BARINGO TECHNICAL COLLEGE

ICT DEPARTMENT

DIPLOMA IN INFORMATION COMMUNICATION TECHNOLOGY (ICT)

MODULE II

SYSTEM ANALYSIS AND DESIGN

STANDARD NOTES

TOPIC 1: INTRODUCTION TO SYSTEM ANALYSIS AND DESIGN

Meaning of system analysis and design

The process of examining a (business) situation with the intent of improving it through better procedures and methods.

System Analysis - Process of gathering and interpreting facts, diagnosing problems, and using the facts to improve the system.

Systems Design - Process of planning a new system to replace or complement the old. Analysis specifies *what* the system should do and design states *how* to achieve the objective.

Note: This examination should always be initiated by the people involved in the situation (or who will be involved in a new situation). It is the job of the analyst to suggest solutions, but not make business decisions. (A computer based solution is not necessarily the only one).

System

A system is a set of components that interact to accomplish some purpose. e.g. College system, Economic system, Language system, a Business and its parts - Marketing, Sales, Research, Shipping, Accounting, Government.

Information system

Information System (I.S.): Interrelated components working together to collect, process, store, and disseminate information to support decision making, coordination control analysis and visualization in an organization.

Information technology

It is the use of any computers, storage, networking and other physical devices, infrastructure and processes to create, process, store, secure and exchange all forms of electronic data.

Components of an information system

Components of an information system include:

a) People – These use the system to fulfil their informational needs. They include end users and operations personnel such as computer operators, systems analysts, programmers, information systems management and data administrators.
b) Computer Hardware – Refers to physical computer equipment and devices, which provide for five major functions.
    - Input or data entry
    - Output
    - Secondary storage for data and programs
    - Central processor (computation, control)
    - Communication
c) Computer Software – Refers to the instructions that direct the operation of the computer hardware. It is classified into system and application software.
d) Telecommunication System/Communication network

e) **Databases** – Contains all data utilized by application software. An individual set of stored data is referred to as a file. Physical storage media evidences the physical existence of stored data, that is: tapes, disk packs, cartridges, and diskettes.
f) **Procedures** – Formal operating procedures are components because they exist in physical forms as manuals or instruction booklets. Three major types of procedures are required.
    - User instructions – for application users to record data, to use a terminal for data entry or retrieval, or use the result.
    - Instructions for preparation of input by data preparation personnel.
    - Operating instructions for computer operations personnel.

Types of information system

Major types of systems include:

1. Transaction Processing Systems (TPS)
2. Management Information Systems (MIS)
3. Decision Support Systems (DSS)
4. Executive Support Systems (ESS)
5. Expert Systems

Transaction Processing System (TPS)

A transaction is any business related exchange, such as a sale to a client or a payment to a vendor. Transaction processing systems process and record transactions as well as update records. They automate the handling of data about business activities and transactions. They record daily routine transactions such as sales orders from customers, or bank deposits and withdrawals. Although they are the oldest type of business information system around and handle routine tasks, they are critical to business organization. For example, what would happen if a bank's system that records deposits and withdrawals and maintain accounts balances disappears?

TPS are vital for the organization, as they gather all the input necessary for other types of systems. Think of how one could generate a monthly sales report for middle management or critical marketing information to senior managers without TPS. TPS provide the basic input to the company's database. A failure in TPS often means disaster for the organization. Imagine what happens when an airline reservation system fails: all operations stops and no transaction can be carried out until the system is up and running again. Long queues form in front of ATMs and tellers when a bank's TPS crashes.

Transaction processing systems were created to maintain records and do simple calculations faster, more accurately and more cheaply than people could do the tasks.

Characteristics of TPS:

- TPS are large and complex in terms of the number of system interfaces with the various users and databases and usually developed by MIS experts.
- TPS's control collection of specific data in specific formats and in accordance with rules, policies, and goals of organisation- standard format
- They accumulate information from internal operations o the business.
- They are general in nature—applied across organisations.
- They are continuously evolving.

The goals of TPS is improve transaction handling by:

- Speeding it up
- Using fewer people
- Improving efficiency and accuracy
- Integrating with other organizational information systems
- Providing information that was not available previously

Examples—Airline reservation systems, Automated Teller Machines (ATMs,) order processing systems, registration systems, Payroll systems and point of sale systems.

Management Reporting System (MRS)

Management Reporting Systems (MRS) formerly called Management information systems (MIS) provide routine information to decision makers to make structured, recurring and routine decisions, such as restocking decisions or bonus awards. They focus on operational efficiency and provide summaries of data. A MRS takes the relatively raw data available through a TPS and converts it into meaningful aggregated form that managers need to conduct their responsibilities. They generate information for monitoring performance (e.g. productivity information) and maintaining coordination (e.g. between purchasing and accounts payable).

