area contains records stored in logical sequence of key field. The prime area is similar to a sequential file.

(ii) ***Overflow Area:*** This area contains those records which can be added later on and cannot be stored in prime area.

(iii) ***Index Area:*** This area contains the values of key fields of records and their storage locations (address) on the disk.

*Advantages of Index Sequential Files*

The indexed sequential files have the following advantages:

- The indexed sequential files provide the user with very powerful tool. The files can not only be processed sequentially but also randomly.
- The searching is very fast in these files because the file management system accesses the data records in the order of index value. This technique is called the Indexed Sequential Access Method (ISAM). Therefore, these files are also referred as ISAM files.
- When records are updated or inserted, unlike sequential files, there is no need to rewrite the entire file.

*Disadvantages of Index Sequential Files*

The indexed sequential files have the following disadvantages.

- These files must be stored on a direct access storage device such as floppy disk, hard disk, etc. and cannot be stored on magnetic tapes.
- These files require regular maintenance otherwise the records added in end overflow area are not incorporated into the main file.

### 11.2.3 Database Interaction

An integrated approach to file design is the database. The general theme is to handle information as an integrated whole, with a minimum of redundancy and improved performance.

*Notes* Various software techniques are applied to manipulate, describe and manage data. Irrespective type of data structure used, the main objectives of database are accuracy and integrity, successful recovery from failure, privacy and security of data.

A database is defined as a system, used to record and maintain data. The most important point to understand database is that database contains data and not necessarily information.

*Example:* Some of the examples of the database are as follows:

1. The personal records of all the employees of a company.
2. The vouchers, bills receipts and other financial data of an organization.
3. The details of students and their marks obtained for preparing their result in a school.

In the above examples, the data can have some meaning or no meaning in itself until it is processed to get useful information.

In technical words, a database may be defined as follows:

Database a collection of interrelated data, which can be used by one or more applications so that it can be integrated and stored in a shared organized way so that it has a controlled redundancy, consistency and integrity with a provision of data independence.

#### *Objectives of Database*

All organizations need to collect, store and process data for their functions. The database is used to store and process data for providing useful information to the organization. The database must have the following basic objectives:

1. ***Centrally Controlled:*** A database must be centrally controlled. It is possible that the data of a particular system may be spread in different branches of a company but it must be controlled from one central location.

   ***Advantages of Centrally Controlled Database:*** The major advantages of the centrally controlled database are as follows:

   (i) Organized way: If the database is centrally controlled, then the data can be stored, processed, modified or accessed in an organized way.

(ii) To follow standard rules: In a centralized database, the standard rules can be followed in storing different items. The interdepartmental agreement is required for standardizing the data formats and definition. If it is not done, then the data can become incompatible when interchanging it between different applications.

(iii) Non redundancy and consistency: Many of the different applications generally use the same set of data.

*Example:* The inventory and invoicing application, both can use, items data as an input.

If separate data would be stored for both applications, there would be a redundancy (duplication) of stored data. The redundant data has many drawbacks. First of all, there would be a wastage of disk space and secondly, most important, if there is any modification, deletion or addition is required, that has to be done twice for both applications. In that case, the data can become inconsistent. The inconsistent data means that the two kinds of applications will be using a non-matching data. Therefore, a database must be organized in such a way that the redundancy and inconsistency can be avoided as much as possible. In reality, some redundancy exists in many applications, basically due to two reasons - first, to simplify logic of the program for accessing the data; and secondly, to save the data as backups so as to recover it if lost accidentally. Therefore, it is advisable to control the redundancy to minimal level so that it does not cause data for becoming inconsistent and also does not waste too much disk space.

(iv) *To maintain integrity:* Integrity means to ensure that the database contains accurate and correct data. If the database is centralized then the proper validation procedures can be applied for checking whether the data are being entered in different applications is valid or not.

(v) *Security and privacy:* If the database is centrally controlled, then the security checks can be applied to prevent the data from unauthorized access. The same data can be restricted to access by different users to maintain privacy and security.

2. ***Logically organized:*** The database must be organized in a logical manner.

*Example:* If the user wants to see the name of person whose basic salary is greater than 5000 and is working in EDP department, then the database must be organized in that logical order so that the data can be accessed faster.

3. ***Integrated:*** The database must be integrated for many applications of different departments in an organization. In an integrated database system, the same collection of data is available for as many applications as possible.

*Example:* The data required for Financial Accounting, Invoicing and Inventory System are generally interrelated.

If the database is not designed in an integrated way, then the data required for different applications would be entered separately and there would be a lot of duplication of data and hence inconsistency in data storage.

4. ***Shared:*** In multi-user applications, it is expected that the database is designed such that the data can be shared or accessed by different users.

*Did u know?* The sharing of data is possible only if the database is integrated.

5. ***Data Independence:*** The most important objective of database is the provision of data independence. Most of the present day applications are generally data-dependent. It means that the logic of the programming depends on the organisation of the database and hence the storage structure of the data files. If the storage structure of the data files is to be changed, then the program should also be modified accordingly. However, the database should be designed such that, if the data storage structure is needed to be changed in future, then there should not be any need to change the programs logic or coding. It means the programming should be independent of data storage. This feature of database is known as data independence and, is the primary demand of present day software industry.

*Caution* An organization must have an accurate data and information for effective decision-making.

## 11.2.4 Input

Once the output requirements have been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs. The basic documents in which these data are available need to be identified. If necessary, these documents may have to be revised or new documents may have to be introduced.

Sometimes, it may happen that the required information may not be readily available in the proper form. This may be because the existing forms are not properly designed. If the information is vital to the system, we should make all possible efforts to make it available. The essential elements of inputs are:

(a) Data should be accurate. If data is not accurate, the outputs will be wrong.

(b) Data should be obtained in time. If data is not obtained in time, the entire system falls into arrears.

(c) The inputs must be available in proper format.

## 11.2.5 Control

The control design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc. This will ensure that the system is functioning as per plan.

## 11.2.6 Procedures

This step involves specifications of how processing will be performed. In this, there are two aspects:

- Computer procedure
- Non-computer procedure

The computer procedure will specify what functions will be carried out on computer, what will be different programs and in what sequence the programs will be run. The non-computer procedure will specify the manual procedures for feeding input data, receiving outputs etc.

*Task* Make distinction between computer procedure and non-computer procedure.

### 11.2.7 Program Specifications

The result of the system design process is a document known as "system specification". The complete details of design about the proposed system have been included in it. It serves as a blue print and helps in developing and implementing the new system. This also forms the primary documentation on which the system maintaining persons will fall back upon after the system is in use. Later on, this document is normally divided into different parts for easy reference. Thus, we get system manual, user manual, operational manual. Since the system specifications are prepared as a plan, it becomes sometimes necessary to modify it after taking into consideration the practical difficulties or bottlenecks or errors found during later stages of development. System specifications should include all the details necessary to implement the system and to understand the whole working of the system.

Generally, these steps as mentioned above are interdependent and some of them may have to be used together and traversed many times until a satisfactory design is prepared. It is just like the situation of "Two-step forward-one step backward" kind. In Figure 11.1, we have tried to present an ideal situation about the progress of a project. But this situation occurs rarely in day-to-day like. Most of the time progress of a project takes different shape which is shown in Figure 11.2.

The system design process is, therefore, an iterative process where decisions made or changed at one step will have a "ripple effect" on other stops.

*Example:* If an output report is modified, it may necessitate changes in input design, file design and control design.

**Figure 11.1: "Ideal" Project Progress**

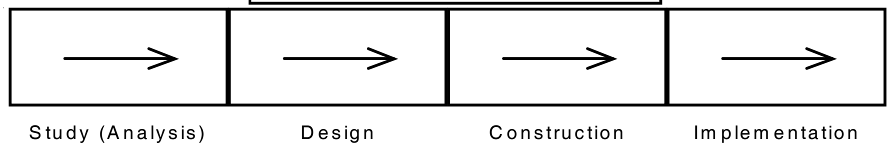
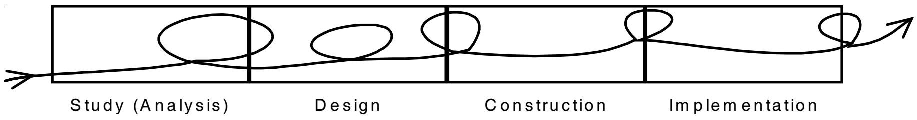


**Figure 11.2: "Reality" of Project Progress**



*Notes* In designing a system, if one tries to design a perfect system, it is his wrong conception. He may perhaps land up with no system at all. What is to be aimed at is the most satisfactory and operable design of a system and then gradual improvements over time.

## Self Assessment

Fill in the blanks:

7. A major objective of a system is to produce an ............................ that has value to its user.

8. ............................ files are collected contents of individual output reports or documents produced by the system.

Notes

9. The ........................... is used to store and process data for providing useful information to the organization.

10. The ........................... design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc.

11. The ........................... procedure will specify what functions will be carried out on computer, what will be different programs and in what sequence the programs will be run.

12. System ........................... should include all the details necessary to implement the system and to understand the whole working of the system.

13. ........................... files are copies of files made for long term storage of data that may be required at a much later date.

14. A basic or individual element of data is called ........................... .

15. ........................... file generally contains application programs, utility programs and system software packages.

## 11.3 Summary

- A reasonable transition from systems analysis to systems design is mainly reliant on an accurate and explicable system requirements document, which is the preliminary point for the systems design stage.
- The objective of systems design is to intend a system that is effective, dependable, and maintainable, while fulfilling the requirements and restrictions defined throughout the systems analysis phase.
- The starting point of the design process is the proper knowledge of system requirements which will normally be converted in terms of output.
- After designing the input and output, the designer begins to pay his attention on the work of file designing or how data should be organized around user requirement.
- An integrated approach to file design is the database. The general theme is to handle information as an integrated whole, with a minimum of redundancy and improved performance.
- The database is used to store and process data for providing useful information to the organization.
- Once the output requirements have been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs.
- Procedures involves specifications of how processing will be performed.

## 11.4 Keywords

***Archival File***: Copies of files made for long term storage of data that may be required at a much later data.

***Backup File:*** A copy of master, transaction or table files made to ensue a copy is available if anything happens to the original.

***Control:*** The control design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc.

***Database:*** A set of interrelated files for real time processing.

***Data Item:*** A basic or individual element of data.

***File:*** A collection of related records.

***Master File:*** Files that contain the essential records for maintenance of the organization's business.

***Record:*** The collection of related data items.

***Report File:*** Collected contents of individual output reports or documents produced by the system.

***Table File:*** Permanent files containing reference data used in processing transactions, updating master file or producing output.

***Transaction File:*** A temporary file used to accumulate data about events as they occurs. It helps in updating master file to reflect the result of current translations.

## 11.5 Review Questions

1. Illustrate the objectives of Analysis to Design Transition.
2. Explicate different features used in Transition from analysis to design.
3. What are the various parts of system design process? Illustrate.
4. Make distinction between input design and output design.
5. Compare and contrast between back up files and archival files.
6. Depict the concept of file design.
7. Describe how database is considered as an integrated approach to file design.
8. Discuss essential elements of inputs.
9. Exemplify the steps used in program specification.
10. A database must be organized in such a way that the redundancy and inconsistency can be avoided as much as possible. Comment.

### Answers: Self Assessment

1. physical
2. systems analysis
3. Collaboration
4. Design
5. Use case
6. functionality
7. output
8. Report
9. database
10. control
11. computer
12. specifications
13. Archival
14. data item
15. Library

Notes

## 11.6 Further Readings

*Books*

D R Jeffery;  M J Lawrence, [Sydney, N.S.W. ; Englewood Cliffs, N.J.], *Systems Analysis and Design*, Prentice-Hall of Australia, 1985, ©1984.

James C Wetherbe, St. Paul, *Systems Analysis and Design*, West Pub. Co., ©1988.

Lars Skyttner, *General Systems Theory*, World Scientific.

Ludwig von Bertalanffy, *General System Theory: Foundations, Development, Applications*, New York : George Brazillier, 1969.

S.A. Kelkar, *Structured System Analysis and Design*, Prentice Hall of India.