The main input to an MRS is data collected and stored by transaction processing systems. A MRS further processes transaction data to produce information useful for specific purposes. Generally, all MIS output have been pre-programmed by information systems personnel. Outputs include:

a) Scheduled Reports – These were originally the only reports provided by early management information systems. Scheduled reports are produced periodically, such as hourly, daily, weekly or monthly. An example might be a weekly sales report that a store manager gets each Monday showing total weekly sales for each department compared to sales this week last year or planned sales.
b) Demand Reports – These provide specific information upon request. For instance, if the store manager wanted to know how weekly sales were going

on Friday, and not wait until the scheduled report on Monday, she could request the same report using figures for the part of the week already elapsed.

c) Exception Reports – These are produced to describe unusual circumstances. For example, the store manager might receive a report for the week if any department's sales were more than 10% below planned sales.

Characteristics of MRS

- MIS professionals usually design MRS rather than end users- using life cycle oriented development methodologies.
- They are large and complex in terms of the number of system interfaces with the various users and databases.
- MRS are built for situations in which information requirements are reasonably well known and are expected to remain relatively stable. This limits the informational flexibility of MRS but ensures that a stable informational environment exists.
- They do not directly support the decision making process in a search for alternative solutions to problems. Information gained through MRS is used in the decision making process.
- They are oriented towards reporting on the past and the present, rather than projecting the future. Can be manipulated to do predictive reporting.
- MRS have limited analytical capabilities. They are not built around elaborate models, but rather rely on summarisation and extraction from the databases according to the given criteria.

Decision Support System (DSS)

Decision support systems provide problem-specific support for non-routine, dynamic and often complex decisions or problems. DSS users interact directly with the information systems, helping to model the problem interactively. DSS basically provide support for non-routine decisions or problems and an interactive environment in which decision makers can quickly manipulate data and models of business operations. A DSS might be used for example, to help a management team decide where to locate a new distribution facility. This is a non-routine, dynamic problem. Each time a new facility must be built, the competitive, environmental, or internal contexts are most likely different. New competitors or government regulations may need to be considered, or the facility may be needed due to a new product line or business venture.

When the structure of a problem or decision changes, or the information required to address it is different each time the decision is made, then the needed information

cannot be supplied by an MIS, but must be interactively modelled using a DSS. DSS provide support for analytical work in semi-structured or unstructured situations. They enable mangers to answer 'What if' questions by providing powerful modelling tools (with simulation and optimization capabilities) and to evaluate alternatives e.g. evaluating alternative marketing plans.

DSS have less structure and predictable use. They are user-friendly and highly interactive. Although they use data from the TPS and MIS, they also allow the inclusion of new data, often from external sources such as current share prices or prices of competitors.

DSS components include:

a) Database (usually extracted from MIS or TPS)
b) Model Base
c) User Dialogue/Dialogue Module

Executive information system (EIS) / Executive Support Systems (ESS)

EIS provide a generalized computing and communication environment to senior managers to support strategic decisions. They draw data from the MIS and allow communication with external sources of information. But unlike DSS, they are not designed to use analytical models for specific problem solving. EIS are designed to facilitate senior managers' access to information quickly and effectively.

ESS has menu-driven user-friendly interfaces, interactive graphics to help visualization of the situation and communication capabilities that link the senior executives to the external databases he requires.

Top executives need ESS because they are busy and want information quickly and in an easy to read form. They want to have direct access to information and want their computer set-up to directly communicate with others. They want structured forms for viewing and want summaries rather than details.

Expert System (ES)

- It is an advanced DSS that provides expert advice by asking users a sequence of questions dependent on prior answers that lead to a conclusion or recommendation. It is made of a knowledge base (database of decision rules and outcomes), inference engine (search algorithm), and a user interface.
- ES use artificial intelligence technology.
- It attempts to codify and manipulate knowledge rather than information
- ES may expand the capabilities of a DSS in support of the initial phase of the

decision making process. It can assist the second (design) phase of the decision making process by suggesting alternative scenarios for "what if" evaluation.

- It assists a human in the selection of an appropriate model for the decision problem. This is an avenue for an automatic model management; the user of such a system would need less knowledge about models.
- ES can simplify model-building in particular simulation models lends itself to this approach.
- ES can provide an explanation of the result obtained with a DSS. This would be a new and important DSS capability.
- ES can act as tutors. In addition ES capabilities may be employed during DSS development; their general potential in software engineering has been recognised.