*Online link*

http://www.peter-lo.com/Teaching/CS211/L06.pdf

# Unit 12: Design of Computer Output and Input

Notes

**CONTENTS**

## Objectives

After studying this unit, you will be able to:

- Understand the concept of computer output design
- Discuss the Objectives of computer output design
- Identify the needs and types of computer output design
- Understand the concept of design input and control
- Discuss the concept of capturing input data
- Explain the input validation

## Introduction

Output design considers the content, the frequency, the format, medium and the distribution of output. Once the analysis and design of the system has been done, it would be necessary to identify the data that are required to be processed to produce the outputs. In this unit, you will understand various concepts of design of computer output such as objectives, needs, and types of computer output design and also discuss design input and control concepts such as input validation, etc.

Notes

## 12.1 Design of Computer Output

Presenting the data processed by a computer based information system in an attractive and usable form has become very essential these days. Success and acceptance of a system to some extent depends on good presentation. Many new output devices are being introduced in the market because of recent development in computer technology. System analyst must be aware of these new technology and try to use these new output devices if possible. Currently, excellent graphic displays are widely available. Speech output systems are also fast emerging.

*Caution* System analyst must know fully how to design output report in an attractive way.

### 12.1.1 Objectives

The system is accepted by the user solely by quality of its output. If the output is not of good quality, user is likely to reject the system. Therefore, an effective output design is the major criteria for deciding the overall quality of system. The output can be of two forms - hard copy (printed report) and screen output.

Output design considers the content, the occurrence, the format, medium and the allocation of output. Content is fairly established during the information analysis and may be only slightly revised or refined during design. Format treats such matters as column and row headings on reports, spacing, graphic displays and so forth. Output frequency may be daily, weekly, monthly or, in interactive systems, continuous. The output medium may be paper, pre-printed forms, mailers, video displays or some combination of these and other media. Standard forms are available to assist in the layout of output specification.

*Did u know?* Distribution instructions tell who is to get the output and, where appropriate, how many copies to be made.

While designing the output, the objective of systems analyst is:

- To determine what information to present.
- To decide whether to display, point or speak the information and select the output medium.
- To arrange the presentation of information in an acceptable format.
- To decide how to distribute the output to intended recipients.

### 12.1.2 Needs

There are three main reasons why outputs from the computer are required. They are:

(i) For communicating to the persons concerned.

(ii) For re-input to the computer for being connected with other data and further processing.

(iii) For permanent storage.

### 12.1.3 Types of Output

The most common outputs of a system are reports, displays on screen, printed forms etc. The outputs also very in terms of their contents, type of stationary, frequency and timing etc. Besides,

**Notes**

due consideration also need to be given as to who will use the output and for what purpose. All these points must be kept in mind while designing outputs so that the objectives of the system are meet in the best possible way. Outputs of a data-processing system can be placed into two categories:

1. ***Application Output:*** These are the outputs desired out of the system to meet its objectives. These are of three types:
   (a) Output as a basis for decision making, required by management for decision making by purposes.
   (b) Output as a requirement to meet a functional objective, such as, Invoices, Excise Gate Pass, Purchase Orders etc.
   (c) Statutory Outputs, required by all organisations to produce a certain amount of reports and forms as required by law. Such as, 'C' forms, '3A' and '6A' forms for provident fund, income tax certificates etc.
2. ***Operating Output:*** These outputs are mainly generated for use of E.D.P Staff and give various indications as to how the system operates.

*Example:* System logs, error messages, status indicators etc., are the examples of such output. These types of output are not concerned for the users.

*Task* Make distinction between application output and operating output.

### 12.1.4 Output Design

We will now discuss the major features of output forms along with guidelines for designing them.

#### Design of Printed Output

After designing input forms, the analyst designs the format of all printed outputs, commonly called as hard copies. There are three common formats of a printed output:

(a) Tabular format (report format);

(b) Form like format; and

(c) Graphic format as illustrated in Exhibit 12.1, 12.2 and 12.3 respectively.

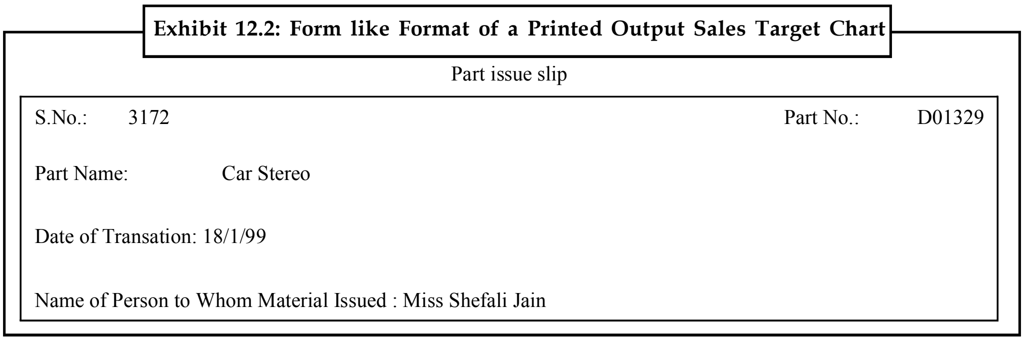
**Exhibit 12.1: Report Format of a Printed Output**

**Stock Summary Report**

| Category Code | Item code | Color Code | Item Name/ Color Name | Size Desc. Size Ratio | Opening Balance | Reced. Qty. | Issued Qty. | Rejected Qty. | Closing Balance |
|---|---|---|---|---|---|---|---|---|---|
| CSI | ES200 | 272 | Large Logo<br>Crew<br>Denim Blue Melange | Men SXL<br>17:33:33:17:0 | 0 | 4600 | 0 | 0 | 4600 |
| | | 40 | Ashgrey Melange | 17:33:33:17:0 | 0 | 760 | 400 | 0 | 360 |
| | | | **ITEM TOTAL** | | 0 | 5360 | 400 | 0 | 4960 |

Notes

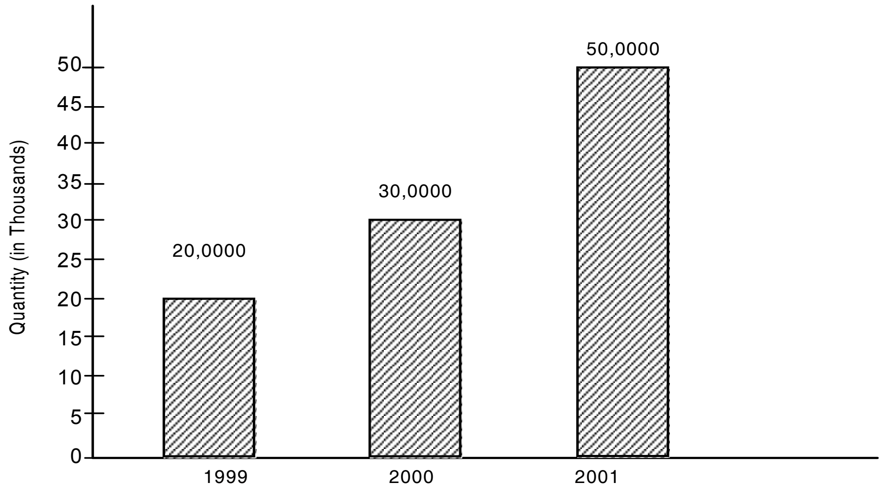
**Exhibit 12.2: Form like Format of a Printed Output Sales Target Chart**



Part issue slip

| | |
|---|---|
| S.No.: 3172 | Part No.: D01329 |
| Part Name: Car Stereo | |
| Date of Transation: 18/1/99 | |
| Name of Person to Whom Material Issued : Miss Shefali Jain | |

**Exhibit 12.3: Graphic Format of a Printed Output**



All the above three formats have their own advantages. The tabular or report format is most suitable to display MIS reports.

> *Notes* The presentation of a printed output is enhanced by using graphic format. Some printed outputs like application forms are designed by using form like formats.

### *Guidelines for Designing Report Formats*

The analyst should design these report formats under following guidelines:

- The width of paper, whether it is 80 column, 132 column or more or less must be specified on the report.
- All reports should have properly aligned titles and column headings.
- The data elements should be properly spaced out according to column headings.
- The important data and titles should be either in bold or underlined.
- The data type, whether it is numeric or character should be specified for each data element by putting 'X' or '9' respectively.
- The size of data element should be specified by repeating 'X' or '9' s.

*Example:* If size of a character data type is 6, it is written as 'XXXXXX'.

- The report must have a page number either on top or bottom of the page.
- All the column totals, major and minor break totals should be specified in the report.
- The footer area and its details should also be included in report design.
- The report has to be printed in normal form or condensed form and should also be specified on the design.
- The name of compatible printers, where the report can be printed, should also be specified.

These are just a few guidelines for designing a good report, however, when a report is designed actually by analyst, he may have many ideas (depending upon his experience) for designing the best quality report.

### *Advantages of Printed Output*

A printed output offers the following advantages to the user:

- A printed output is permanent and is not erased.
- It can be carried anywhere, i.e., it is portable.
- It provides a detailed information.
- Its information cannot be changed by the user.

### Design of Screen Output

A printed output is the basic requirement of any computerized system. However, it is not always possible and even not advised to print each and every output. Therefore, the system should also display all outputs on screen. The screen output offers following advantages over hard copy:

- A screen output provides any information immediately.
- It can be viewed in different formats.
- It is more secure as it can be accessed by only authorized users (printed report can be leaked out).
- Its information can be changed by the user if the user has access to modify it.

The output screens are designed on following important guidelines:

- The screen layout should be simple and may be similar to printed output.
- The number of key strokes must be minimum for displaying any information.
- The output screens may have a multi-window presentation.
- The information displayed on screens need not be as detailed as in printed output.
- It should provide the on-line instructions to the user, which are generally displayed on the bottom line.

These are just a few guidelines for designing a good output screen. However, there can be many new features in a good screen depending upon the experience of analyst and features of DBMS/ language used by the programmer.

## Self Assessment

Fill in the blanks:

1. ............................ design considers the content, the occurrence, the format, medium and the allocation of output.

2. Content is fairly established during the ............................ analysis and may be only slightly revised or refined during design.
3. ............................ Outputs are the outputs desired out of the system to meet its objectives.
4. ............................ Outputs are mainly generated for use of E.D.P Staff and give various indications as to how the system operates.
5. After designing input forms, the analyst designs the format of all printed outputs, commonly called as ............................ .
6. The presentation of a printed output is enhanced by using ............................ format.
7. The ............................ format is most suitable to display MIS reports.

## 12.2 Design Input and Control

Once the analysis and design of the system has been done, it would be necessary to identify the data that are required to be processed to produce the outputs. Input is one of the most expensive phases of the operation of a computerized system and creates sometimes a major problem. Different type of problems with a system can usually be traced back to faulty input design method. Needless to say, therefore, that the input data are the lifeblood of a system and have to be analyzed and designed with utmost care and consideration.

> *Notes* Input design features can ensure the reliability of the system and generate correct reports from the accurate data. The input design also determines whether the user can interact efficiently with the system.

### 12.2.1 Objectives of Input Design

The quality of input design determines the reliability of the system. The basic objectives of input design are:

- Input forms must be complete and accurate.
- They must be designed in such a way that the data entry becomes easier.
- There should not be any inconsistency of data input forms.
- Input forms should be attractive to the user.
- They must be easy to fill out.

### 12.2.2 Capturing Input Data

By keeping the above discussed objectives in mind, the analyst designs the input forms in following steps:

(a) Detailed studying of user-originated documents;

(b) Studying of data dictionary;

(c) Capturing of data items from user-originated documents or data dictionary;

(d) Identification of data items whether they are constant;

(e) Study of Software Requirements Specification (SRS) document;

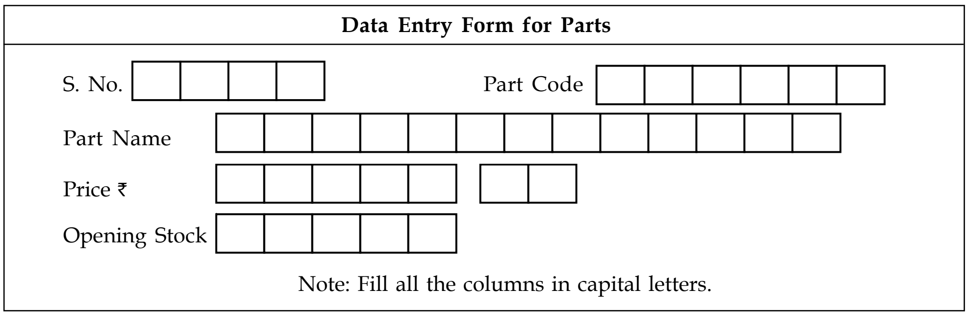
(f) Designing of input forms on the basis of both SRS and user-originated forms.