Other Information Systems

These are special purpose information systems. They are more recent types of information systems that cannot be characterized as one of the types discussed above.

(i) Office Automation Systems (OAS)

Office automation systems support general office work for handling and managing documents and facilitating communication. Text and image processing systems evolved as from word processors to desktop publishing, enabling the creation of professional documents with graphics and special layout features. Spreadsheets, presentation packages like PowerPoint, personal database systems and note-taking systems (appointment book, notepad, card file) are part of OAS.

(ii) Artificial Intelligence Systems

Artificial intelligence is a broad field of research that focuses on developing computer systems that simulate human behaviour, that is, systems with human characteristics. These characteristics include, vision, reasoning, learning and natural language processing.

Examples: Expert systems, Neural Networks, Robotics.

(iii) Knowledge Based Systems/ Knowledge Work Systems (KWS)

Knowledge Work Systems support highly skilled knowledge workers in the creation and integration of new knowledge in the company. Computer Aided Design (CAD) systems used by product designers not only allow them to easily make modifications without having to redraw the entire object (just like word processors for documents), but also enable them to test the product without having to build physical prototypes.

Architects use CAD software to create, modify, evaluate and test their designs; such systems can generate photo-realistic pictures, simulating the lighting in rooms at different times of the day, perform calculations, for instance on the amount of paint required. Surgeons use sophisticated CAD systems to design operations. Financial institutions use knowledge work systems to support trading and portfolio management with powerful high-end PCs. These allow managers to get instantaneous analysed

results on huge amounts of financial data and provide access to external databases.

Workflow systems are rule-based programs - (IF 'this happens' THEN 'take this action')- that coordinate and monitor the performance of a set of interrelated tasks in a business process.

(iv) Geographic Information Systems (GIS)

Geographic information systems include digital mapping technology used to store and manipulate data relative to locations on the earth. An example is a marketing GIS database. A GIS is different from a Global Positioning System (GPS). The latter is a satellite-based system that allows accurate location determination.

Roles of information system stake holders

A stakeholder is any person who has an interest in an existing or proposed information system. Stakeholders can be technical or nontechnical workers. They may also include both internal and external workers.

Some of the stakeholders include:

a) Systems owners

b) Systems users

c) Systems analyst

d) Systems designers

e) Systems developer

TOPIC 2: SYSTEMS THEORY

Introduction

A system is a set of interacting components that work together to accomplish specific goals. For example, a business is organized to accomplish a set of specific functions. Any situations, which involve the handling or manipulation of materials or resources of any kind whether human, financial or informative, may be structured and represented in the form of a system.

Systems theory concepts

- Entropy – The tendency towards disorder (chaos) in a system. The more closed a system is, the greater the entropy.
- Feedback – This is a control mechanism in open systems. Feedback involves measuring the output of the system, comparing the output with a standard and using any difference to modify subsequent input to ensure that the outputs conforms to the required standard.
  Elements of control include:
  - Goal: This is the expected performance, plan or results.
  - Sensor: Measures actual performance.
  - Comparator: Compares expected results to actual results obtained.
  - Effector: Reports deviation and initiates the response which may lead to a redirection of activity, revision of the expectation or changing the plan.
- Feed-forward – It means to take steps to make some adjustments to the system in advance in order to face any expected deviations in future. Feedback monitors the past results whereas feed-forward deals with future outcomes.
- Functional Decomposition – This involves factoring a system to its constituent subsystems. The subsystems are also decomposed further into manageable sizes resulting in a hierarchy structure of a system. Decomposition is used to analyse the existing system, to design and finally implement a new system.
- Functional cohesion – Involves dividing into subsystems by grouping activities that logically go together.

- Coupling – Occurs when two systems are highly interrelated.
- Decoupling – This is a process in which the subsystems are given autonomy and independence. The subsystems operate independently thereby pursuing own objectives and enhancing flexibility.
- Synergy – The whole is greater than the sum of its parts. At this point the focus is on global system needs, not local issues. It means that more than one system working together produce more result than each would achieve independently.
- Optimization – It is possible to achieve a best solution.
- Sub-optimization – It is an occurrence that occurs when the objectives of one element or subsystem conflicts with the objectives of the whole system.
- Equifinality – Certain results may be achieved with different initial conditions and in different ways. In open systems the same final state can be reached from several starting points, one result can have different causes, or through different methods, there is more than one way to achieve the objective.
- Goal-seeking – systems attempt to stabilize at a certain point.
- Holism – the analysis of a system is considered from the point of view of the whole system and not on individual subsystems. Subsystems are studied in the context of the entire system.

Characteristics of a system

a) **Purpose** – Systems exist to fulfil some objective or satisfy a need. A system may accomplish more than one task. The purpose of a system is closely tied to its rationale.
b) **Rationale** – This is the justification for a system's existence.
c) Efficiency – This is how well a system utilizes its resources, that is, doing things right.
d) **Effectiveness** – How well a system fulfils its purpose, assuming that its purpose is the right one. Involves a system doing the right things.
e) **Inputs** – Entities that enter the system to produce output or furnish information.
f) **Outputs** – Entities that exit from the system either as interfaces or for end-user

activities. They may be used to evaluate system's efficiency and effectiveness.

g) **Transformation rules** – Specify how the input is processed to produce output.
h) **Throughput** – Measures the quantity of work a system accomplishes. Does not consider the quality of the output.
i) **Boundary** – Artificially delimits a system for study or discussion purposes. System designers can only control those system components within the boundary.
j) **Environment** – That which impacts the system but is outside the system's boundary. The system cannot control events in the environment.
k) **Interfaces** – Points where two systems meet and share inputs and outputs. Interfaces belong to the environment although they may be inside the system boundary.
l) **Feedback** – Recycles outputs as subsequent inputs, or measures outputs to assess effectiveness.

Components of a System

a) Inputs: These provide the system with what it needs to operate. It may include machines, manpower, raw materials, money or time.
b) Processes: Include policies, procedures, and operations that convert inputs into outputs.
c) Outputs: These are the results of processing and may include information in the right format, conveyed at the right time and place, to the right person.
d) Systems Boundary: A system boundary defines the system and distinguishes it from its environment.
e) Subsystems: A subsystem is a unit within a system that shares some or all of the characteristics of that system. Subsystems are smaller systems that make up a super-system / supra-system. All systems are part of larger systems
f) Environment: This is the world surrounding the system, which the system is a subsystem of.

Types of Systems

There are two types of systems i.e.

a) Man-made system

b) Automated

Classification of systems

Each system can be characterized along a wide range of various characteristics.

Physical systems Vs Abstract systems

A physical system consists of a set of elements, which are coordinated and operate as a whole entity to achieve a certain objective. This system may also be called a concrete system.

An abstract system is an orderly arrangement of conceptual items or components.

Simple systems Vs Complex systems

A simple system has few components, and the relationship or interaction between elements is uncomplicated and straightforward.

A complex system has many elements that are highly related and interconnected.

Open systems Vs Closed systems

An open system interacts with its environment. It is a system with a feedback mechanism that promotes the free exchange of information between the system and the external entities. Organizations are open systems.

A closed system has no interaction with the environment. This is a system that neither transmits information to the outside world nor receives any information from the outside world. It is mainly a scientific concept (e.g. physics experiments).

Open loop systems Vs closed loop systems

An open-loop system is a system, which does not act in a controlled manner, that is, there is no feedback loop, and so no measure of performance against standards.

A closed-loop system is a system that functions in a controlled manner, such a system accepts inputs, works upon them according to some predefined processing rules and produces outputs. Such a system is controlled via a feedback loop.

Stable/Static systems Vs Dynamic systems

A stable system undergoes very little change over time. A dynamic system undergoes rapid and constant change over time.

Adaptive systems Vs Non-adaptive systems

An adaptive system is able to change in response to changes in the environment. These systems can also be described as cybernetic or self-organizing systems.

A non-adaptive system is not able to change in response to changes in the environment.

Deterministic systems Vs Probabilistic systems

Deterministic systems operate in a predictable manner. For example, thermostats and computer programs. In probabilistic systems however, it is not possible to determine the next state of the system. These systems depend on probability distribution.

The system even if the current state is known. For example, a doctor's diagnostic system.

Permanent systems Vs Temporary systems

A permanent system exists for a relatively long period of time.

A temporary system exists for only a relatively short period of time.

System Properties

Organizational systems contain both hard and soft properties.

- ***Hard properties*** are those that can be assessed in some objective way e.g. the amount of PAYE tax with tax code, size of product- quantifiable
- ***Soft properties*** - constitute individual taste. They cannot be assessed by any objective standard or measuring process e.g. appearance of a product, suitability

of a person for job and any problem containing a *political* element.

TOPIC 3: SYSTEM DEVELOPMENT LIFE CYCLE

Meaning of SDLC

SDLC is a framework for describing the phases involved in developing and maintaining information systems. This is also known as traditional system development method or function driven method or process driven method. The method requires the analyst to follow a sequence of phases during the development and implementation of an information system. This involves people and is described as information system development project.