The layout of an input form is illustrated in Exhibit 12.4.

### 12.2.3 Input Screen Designs

Notes

**Exhibit 12.4: The Layout of an Input Form**



**Data Entry Form for Parts**

S. No. □□□□ Part Code □□□□□□

Part Name □□□□□□□□□□□□□

Price ₹ □□□□□ □□

Opening Stock □□□□□

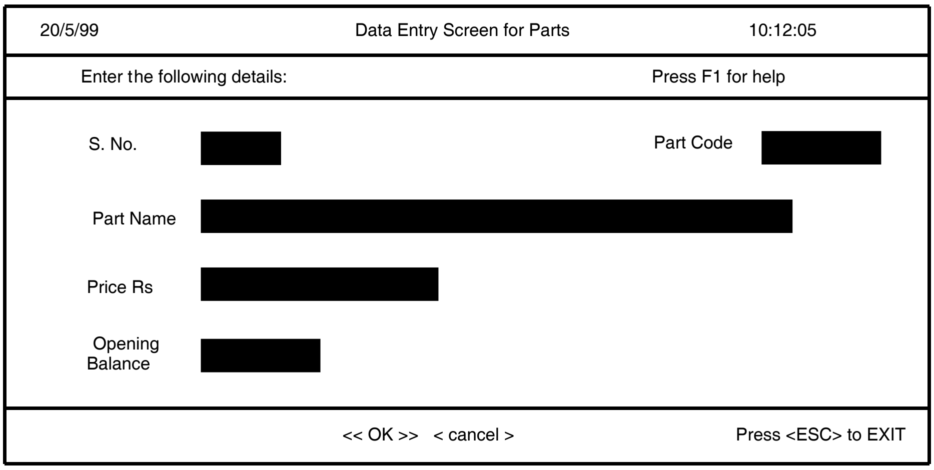
Note: Fill all the columns in capital letters.

After designing input forms, all the screen layouts of the system are designed. For a better user interface, it is most important to design user-friendly screens. A user-friendly screen should have following main features:

- The screen layout should be simple and almost similar to input form.
- It must have interactive dialog (interchange of messages between the computer and the user) during data entry.
- The interface should not fail in case of data entry errors.
- The number of key strokes must be minimum for feeding of any data or for giving any command.
- On-line help screens must be available with all data entry screens.
- The screens must have a multi-window presentation.
- The screens layout of multi-user systems must show interactions between users.

The layout of an input screen is illustrated in Exhibit 12.5.

**Exhibit 12.5: The Layout of an Input Screen**



20/5/99 Data Entry Screen for Parts 10:12:05

Enter the following details: Press F1 for help

S. No.

Part Code

Part Name

Price Rs

Opening Balance

Notes

*Did u know?* The screen layouts provide the way by which user interacts with the computer.

*Caution* The screens should be designed such that the people who do not know computer, must be able to do the data entry.

### 12.2.4 Input Validation

If incorrect data enters the system, it is usually very costly to make the necessary corrections. Also, how expensive would it be to have your operator record a quantity of 100 rather than 10 for a shipment of sport cars? The shipping charges for sending the cars to the customer and then of having them returned would be only one of the costs. While the 90 extra cars were in transit, they would not be a available to other customers (which could result in a loss of sales) or could be damaged. There are many methods which are commonly used to verify data entering the system as input. Some of them are:

(a) ***Key Verification:*** A second operator re-keys the data already recorded. This method is used for verifying data recorded in punched cards or on diskettes and magnetic tapes. Then two floppies are compared to correct record by record which mismatched during comparison after verifying from the original documents. This is most effective method used by Computer Service Bureaus for data validation.

(b) ***Use of Self-checking Numbers:*** The computer can be programmed to reject numbers that have been transposed or have one or more wrong digits. Check digits and self checking numbers routines can be effectively used for numbers in a series, such as student roll numbers, account numbers, part numbers, or invoice numbers are popular for such jobs.

(c) ***Visually displaying and Identifying Characteristics:*** When using a terminal, a part number is entered. Displayed in the VDT is the description of the part, which is then visually confirmed by the operator.

(d) ***Hash Totals:*** Sometimes numbers are added to produce a meaningless total called a hash total.

*Example:* Totaling is made of the quantity of all items purchased. When the records are entered and processed, the hash total is compared to the original total. If the two totals agree, it is an indication that all quantities were entered correctly and all records were processed.

(e) ***Checking between a range of Numbers:*** The numbers on the orders being processed on a given day should fall between, say, 4999 (the last number from the previous day) and 6001 (the next order number that will be on all the orders processed by the next day). If the order number recorded on the input record does not fall within that range, an error message will be generated.

(f) ***Reasonableness Test:*** Based upon past history, some inputs can be checked to see if it is reasonable.

*Example:* Because of long standing company policy, it is unlikely that any employee will have more than 20 hours of overtime. If more than 20 hours of overtime are recorded in an employee's current transaction record, an error message will be generated as the data is being edited. Similarly, in "date of birth" field, it is checked that no date is more than 31, month number is not more than 12, and the year is not more than the current year or current year minus minimum age prescribed.

(g) ***Verification of Codes:*** The pay and fringe benefits are calculated for employees based upon their payroll status, assuming that the valid status code must be either an H' (hourly), 'S' (salaried), 'T' (trainee), or a 'P' (part-time), an error message would be generated if the code used was not on H,S,T or P.

(h) ***Verification of Data type:*** Some input fields should contain only numeric data while others should contain only alphabetic data. The fields can be edited to make certain that only the right type of data is recorded in each field.

(i) ***Verification that Certain Combination of Data Exists:*** All students may be coded with either a 'W' or 'V'. The 'V' denotes a non-work-study student while the 'W' indicates that the student is on work-study. The only valid account numbers for a work-study student are 2155 and 2156. Any other account number for a 'W' coded student is invalid.

(j) ***Sequence Check:*** If the numbers in the source document are serial and the documents are in order, the input records will also be in numerical sequence. A check can be made by the program to determine whether the records are in either ascending or descending order.

*Task* Illustrate how the verification of codes is performed.

## Self Assessment

Fill in the blanks:

8. Once the analysis and design of the system has been done, it would be necessary to identify the data that are required to be processed to produce the............................ .

9. The quality of input design determines the ............................ of the system.

10. For a better user interface, it is most important to design ............................ screens.

11. The ............................ screen layouts provide the way by which user interacts with the computer.

12. If ............................ data enters the system, it is usually very costly to make the necessary corrections.

13. ............................ method is used for verifying data recorded in punched cards or on diskettes and magnetic tapes.

14. Sometimes numbers are added to produce a meaningless total called a ............................

15. A ............................ can be made by the program to determine whether the records are in either ascending or descending order.

## 12.3 Summary

- Output design considers the content, the occurrence, the format, medium and the allocation of output.
- Application Outputs are the outputs desired out of the system to meet its objectives.
- Operating Outputs are mainly generated for use of E.D.P Staff and give various indications as to how the system operates.

- It is not always possible and even not advised to print each and every output. Therefore, the system should also display all outputs on screen.
- Once the analysis and design of the system has been done, it would be necessary to identify the data that are required to be processed to produce the outputs.
- Input design features can ensure the reliability of the system and generate correct reports from the accurate data.
- The screen layouts provide the way by which user interacts with the computer.
- If incorrect data enters the system, it is usually very costly to make the necessary corrections.

## 12.4 Keywords

***Application Outputs:*** Outputs desired out of the system to meet its objectives.

***Hash Totals:*** Sometimes numbers are added to produce a meaningless total called a hash total.

***Input Data:*** Data given to the computer by the user through any of its input devices.

***Operating Outputs:*** There are mainly generated for use of E.D.P Staff and give various indications as to how the system operates.

***Output*:** The result of the procedure received by the computer after processing the input.

***Screen Layout:*** The screen layouts provide the way by which user interacts with the computer.

## 12.5 Review Questions

1. What are the different types of output? Illustrate.
2. Enumerate the common formats of a printed output. Discuss the basic guidelines for designing report formats.
3. What are the advantages of a printed output? Illustrate.
4. Depict the basic guidelines for designing a good output screen.
5. Why do we require output from computer?
6. Enlighten briefly the objectives of input design.
7. How does the analyst design the input form? Discuss.
8. List out the various methods commonly used for input verification and control.
9. What is an output Design? Make distinction between design of printed output and screen output.
10. Exemplify the goal of designing Input data.

### Answers: Self Assessment

| | | | |
|---|---|---|---|
| 1. | Output | 2. | information |
| 3. | Application | 4. | Operating |
| 5. | hard copies | 6. | graphic |
| 7. | tabular or report | 8. | outputs |
| 9. | reliability | 10. | user-friendly |

11. user-friendly
12. incorrect
13. Key verification
14. hash total
15. check

## 12.6 Further Readings

*Books*

Alan Dennis; Barbara Haley Wixom, *Systems Analysis Design,* New York: J. Wiley, ©2003.

Jerry FitzGerald; Ardra F FitzGerald, *Fundamentals of Systems Analysis : Using Structured Analysis and Design Techniques,* New York: Wiley, ©1987.

Pankoj Jalote, *An Integrated Approach to Software Engineering*, Springer.

*Online link*

http://www.peter-lo.com/Teaching/CS211/L07.pdf

Notes

# Unit 13: Online Dialogues Design & its Interface and File Design

CONTENTS

## Objectives

After studying this unit, you will be able to:

- Understand the design of online dialogues & its interface
- Discuss the design of files
- Understand the use of auxiliary storage devices

## Introduction

User interface design or user interface engineering is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction. In this unit, you will understand the design of online dialogues & its interface. Also you will study the design of files. Auxiliary storage points to storage excluding the main storage. We will discuss the uses of auxiliary devices in this unit.

## 13.1 Design of Online Dialogues & its Interface

There are different types of user interface designs, each of which has a distinctive character and aptitude. The design type is needed to be appropriate to the system's tasks and to its consumers who will interrelate directly with the computers. Important norm for the assessment of dialogue type:

- ***Easy to use:*** easy even with inexpert users
- ***Easy to learn*** easy for consumers to remember

- Processing and responding speed
- Easy to build up

## Important Types of User-computer Interface Design

Questions or pop-up reminders on computer are by turn replied by users. This kind of design is easy and appropriate with inexpert users:

***Menu table:*** Choices are classified and represented on the screen. Menu table is often used as a method for linking to the system. It is a fine strategy if the screen represents a full menu. This type of design is appropriate with inexpert users but uninteresting with more knowledgeable users.

***Symbols:*** Symbols are represented on the screen to symbolize for various functions. They are simple to learn and they facilitate fast access. Actually, graphics engage more space on the screen and they are not as profitable as menu table. The main limitation of symbols is their incapacity to describe the choices lively, clearly and meaningfully. To develop symbol, dialogue, professional softwares are essential.

***Form:*** Filling in the form is an accepted type of dialogue on data and data processing. Forms are represented on the screen likewise to the manner tables are organized. The screen also represents form name, field name and instruction information. The pointer handled by a software shifts automatically between fields or by means of TAR or carriage return – enter keys. The benefit of form is its close contact with users. This type of design is appropriate to all users.

***Language command:*** This is a broad but easy area comprising of both straightforward commands and grammatically difficult commands. Commands will consequence in a move of the system when it is entered by the customer. The most important benefit of language command is that its suppleness is restricted by the language's grammar only. Though, intakes time for users to study by heart the commands and customers are needed to have a background knowledge of the system in case there are no information represented on the screen. Language command queries for great attempts while producing it. It is appropriate for customers who are professionals.

## Essential Instructions in Dialogue Design

***Feedback information:*** Offers users with the information on what are being performed

***Status***: Keep users updated of the system's parts they are accessing

***Escape:*** Permit users to exit from one manipulation

***Minimum tasks:*** Avert customers from making too many manipulations

***Default:*** Set the often used parameter

***Support:*** Offer users with essential supporting information

***Cancel:*** Users can cancel and restart

***Consistence:*** The implementation of commands must be reliable through interface.

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design may be utilized to apply a theme or style to the interface without compromising its usability. The design process of an interface must balance the meaning of its visual elements that conform the mental model of operation, and the functionality from a technical engineering perspective, in order to create a system that is both usable and easy to adapt to the changing user needs.

Notes

User interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interaction yet also require some unique skills and knowledge. As a result, user interface designers tend to specialize in certain types of projects and have skills centered around their expertise, whether that be software design, user research, web design, or industrial design.

### 13.1.1 Processes

There are several phases and processes in the user interface design some of which are more demanded upon than others depending on the project. (Note for the remainder of this section the word system is used to denote any project whether it is a web site, application, or device)

- ***Functionality requirements gathering:*** Assembling a list of the functionality required of the system to accomplish the goals of the project and the potential needs of the users.
- ***User analysis:*** Analysis of the potential users of the system either through discussion with people who work with the users and/or the potential users themselves. Typical questions involve:
  - What would the user want the system to do?
  - How would the system fit in with the user's normal workflow or daily activities?
  - How technically savvy is the user and what similar systems does the user already use?
  - What interface look & feel styles appeal to the user?
- ***Information architecture:*** Development of the process and/or information flow of the system (i.e. for phone tree systems, this would be an option tree flowchart and for web sites this would be a site flow that shows the hierarchy of the pages).
- ***Prototyping:*** Development of wire frames, either in the form of paper prototypes or simple interactive screens. These prototypes are stripped of all look & feel elements and most content in order to concentrate on the interface.
- ***Usability testing:*** Testing of the prototypes on an actual user—often using a technique called talk aloud protocol where you ask the user to talk about their thoughts during the experience.
- ***Graphic-interface design:*** Actual look & feel design of the final Graphical User Interface (GUI). It may be based on the findings developed during the usability testing if usability is unpredictable, or based on communication objectives and styles that would appeal to the user. In rare cases, the graphics may drive the prototyping, depending on the importance of visual form versus function. If the interface requires multiple skins, there may be multiple interface designs for one control panel, functional feature or widget. This phase is often a collaborative effort between a graphic designer and a user interface designer, or handled by one who is proficient in both disciplines.

User interface design requires a good understanding of user needs.

### 13.1.2 Current Research

#### *Microsoft*

The MSN User Experience Team developed new user-centered methods to provide structured user input on the visual design of the newly-released MSN Explorer, an integrated software package. In the final product, users rated "appearance" above all of the product's features. This case describes how the MSN User Experience Team derived a design direction to set the most appropriate pace of visual change for millions of users with widely variant preferences. It discloses how these new methods maximized the product's visual appeal to the widest segment of the potential user base. The methods included design mark-up, a semantic design-description task, a statement rating task, a semantic desirability group card sort task, and a modified focus group discussion. This case documents the value of these new methods in predicting user reaction to visual design.

### *HUMANOID*

HUMANOID is a user interface design tool that lets designers express abstract conceptualizations of an interface in an executable form, allowing designers to experiment with scenarios and dialogues even before the application model is completely worked out. Three properties of the HUMANOID approach allow it to do so: a modularization of design issues into independent dimensions, support for multiple levels of specificity in mapping application models to user interface constructs, and mechanisms for constructing executable default user interface implementations from whatever level of specificity has been provided by the designer.

### *Software Tool*

A user interface software tool helps developers design and implements the user interface. Research on past tools has had enormous impact on today's developers—virtually all applications today are built using some form of user interface tool. In this article, we consider cases of both success and failure in past user interface tools. From these cases we extract a set of themes which can serve as lessons for future work. Using these themes, past tools can be characterized by what aspects of the user interface they addressed, their threshold and ceiling, what path of least resistance they offer, how predictable they are to use, and whether they addressed a target that became irrelevant. We believe the lessons of these past themes are particularly important now, because increasingly rapid technological changes are likely to significantly change user interfaces. We are at the dawn of an era where user interfaces are about to break out of the "desktop" box where they have been stuck for the past 15 years. The next millennium will open with an increasing diversity of user interface on an increasing diversity of computerized devices. These devices include hand-held Personal Digital Assistants (PDAs), cell phones, pages, computerized pens, computerized notepads, and various kinds of desk and wall size-computers, as well as devices in everyday objects (such as mounted on refrigerators, or even embedded in truck tires). The increased connectivity of computers, initially evidenced by the World Wide Web, but spreading also with technologies such as personal-area networks, will also have a profound effect on the user interface to computers. Another important force will be recognition-based user interfaces, especially speech, and camera-based vision systems. Other changes we see are an increasing need for 3D and end-user customization, programming, and scripting. All of these changes will require significant support from the underlying user interface software tools.

### *Peridot*

Peridot is an experimental tool that allows designers to create user interface components without conventional programming. The designer draws pictures of what the interface should look like and then uses the mouse and other input devices to demonstrate how the interface should operate. Peridot generalizes from these example pictures and actions to create parameterized procedures, such as those found in conventional user interface libraries such as the Macintosh Toolbox. Peridot uses visual programming, programming by example, constraints, and plausible inferencing to allow non-programmers to create menus, buttons, scroll bars, and many other interaction techniques easily and quickly.

*Did u know?* Peridot created its own interface and can create almost all of the interaction techniques in the Macintosh Toolbox.

*Notes* Peridot demonstrates that it is possible to provide sophisticated programming capabilities to non-programmers in an easy-to-use manner and still have sufficient power to generate interesting and useful programs.

### *Aesthetics*

An important aspect of the empirical study of user experience is the process by which users form aesthetic and other judgments of interactive products. The current study extends test users with a context (mode of use) in which to make their judgments, using sets of web pages from specific domains rather than unrelated pages, studying the congruence of perceptions of aesthetic value over time, including judgments after use of a web site, manipulating the aesthetic design of web pages and studying the relationship between usability and aesthetic value. Perceptions are not correlated with measures of task performance or mental effort. We conclude that context is a pivotal factor influencing the stability of users' perceptions, which must be explicitly addressed in the study of users' product experience.

*Notes* The type of aesthetics that is relevant to users' perceptions appears to depend on the application domain. The principle 'what is beautiful is usable' is not confirmed.

## Self Assessment

Fill in the blanks:

1. .................................. is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction.
2. Menu table is often used as a method for .................................. to the system.
3. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called .................................. design.
4. .................................. is a user interface design tool that lets designers express abstract conceptualizations of an interface in an executable form, allowing designers to experiment with scenarios and dialogues even before the application model is completely worked out.
5. A user interface .................................. tool helps developers design and implements the user interface.
6. .................................. is an experimental tool that allows designers to create user interface components without conventional programming.

## 13.2 Design of Files

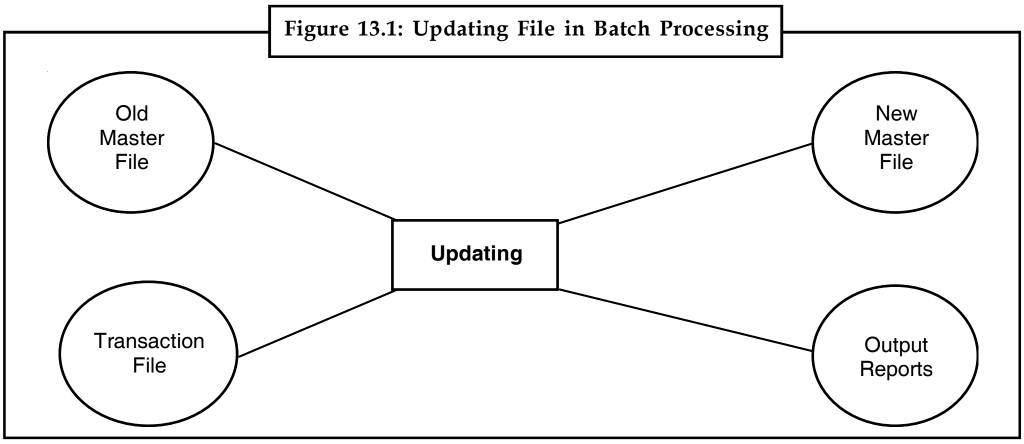
Notes

The basic factors to be considered in the selection of file media and file organisation method are:

- The method of processing for updating files
- Size of the file
- File inquiry capabilities
- Activity ratio of records in the file
- File volatility, and
- The response time.

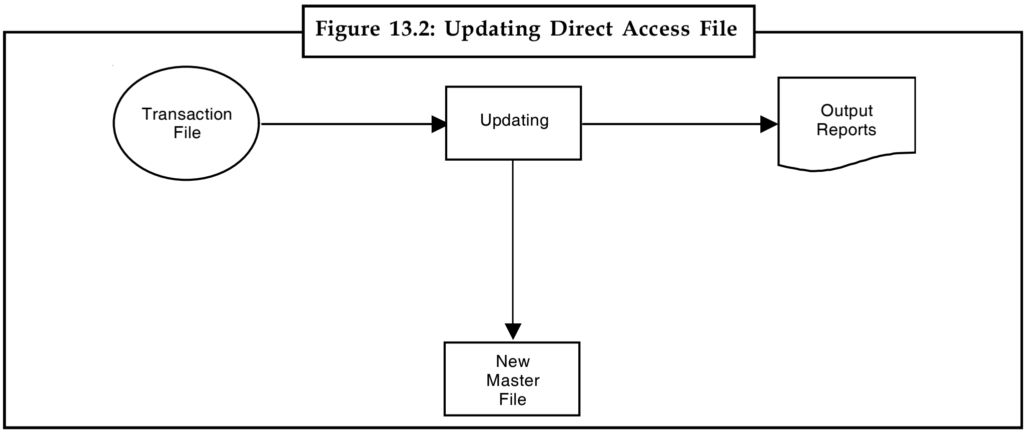
In batch processing, the sequential method of processing, using magnetic, is employed. This method of updating involves re-creation of the master file every time the file is updated. In order to reduce the setup time, the updating of the file is done after accumulation of a fairly large batch of transactions.

**Figure 13.1: Updating File in Batch Processing**



The method of random processing is used to update the files as the transactions occur. This is widely used method for on-line processing of files through demote terminals.

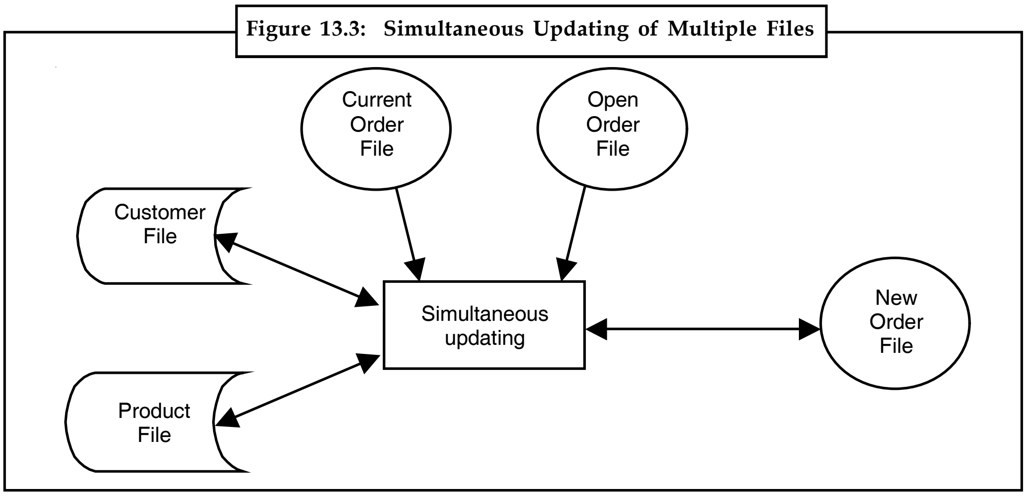
**Figure 13.2: Updating Direct Access File**



On-line processing allows for dispersing input/output terminals throughout the organization, so that various users can have access to the files. This can also be used in a batch processing mode where several files can be updated simultaneously, as illustrated in Figure 13.3. While the new open order file is created from a batched file or current orders, the customer file and the product file are accessed simultaneously and updated.

Notes

**Figure 13.3: Simultaneous Updating of Multiple Files**



Magnetic tape and magnetic disk can accommodate large files. Magnetic tape is more convenient and economical for holding large files, particularly when the records are processed sequentially in a batch mode. The magnetic disk can also holds large volumes of data, but it is relatively expensive. However, it is really suited for on-line processing. In designing the files, the growth potential should be taken into account.

The inquiry capabilities related to the ease of referring to a specific record in file without any alteration. Direct Access Storage Devices (DASD) can quickly handle file inquiries. A teletype or CRT is used to enter the inquiry specifying the record and the information desired. After the record in the file is accessed, the desired information is communicated by the CPU to the CRT or the teletypewriter. All these steps are completed within a few seconds after entering the inquiry. This inquiry capabilities were severely restricted before the advent of DASD.

*Caution* The inquiry capability of the DASD can be advantageously used without interrupting the normal processing operations.

The time interval between entering an inquiry and getting the reply is called response time. Some applications require a fast response as in the case of airline or hotel reservations, stock quotations in stock exchange, etc. The most suitable medium for such purposes is the DASD.

The activity ratio of a file is an important feature in file design. It is a measure of the proportion of records processed in an updating run. A file with high activity ratio can be processed more economically using the sequential processing method. If more than thirty per cent of the records in a file are used for updating, the activity ratio is considered high. Random or direct accessing is more suitable for updating files whose activity ratio is low, i.e., less than thirty per cent.

Another characteristic of a file is its volatility, which indicates the additions, deletions and changes to the file. When a file is accessed frequently, as in banks, stock exchanges, airlines, etc. in a working period, the file is regarded as highly volatile.

*Task* Depict the use of Direct Access Storage Devices (DASD).

## Self Assessment

Fill in the blanks:

7. In ................................., the sequential method of processing, using magnetic, is employed.
8. The method of ................................. processing is used to update the files as the transactions occur.
9. ................................. processing allows for dispersing input/output terminals throughout the organization, so that various users can have access to the files.
10. The time interval between entering an inquiry and getting the reply is called ................................. .
11. A file with high activity ratio can be processed more economically using the ................................. processing method.

## 13.3 Use of Auxiliary Storage Devices

Auxiliary storage points to storage excluding the main storage (such as, Magnetic tape or direct access devices). Other names provided to auxiliary storage are secondary and backing storage.

***Need of Auxiliary storage devices:*** When customer enters the data/program by means of input device, those are accumulated in RAM. RAM is a Volatile memory which means that all the contents of RAM are removed when the power is switched-off. Consequently, so as to amass the entered data enduringly for future reference, auxiliary memory is needed. Auxiliary storage devices such as Floppy disk, Magnetic disk and Magnetic tape contains auxiliary memory.

*Did u know?* Typically, data and information is read from the medium by means of auxiliary storage device and passed to the computer's main storage.

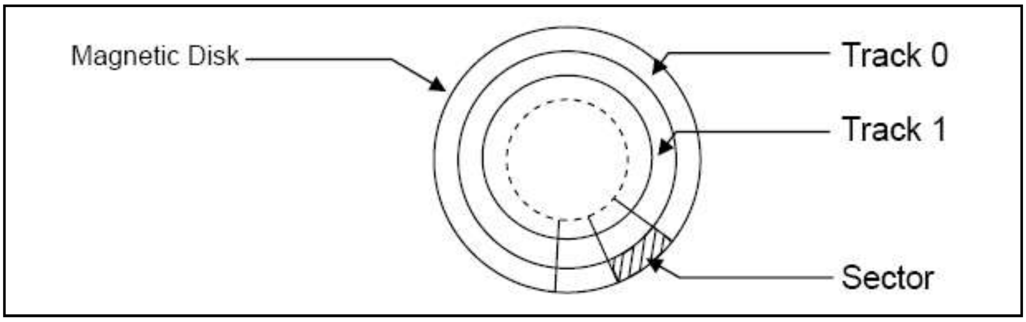
### 13.3.1 Magnetic Disk

Magnetic disks varies in size from the old 3" (not used these days) to the 3 ½". Hard disks are frequently located to be fixed inside computers and are faster and have superior storage capacities (10Mb – 100Mb). Hard disks are used to accumulate anything which is lastingly needed on line.

*Example:* Windows '98, MS Office, and customer documents

Exchangeable Hard Disks are erasable from the drive and restored by another. They are accessed for storage of files not needed to be always on-line (Backup).

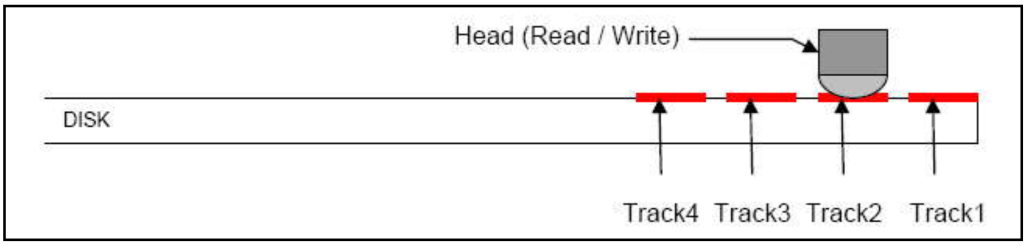
The amount of data that can be accumulated on a disk is based on how many tracks are there.



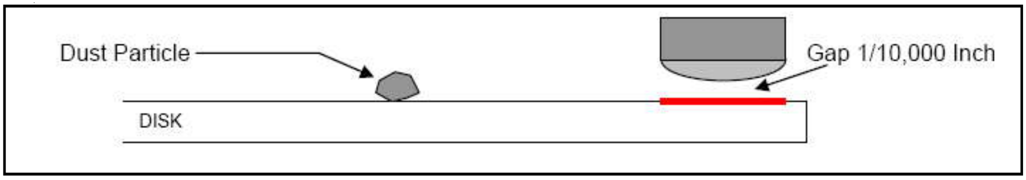
Notes

### *Head Positioning*

To read data from a disk it is necessary that the head is positioned accurately over the necessary track, else the output from the head may be too stumpy to be read. Clearly the closer the tracks are to each other the more decisive head positioning appears.



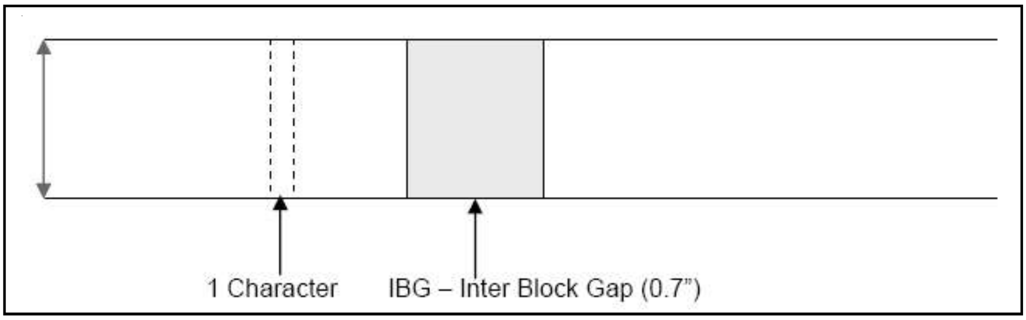
More complicated disk drives utilize a system called servo positioning to acquire exact head positioning. The head is shifted to its suitable position by a mechanical drive. An electronic system then gauges the output from the head and shifts the head to some extent until its output is maximum which appears when the head is precisely over the track.



*Task* Illustrate the process of head positioning in magnetic disk.

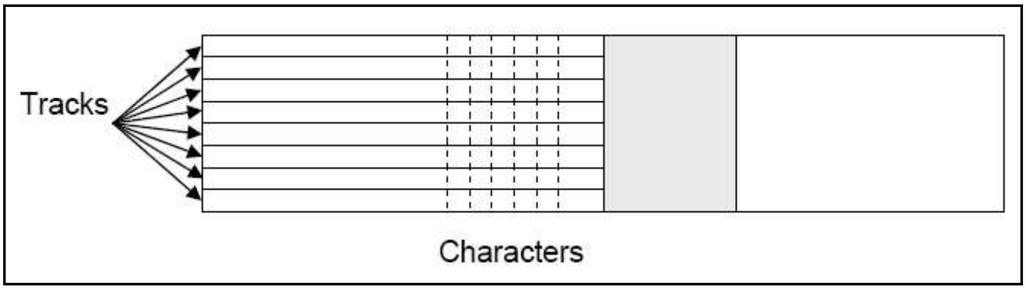
## 13.3.2 Magnetic Tape

With magnetic tapes, every character is recorded across the thickness of the tape.



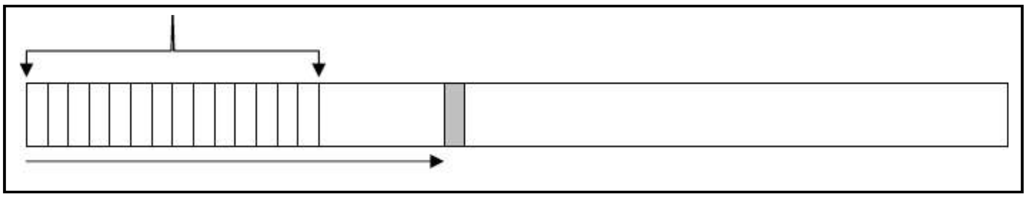
Recording density is calculated in bits/inch or characters for each inch. The IBG displays the end of every record or block. Thus only a fraction of the tape sustains data. The idea of the IBG is so that the tape can increase speed and slow down to the suitable data block and to indicate the start and end of each block.

A character comprises of 1's and 0's and is recorded on individual tracks running across the magnetic tape:



*Caution* Magnetic tapes can only be read in sequence, therefore all records must be read irrespective if those records are needed.

*Example:* To find a record stored half way on the magnetic tape will necessitate all records accumulated previously to that being read!



Read all records until getting necessary record.

## Self Assessment

Fill in the blanks:

12. ................................. is a Volatile memory which means that all the contents of RAM are removed when the power is switched-off.
13. ................................. are used to accumulate anything which is lastingly needed on line.
14. More complicated disk drives utilize a system called ................................. positioning to acquire exact head positioning.
15. With ................................., every character is recorded across the thickness of the tape.

## 13.4 Summary

- There are different types of user interface designs, each of which has a distinctive character and aptitude.
- User interface design or user interface engineering is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction.
- Filling in the form is an accepted type of dialogue on data and data processing

- The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design.
- There are several phases and processes in the user interface design some of which are more demanded upon than others depending on the project.
- HUMANOID is a user interface design tool that lets designers express abstract conceptualizations of an interface in an executable form, allowing designers to experiment with scenarios and dialogues even before the application model is completely worked out.
- A user interface software tool helps developers design and implements the user interface.
- Peridot is an experimental tool that allows designers to create user interface components without conventional programming.
- A file with high activity ratio can be processed more economically using the sequential processing method.
- Auxiliary storage points to storage excluding the main storage (such as. Magnetic tape or direct access devices).

## 13.5 Keywords

***Auxiliary Storage:*** Auxiliary storage points to storage excluding the main storage (such as, Magnetic tape or direct access devices).

***HUMANOID:*** It is a user interface design tool that lets designers express abstract conceptualizations of an interface in an executable form, allowing designers to experiment with scenarios and dialogues even before the application model is completely worked out.

***Peridot:*** It is an experimental tool that allows designers to create user interface components without conventional programming.

***User Interface Design:*** User interface design or user interface engineering is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction.

## 13.6 Review Questions

1. Illustrate some important norms for the assessment of dialogue type.
2. Enlighten some essential instructions used in dialogue design.
3. Explain the concept of Online Dialogues & its Interface design.
4. Describe several phases and processes used in the user interface design which are more demanded upon than others depending on the project.
5. Describe some tools used to create user interface components.
6. Describe various considerations of file design.
7. What is Auxiliary storage? Illustrate the need of Auxiliary storage devices.
8. Elucidate the concept of magnetic tape and magnetic disk.
9. Describe the concept of user interface software tool.
10. A file with high activity ratio can be processed more economically using the sequential processing method. Comment.

## Answers: Self Assessment

1. User interface design
2. linking
3. user-centered
4. HUMANOID
5. software
6. Peridot
7. batch processing
8. random
9. On-line
10. response time
11. sequential
12. RAM
13. Hard disks
14. servo
15. magnetic tapes

## 13.7 Further Readings

*Books*

Cäsar, M.; Alt, R.; Grau, J. (2002), Collaboration in the Consumer Product Goods Industry Analysis of Marketplaces, *Proceedings of 10th European Conference on Information Systems* (ECIS), Gdansk.

Devlin, B. (1997), *Data Warehouse - From Architecture to Implementation*, Addison-Wesley, Reading etc.

IBM Corp. (1984), *Business Systems Planning - Information Systems Planning Guide*, 4th ed.,IBM-Form GE20-0527-4, Atlanta.

Imhoff, C. (2000), *The Corporate Information Factory*, DM Review, December 1999, accessed 2000-03-29, <http://www.dmreview.com/editorial/dmreview>.

King, W.R. (1995), *Creating a Strategic Capabilities Architecture, Information Systems Management*, 12 (1), pp. 67-69.

Knecht, R. (2003), *Enterprise Architecture, Lecture Notes, Executive MBA in Business Engineering*, University of St. Gallen, accessed 2002-02-01, <http://seward.iwi.unisg.ch/bec/becportal.nsf>.

Malhotra, Y. (1996), *Enterprise Architecture: An Overview [WWW document]. @BRINT Research Institute*, accessed 2002-09-13, <http://www.brint.com/papers/enterarch.htm>.

McDavid, D. W. (1999), *A Standard for Business Architecture Description*, IBM Systems Journal, 38 (1), pp. 12-31.

Österle, H.; Brenner, W.; Hilbers, K. (1992), *Unternehmensführung und Informations System, Teubner*, Stuttgart.

Rowohl, F., Schelp, J. (2002), Enterprise Application Integration, Khosrowpour, M. (Ed.): *Issues & Trends of Information Technology Management in Contemporary Organizations*, Idea Group Publishing, Hershey etc., pp. 775-779.

Scheer, A.W. (2000), *ARIS - Business Process Modeling*, 3rd Ed., Springer, Berlin etc.

*Online link*

www.visualwebz.com

Notes

# Unit 14: Systems Engineering and Quality Assurance

CONTENTS

## Objectives

After studying this unit, you will be able to:

- Understand the concept of system engineering & quality assurance
- Discuss its design objectives
- Explain the Design of software & documentation

## Introduction

Systems engineering comprises of two important disciplines: the technical knowledge area in which the systems engineer functions, and systems engineering management. In this unit, we concentrate on the process of systems engineering management. Also you will understand the concept of quality assurance. Document production and desktop publishing tools support nearly every aspect of software development. These tools have been discussed in this unit.

## 14.1 System Engineering

Three usually used definitions of systems engineering are offered by the best recognized technical standards that relate to this subject. They all have a general idea:

- A logical series of activities and decisions that converts an operational requirement into a explanation of system performance parameters and a favored system configuration.
- An interdisciplinary strategy that includes the complete technical effort, and develops into and verifies an incorporated and life cycle balanced set of system people, products, and process solutions that convince customer requirements.

- An interdisciplinary, mutual strategy that derives, develops, and verifies a life-cycle balanced system solution which fulfils customer expectations and accomplishes public acceptability.

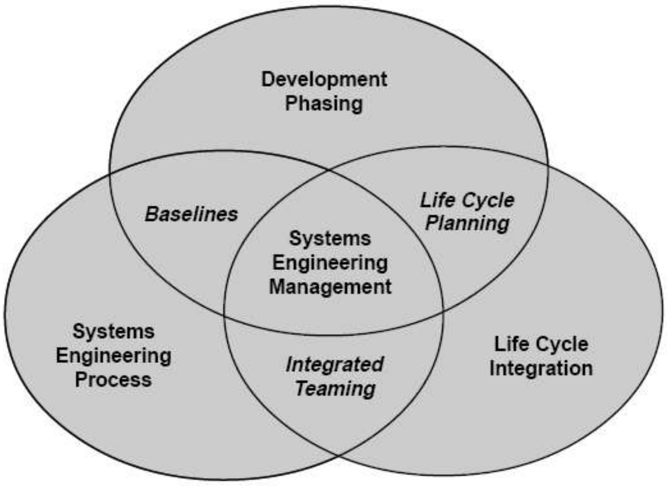
In a nutshell systems engineering is an interdisciplinary engineering management procedure that develops and authenticates an incorporated, life-cycle balanced set of system solutions that assure customer requirements.

### 14.1.1 Systems Engineering Management

As described by Figure 14.1, systems engineering management is accomplished by amalgamating three main activities:

- Development phasing that handles the design process and offers baselines that coordinate design efforts,
- A systems engineering process that offers a structure for solving design problems and tracking requirements flow via the design effort, and

**Figure 14.1: Three Activities of Systems Engineering Management**



- Life cycle integration that includes users in the design process and makes sure that the system produced is feasible all through its life.

All these activities is essential to attain proper management of an expansion effort. Phasing has two main objectives: it handles the design effort and is the main association among the technical management effort and the on the whole achievement effort. It handles the design effort by generating design baselines that administer each level of development. It interfaces with acquisition management by offering main events in the development process, where design feasibility can be assessed.

The feasibility of the baselines generated is a key input for acquisition management Milestone (MS) decisions. As an effect, the timing and coordination among technical development phasing and the acquisition schedule is significant to preserve a strong acquisition program.

The systems engineering process is the spirit of systems engineering management. Its idea is to offer a structured but supple process that converts needs into specifications, architectures, and configuration baselines. The discipline of this process offers the control and traceability to enlarge solutions that accomplish customer requirements. The systems engineering process may be repetitive one or more times throughout any phase of the development process.

*Notes* Life cycle incorporation is essential to make sure that the design solution is feasible throughout the life of the system. It involves the planning connected with product and process development, in addition to the integration of multiple functional issues into the design and engineering process. In this way, product cycle-times can be compacted, and the need for redesign and rework substantially abridged.

*Example:* Example of system engineering

The design of the saleable transport plane is an example of a systems engineering. In such a design the aerodynamic lift, the drag of fuselage and wings, the control equipment, the forward motion system, and such auxiliary hardware as the landing gear all interact considerably. One element cannot be disturbed without disturbing the others; all elements and facets of the total system, and the communications between them, must be measured. Therefore, if designers make the fuselage fatter and the wings smaller in an attempt to bear more payload at the similar or higher speeds, a new control system might be required due to the changes generated in the overall mechanical and aerodynamic traits of the vehicle.

### Self Assessment

Fill in the blanks:

1. Systems engineering comprises of two important disciplines: the .............................. knowledge area in which the systems engineer functions, and systems engineering management.

2. .............................. is an interdisciplinary engineering management procedure that develops and authenticates an incorporated, life-cycle balanced set of system solutions that assure customer requirements.

3. Phasing handles the .............................. effort by generating design baselines that administer each level of development.

## 14.2 Quality Assurance

The quality of an information system depends upon its design, development, testing and implementation. One aspect of systems quality is its reliability. A system is reliable if, when used in a reasonable manner, it does not produce failures that can be dangerous or costly. This definition distinguishes between software errors, instances of the system's not producing the expected result and failures as well as the occurrences of software errors. Although it is virtually impossible to develop software that can be proven to the error-free, yet software developers strive to prevent the occurrence of errors, using methods and techniques that include error detection and correction and error tolerance. Both these strategies are useful for keeping the system operating and preventing failure, of software. Failures are result of design errors that were introduced when specifications were formulation and software written.

Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. And, of course, it includes assurance that the system meets the specifications and the requirements for its intended usage and performance.

*Example:* A cooking recipe can be considered as an example of quality assurance. A recipe is a system for organizing a specific dish. It illustrates the ingredients and utensils required to prepare the food, the way of cooking it, how to test when it is prepared, how to accumulate it, and how to serve it. Cooking to a recipe generates better and more reliable outcomes. And the equivalent applies to using systems in other circumstances.

An additional aspect of quality assurance is avoiding the need for enhancement on the one hand and developing software that is maintainable on the other.

*Did u know?* The need for maintenance is very high and impedes new developments.

Quality assurance has been defined as a planned and systematic pattern of all the actions necessary to provide adequate confidence that software conforms to established technical requirements. The key ideas of quality assurance are as follows:

1. ***Comprehensiveness:*** Quality assurance is not restricted to the function of a software quality groups or phase. It includes all the necessary activities that contribute towards the quality of software through out the entire life-cycle of a project.
2. ***Planning:*** The emphasis is on a systematic plan to achieve the objectives of software quality. The quality of a piece of software is not left to the efforts of individuals.
3. ***Relativity:*** The notion of quality is relative to some requirements. The purpose of quality assurance is not to guarantee 100% reliability or zero detect software. It is rather to increase confidence that every reasonable effort has been made to ensure the quality of the end product. Quality, therefore, equals conformance to requirements, not excellence.
4. ***Cost:*** While purchasing any product, the level of quality of a product is usually reflected in its price. Hence, software quality involves increased cost. Product quality is, therefore, wholly a matter of customer choice. It is essential at the requirements analysis stage for the analyst to identify appropriate customer quality needs.

Preparation of a software quality assurance plan for each software project is a primary responsibility of the software quality group.

Topics is a software quality assurance plan include the following:

1. Purpose and scope of the plan.
2. Documents referred in the plan.
3. Organisational structure, tasks to be performed and specific responsibilities as they relate to product quality.
4. Documents to be prepared and checks to be made for the adequacy of the documentations.
5. Standards, practices and conventions to be used.
6. Reviews and audits to be conducted.
7. A Configuration management as well as plan that identifies software product items, controls and implements changes and records reports change status.
8. Practices and procedures to be followed in reporting, tracking and resolving software problems.
9. Specific tools and techniques to be used to support quality assurance activities.

10. Methods and facilities to be used to maintains and store controlled versions of identified software.
11. Methods and facilities to be used to protect computer program physical media.
12. Provisions for ensuring the quality of vendor - provided and subcontractor-developed software.
13. Methods and facilities to be used in collecting, maintaining and relating quality assurance records.

Other duties performed by quality assurance personnel include the following:

1. Development of standard policies, practices and procedures.
2. Development of testing tools and other quality assurance aids.
3. Performance of the quality assurance functions described in the software quality assurance plan for each project.
4. Performance and documentation of final product acceptance tests for each software product.

More specifically, a software quality assurance group may perform the following functions:

1. During analysis and design, a software verification plan and an acceptance test plan all prepared. The verification plan describes the methods to be used in verifying that the requirements are satisfied by the design documents and that the source code is consistent with the requirements, specification and design documentation. The source-code test plan is an important component of the software verification plan. The acceptance test plan includes test cases, expected outcomes and capabilities demonstrated by each test case. Often, quality assurance personnel would work with the customer to develop a single acceptance test plan.
2. During product evolution, in-process audits are conducted to verify consistency and completeness of the work products. Items to be audited for consistency include interface specifications for hardware, software and people internal design versus functional specifications source code versus design documentation and functional requirements versus test descriptions.
3. Prior to product delivery, a functional audit and a physical audit are performed. The functional audit reconfirms that all requirements have been met. The physical audit verifies that the source code and all associated documents are complete, internally consistent and consistent with one another and ready for delivery. A software verification summary is prepared to describe the results of all reviews, audits and tests conducted by quality assurance personnel throughout the development cycle.

Quality assurance personnel are sometimes in charge of arrangements for walk throughs, inspections and major milestone reviews. In addition, quality assurance personnel often conduct the project's postmortem, write the project legacy document and provide long-term retention of project records. The quality assurance organization can also serve as a focal point for collection, analysis and dissemination of quantitative data concerning cost distribution, schedule slippage, error rates and other factors that influence quality and productivity.

### 14.2.1 Quality Assurance and the Organization

The quality ethic has become influential in business circles ever since the Japanese demonstrated the marketability of the concept. In organizations, where primary business is the production of information systems, the question of quality is influenced by a number of issues as follows:

1. ***Project Management:*** It is argued that an effective project manager would build, into his monitoring process, clear checkpoints for assessing the quality of a developing information system.

2. ***Case:*** Many argue that case tools offer the developer a route to better quality systems because fewer errors are likely to occur between states such as design and implementation.

3. ***Formal Methods:*** The advocates of formal methods would claim that approach offer a key to quality be and implementation can proved to meet its specification.

4. ***Object-orientation:*** Object-oriented languages foster the reuse of coding of an information system from well proven parts and are likely to lead to increase quality.

### 14.2.2 Design Objectives

The objectives of the Quality Assurance System are:

- To preserve an effective Quality Assurance System complying with International Standard ISO9001 (Quality Systems).
- To attain and preserve a level of quality which improves the Company's reputation with customers.
- To ensure observance with relevant statutory and safety requirements.
- To attempt, always, to maximize customer satisfaction with the services provided.

The objective of systems engineering is to generate systems that convince the customers' needs, enlarges the probability of system success, decrease risk and diminish total-life-cycle cost.

*Task* Depict the objectives of the Quality Assurance System.

### Self Assessment

Fill in the blanks:

4. The .............................. of an information system depends upon its design, development, testing and implementation.

5. .............................. is the review of software products and related documentation for completeness, correctness, reliability and maintainability.

6. .............................. are result of design errors that were introduced when specifications were formulation and software written.

7. Preparation of a software quality assurance plan for each software project is a primary responsibility of the .............................. group.

8. The .............................. test plan includes test cases, expected outcomes and capabilities demonstrated by each test case.

9. The .............................. audit reconfirms that all requirements have been met.

10. The .............................. audit verifies that the source code and all associated documents are complete, internally consistent and consistent with one another and ready for delivery.

Notes

Caselet

## Software Quality Assurance Suffers due to Cynical Staffing

Why is it so very difficult to staff the software quality assurance with the right person? After researching on this question for many years, Mr Raghav S. Nandyal, author of *Making Sense of Software Quality Assurance* comes up with an answer: "The root cause of this problem, I find, lies much deeper in an organisational idiosyncrasy linked to ineffective staffing which can at best be described as a bizarre and a cynical approach to staffing software quality assurance — 'staff it with the non-performers'!"

Such thinking, he says, stems from a misplaced reasoning abetted by a wrong compensation system in which fire fighting is considered glamorous, whereas preventing fire is not.

Software quality assurance is a multifaceted function with a significant strategic importance, argues Mr Nandyal, in the course of a recent e-mail interaction with *Business Line*.

"Its primary responsibilities are to effect holistic and balanced process improvements, to retain institutional or organisational memory of execution, and to periodically use a structured diagnostic mechanism such as an appraisal or an audit to reinforce management commitment to the vision of excellence in execution."

*Excerpts from the interview:*

*What, according to you, are the much-wanted skills in software quality assurance?*

To be a successful software quality assurance professional, one needs to display a keen and sharp intellect in a range of organisational development and process management competencies.

I would also conclude that only those well-rounded individuals who have successfully served in all project capacities and positions — having graduated from a test engineer to a software developer, to a project leader, to a project manager, and having managed business engagements while making positive contributions to process improvement — become eligible to assume the role and function of a software quality assurance professional.

Of course, every position in an organisation competes for the best talent available. Yet, an organisation has a lot to lose from a weak software quality assurance function, given its multi-dimensional make-up.

Are there different certification courses available for wannabe software quality assurers? How expensive are these?

The most recognised certification for a software quality appraiser is defined by the Software Engineering Institute within its SCAMPI Appraiser authorisation program. The prerequisites and the process of gaining recognition as a SCAMPI appraiser are very involved and cumbersome.

The training cost alone up to completing the SCAMPI Lead Appraiser training adds up to a total of about $15,000. The three training programs that are required are the SEI Authorised Introduction to CMMI or People CMM, Intermediate Concepts of CMMI or People CMM, and the SCAMPI Lead Appraiser Training.

To gain the prerequisites and the necessary experience prior to undertaking these training programs, other costs such as participating as a team member on two SCAMPI appraisals, have to be factored-in. The process of gaining and retaining the recognition as a SCAMPI appraiser also places its own demands, cost-wise.

Since October 2006, the SEI has further drawn up a distinction among appraisers within its High Maturity Lead Appraiser (HMLA) certification program. Out of a total of 476 SCAMPI

Contd.....

appraisers today, about 26 per cent belong to the HMLA category. There are again expenses to gain this distinction and retain it.

So, as you can easily make out, the expenses to attain the SCAMPI appraiser certification or authorisation at the Software Engineering Institute are quite steep given a mix of competency and authorisation requirements. It would roughly cost an organisation upwards of $2,00,000 just to create a single SCMAPI lead appraiser from scratch.

What are CMMI and People CMM? How long are certifications valid?

Briefly, the CMMI is a well-documented model to engineer mature software processes. People CMM is very helpful to define a Competency Framework for an organisation and must be used hand-in-hand with the CMMI if one wants to build a robust organisation.

CMMI is structured to effect improvements in a localised context — to the software process, whereas the People CMM is more holistic in its approach.

People CMM is most definitely an organisational development framework. If CMMI implementation amounts to left-brain thinking, People CMM implementation is most definitely right-brain thinking! Organisations need both types of thought to reinvent themselves day after day.

Since organisational maturity cannot be built over a weekend retreat, both models approach the process of growing process and workforce maturity in measured steps, incrementally. There is therefore a characterisation of maturity into five degrees of freedom, Level 1 through Level 5 maturity. In each stage of process evolution, existing practices are made robust with the inclusion of more sophisticated practices.

SCAMPI appraisals are valid for a three-year period. Depending upon the class of SCAMPI appraisal, the manpower, cost and time required vary. Generally, it is true that SCAMPI Class C appraisals are less expensive when compared to Class B, and Class B is less expensive compared to a SCAMPI Class A.

A rough order estimate for manpower, cost and time depending upon the competencies of the lead appraiser for the three classes of appraisals, are as follows: SCAMPI C: 2 person-weeks, $10,000-15,000, five days; SCAMPI B: 3 person-weeks, $15,000-$20,000, five days; SCAMPI A: 6 person-weeks, $20,000-$25,000, seven days.

And what is the return?

Return on this investment is a highly subjective and emotive issue. Appraisals at best tell you what the gaps in your process infrastructure are. They may even make recommendations, in the final appraisal report, as to how you should prioritise and fix these gaps. How the organisation responds to this report is entirely up to the leadership in the organisation.

If the leadership ignores the report entirely and dumps it as garbage, then the investment goes down the proverbial drain! If the leadership only addresses the symptomatic causes and not the deeper dynamic issues, then the results if anything are bound to be superficial and the benefits are likely to be unpredictable in both the short-term and long-term.

If the leadership takes these appraisal recommendations seriously and embarks on a follow-up process improvement program with renewed vigour, then I would believe that the gains from making this investment would show up manifold — both quantitatively and qualitatively.

Since effective process evaluations and appraisals are a necessary prerequisite for improving both the process and product quality, I would say that well-intended appraisals show dramatic returns on investment on dimensions such as employee satisfaction, customer satisfaction, project profitability, reduced cost of rework and in general paving the way for the huge profit margins posted and reported by high process maturity work cultures.

*Contd.....*

Notes

A 20-30 per cent profit margin after tax is impossible if there is no discipline in execution. What I am certain about is that, when process appraisals are done well in order to serve the business interests, and are able to highlight the problems of efficiency and leverage confronting the organisation, a deep sense of discipline is ingrained into the follow-up process action plan. There is no question about it.

*Source:* http://www.thehindubusinessline.in/mentor/2008/03/31/stories/2008033150471200.htm

## 14.3 Design of Software & Documentation

After successful testing of the system, all the work done during systems analysis and design is required to be properly organized. The organized way of keeping records of all the documents, programs and diagrams prepared during all the phases of System Development Life Cycle (SDLC) is called documentation.

*Caution* All types of written instructions, which are prepared during SDLC and are required for operating and maintaining the system, must be included in the documentation.

Documentation can be prepared manually or by using computers. Manual methods are always time-consuming and require more manpower. Computers assist analysts in preparing documentation by many ways. During each phase of SDLC, the analysts prepare documents and draw diagrams by using word processor graphics software and CASE tools.

All these documents, files and graphics are later on merged into single file before implementation phase for making each project report. The project reports are designed by using a DTP software and laser printouts are taken. The project reports are then printed, binded and presented in a book form.

### 14.3.1 Documentation and Design Tools

Document production and desktop publishing tools support nearly every aspect of software development. Most software development organizations spend a substantial amount of time in developing documents, and in many cases the documentation process itself is quite inefficient. It is not unusual for a software development organization to spend as much as 20 or 30 per cent of all software development effort on documentation. For this reason, documentation tools provide on important opportunity to improve productivity.

Design tools enable a software developer to create models of the system to be built. The models contain a representation of data, function, and behaviour, and characterizations of data, architectural, procedural, and interface design. Well designed, modular software is more likely to meet the maintenance, reliability and testing requirements. Three specific tools are described below:

- Structured flowchart
- HIPO diagrams
- Warnier/Orr diagrams

*Notes* By performing consistency and validity checking on the model design tolls provide a software developer with some degree of insight into the analysis representation and help to eliminate errors before they propagate into the design, or worse, into implementation itself.

### *Structured Flowchart*

Structured flowcharts, also called Nassi-Schneiderman charts, are graphic tools that force the designer to structure software in modular as well as top-down form. They provide a proper structure that can be retained by a programmer for developing the application software.

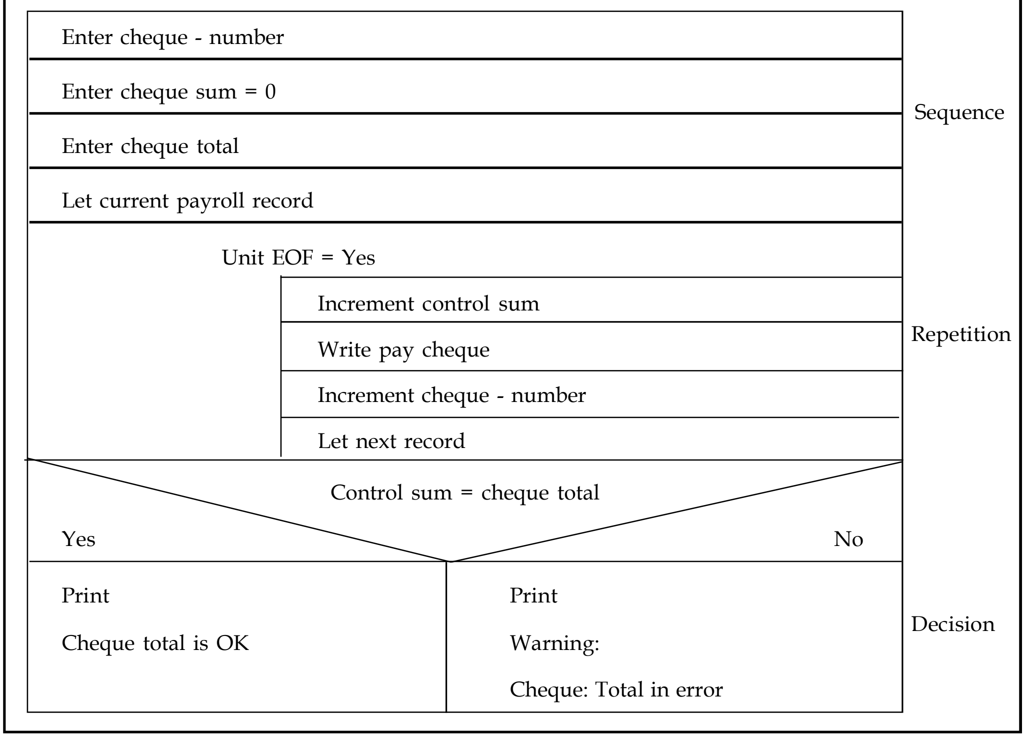
*Caution* The programmer should be expert in using the structured flowcharts.

The basic elements used in developing structured flowcharts are:

- ***Process:*** Simple processes or steps in a program are shown by a rectangular box, the process symbol. This symbol represents initialization of values, input and output operations and calls to execute other procedures.
- ***Decision:*** The decision symbol represents alternative conditions that can occur and that the program must have a manner of handling. The decision symbol may show actions for more than two alternatives at the same time.
- ***Iteration:*** The iteration symbol represents looping and repetition of operations while a certain condition exists or until a condition exists.

The structured flowcharts use no arrows or continuations on separate pages. Each structured flowchart is shown on a single sheet of paper. When designing a structured flowchart, the systems analyst specifies the logic in a top down fashion. The first consideration in a process is the top element. The second in sequence is next one shown and so forth. Similarly, there is a single exit from the process. Figure 14.2 illustrates on N-S chart designed for processing payroll cheques.

**Figure 14.2: Nassi-Schneidermann Chart**



Notes

Nassi-Schneidermann charts are winning wider acceptance in the computer industry because they so simply illustrate complex logic. Perhaps as analysts evaluate the various tools at this disposal, these charts will gain even more followers. As with other structured tools, they are single entry and single exist, and efficiently accommodate modules. However, they are not as useful for conveying system flow as they are for detailing logic development and they are difficult to maintain.

### *Hipo Diagram*

It is another tool commonly used for developing systems software. It is an acronym for hierarchical input process output. This method was originally developed to provide documentational assistance for programmers/analysts.

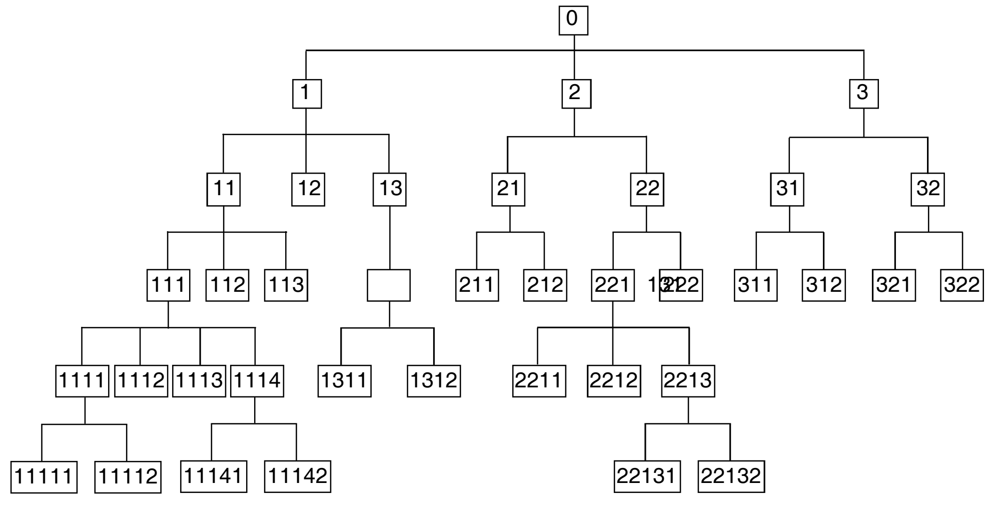
The major concept upon which HIPO is based is the highly structured modular design. The structure allows the enforcement of major principles to HIPO, a top-to-bottom approach to design. The emphasis is made on forcing the flow of data down through the system, not in the opposite direction.

The main idea behind the top-to-bottom approach is the elimination of "output-oriented" systems situations. An output-oriented system is concerned with providing output and does not bother about the sound principles of system design. In essence, an output-oriented system often gets the job done without delay.

Unfortunately, many data processing organization try to employ this type of rationale in their system designs. Output-oriented systems are often fragmentary, with large gaps evident in the logic and flow of data throughout the system. Programs written for this type of system often duplicate each other in part. The net effect is that more programming efforts are required, with a resultant, loss of manpower and time.

The HIPO concept, with its highly order structure and top-to-bottom approach, tries to eliminate piecemeal system design. A view of general HIPO structure is shown in Figure 14.3 .

**Figure 14.3: Structure of HIPO Chart Identifying all the Components within the Chart by Number at each of the Sub-level**



As you can observe that HIPO structure is quite similar to that of a manager's organizational chart. The numbers shown in various boxes of HIPO chart provide a mean of identifying each of this sub-levels and component blocks on the chart. The rationale of subdividing and identifying the component blocks within a HIPO design is extremely important. Applying this concept, the analyst is capable of defining and completely laying out the overall structure of the entire system under study.

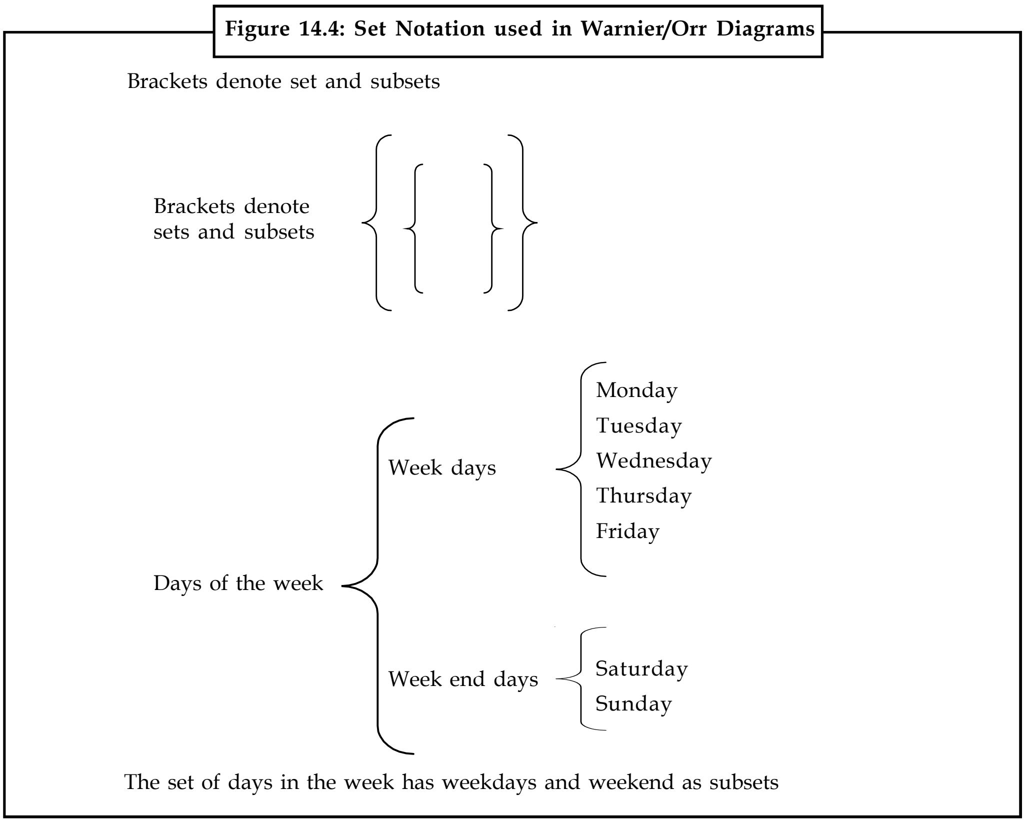
*Did u know?* The HIPO approach is mainly designed to accommodate the development of a system.

### *Warnier/Orr Diagram*

Warnier/Orr diagrams, also known as logical construction of programs/logical construction of systems are powerful tools aimed at designing of program structures by identifying the output and processing results and then working back-words to determine the steps and combinations of input to produce them. The simple graphic methods used in Warnier/Orr diagrams make the levels in the system evident and movement of the data between them vivid.

Warnier/Orr diagrams clearly show the various processes and sequences in which they are performed. Each process is defined in a hierarchical way. At each level, the process is shown in a bracket that groups its components (Figure 14.4). Since a process consists of different sub processes, a warnier/Orr diagram employs a set of brackets to indicate each level of system clearly.

**Figure 14.4: Set Notation used in Warnier/Orr Diagrams**



The set of days in the week has weekdays and weekend as subsets

Warnier/Orr diagrams are very powerful design tools and offer some distinct advantages to systems experts. They are simple in appearance and easy to understand.

Notes

*Task* Make distinction between HIPO diagrams and Warnier/Orr diagrams.

## Self Assessment

Fill in the blanks:

11. The organized way of keeping records of all the documents, programs and diagrams prepared during all the phases of system development life cycle (SDLC) is called ............................... .
12. ............................... tools enable a software developer to create models of the system to be built.
13. Structured flowcharts, also called ............................... charts, are graphic tools that force the designer to structure software in modular as well as top-down form.
14. ............................... method was originally developed to provide documentational assistance for programmers/analysts.
15. ............................... diagrams, also known as logical construction of programs/logical construction of systems are powerful tools aimed at designing of program structures.

## 14.4 Summary

- Systems engineering is an interdisciplinary engineering management procedure that develops and authenticates an incorporated, life-cycle balanced set of system solutions that assure customer requirements.
- The idea of system engineering process is to offer a structured but supple process that converts needs into specifications, architectures, and configuration baselines.
- The quality of an information system depends upon its design, development, testing and implementation.
- Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability.
- The organized way of keeping records of all the documents, programs and diagrams prepared during all the phases of system development life cycle (SDLC) is called documentation.
- Design tools enable a software developer to create models of the system to be built.
- Structured flowcharts, also called Nassi-Schneiderman charts, are graphic tools that force the designer to structure software in modular as well as top-down form.
- Hipo Diagram method was originally developed to provide documentational assistance for programmers/analysts.
- Warnier/Orr diagrams, also known as logical construction of programs/logical construction of systems are powerful tools aimed at designing of program structures by identifying the output and processing results and then working back-words to determine the steps and combinations of input to produce them.

## 14.5 Keywords

***Documentation:*** The organized way of keeping records of all the documents, programs and diagrams prepared during all the phases of system development life cycle (SDLC) is called documentation.

***HIPO Diagram:*** Acronym for hierarchical input process outputs. This method provide documentational assistance for programmers/analysts.

***Quality Assurance:*** Quality Assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability.

***Structured Flowcharts:*** Structured Flowcharts are the graphic tools that force the designer to structure software in modular as well as top-down form.

***Systems Engineering:*** Systems engineering is an interdisciplinary engineering management procedure that develops and authenticates an incorporated, life-cycle balanced set of system solutions that assure customer requirements.

***Worrier/Orr Diagram:*** Powerful tools aimed at designing of program structures by identifying the output and processing results and then working backwards to determine the steps and combination of input to produce them.

## 14.6 Review Questions

1. Illustrate the activities accomplished in systems engineering management.
2. What is quality assurance? Depict the key ideas of quality assurance.
3. Explain various functions performed by software quality assurance group.
4. Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. Comment.
5. Describe various design tools used for documentation of the system.
6. How will you prepare documentation? Illustrate.
7. Explain the role of design tools in documentation.
8. What are the basic elements used in developing structured flowcharts? Explain.
9. What is Hipo Diagram? Illustrate the idea and structure of Hippo chart.
10. Describe the process of Warnier/Orr diagrams.

### Answers: Self Assessment

1. technical
2. Systems engineering
3. design
4. quality
5. Quality assurance
6. Failures
7. software quality
8. acceptance
9. functional
10. physical
11. documentation
12. Design
13. Nassi-Schneiderman
14. Hipo diagram
15. Warnier/Orr

Notes

## 14.7 Further Readings

*Books*

James C Wetherbe, St. Paul , *Systems Analysis and Design,* West Pub. Co., ©1988.

Jerry Zeyu Gao, H. S. Jacob Tsao, Ye. Wu, *Testing and Quality Assurance for Component-Based Software,* Artech House.

Pankoj Jalote , *An Integrated Approach to Software Engineering,* Springer.

*Online link*

http://www.takeda.com/pdf/usr/default/ar2009e_16_35472_1.pdf

**LOVELY PROFESSIONAL UNIVERSITY**
Jalandhar-Delhi G.T. Road (NH-1)
Phagwara, Punjab (India)-144411
For Enquiry: +91-1824-300360
Fax.: +91-1824-506111
Email: odl@lpu.co.in