SDLC stages

The following are the system development cycle phases or stages:

1. Preliminary survey/study: This stage involves determination of whether there is need to change the existing system or business procedures. It may require management requests for a change of the existing system to give an organization a competitive advantage or to improve the staff morale.
2. Feasibility study: This is a more detailed study carried out by a feasibility study team. Its purpose is to define the problem and decide whether or not a new system to replace the existing one is viable or feasible.
3. Facts finding and recording: This involves collection of information about the existing system on which to base analysis in order to determine whether users current needs are being met.
4. Analysis: A system analysis involves evaluation of the current system using the gathered facts or information.
5. System design: decides *how* the system will operate, in terms of the hardware, software, and network infrastructure; the user interface, forms and reports; and the specific programs, data- bases, and files that will be needed. Although
6. System development: This involves programming, testing and documentation activities.

7. System implementation:This phase involves the following activities

    a) Hardware selection, acquisition and installation
    b) User training
    c) File conversion/creation
    d) Changeover

TOPIC 4: PROBLEM DEFINITION

Problem definition

It is also referred to as preliminary study. This stage involves determination of whether there is need to change the existing system or business procedures. It may require management requests for a change of the existing system to give an organization a competitive advantage or to improve the staff morale. The user department should be involved during the definition of the problem. The problem to be solved should be very specific, clear and precise. It should not be too broad to cause ambiguities in its solution.

Objectives of the preliminary study are:

- To understand organizational characteristics and its objectives
- To understand organizational structure
- To identify organizational mission and collect relevant data or document regarding organizational information.

To develop a brief and accurate problems statement usually known as system term of reference (TOR).

Indicators of problems

Methods of identifying the problem

Terms of reference (TOR)

It is a documentation prepared by steering committee to act as a reference document throughout system development stages. Its contents include:

- Project title

- Subject of study
- Purpose of study
- Personnel
- Departments
- Sections affected or involved during the system implementation
- Available resources and constraints, the analyst, the project leader should consider
- The projects estimated duration and schedule

The importance of terms of reference is:

i. Provides information about the proposed system

ii. It may act as a reference document throughout the system development process

iii. It acts as an authorization document to the project development team by the management

iv. It gives the scope and extent of the proposed system project, thus setting out systems limitation and capability

v. It sets out the objectives of the proposed system

Steering committee

It is formed by two or three people to oversee the system development project from its initiation to its completion. It comprises of the system analyst as the project leader and a representative of the user department. They should understand all the processing objectives and procedures within the affected department. A management representative and accountant or auditors may be incorporated to advise initially on financial aspects on the project.

The functions of the steering committee are:

a) To study the current processing procedures that may require to be improved.

b) To prepare problem statement in form of terms of reference.

c) To coordinate system development activities throughout the development life

cycle.

d) To interface the project development team with organizational management.

e) To resolve conflict that may arise during system development.

To direct, control and monitor the system development progress.

TOPIC 5: FEASIBILITY STUDY

This is a more detailed study carried out by a feasibility study team. Its purpose is to define the problem and decide whether or not a new system to replace the existing one is viable or feasible. During the study, the analyst should assess the magnitude of the problem and attempt to restrict or at least identify the scope of the project. The analyst must list precisely the problems of the current system and also indicate what would be required of the new system. He must identify alternative solutions to the problems and recommend the most cost effective solution.

Feasibility study activities include:

- Identification of main characteristics of the existing system
- Determination of the main output requirements
- Considerations of alternative ways of meeting similar requirements.
- Preparation of gross estimates of developments, implementation and operation costs for each probable alternative solution.
- Documentation of the study i.e. writing of feasibility study report.
- Preparation of gross estimates of possible direct and indirect benefits for each probable alternative.

The following are the areas of feasibility study:

a) Technical Feasibility

b) Social Feasibility

c) Economical Feasibility

d) Legal Feasibility

Technical Feasibility

Technical questions are those that deal with equipment and software e.g. determination of whether the new system can be developed using the current computer facilities within the company. Technical feasibility is thus aimed at evaluation of the following:

i. The hardware required for the new system

ii. The software required for the new system

iii. Determination of whether the current facilities are adequate or inadequate for

the new system after implementation.

iv. Evaluation of the current technology and how applicable it is to the new system

v. Determination of the need for telecommunication equipment in the new system to improve both data capture and preparation activities.

vi. The inputs, outputs, files and procedures that the proposed system should have as compared to the outputs, files and procedures for the current system.

vii. Determination of whether training is necessary for the employees before the new system is implemented and the relevant skills required.

viii. Suggesting a suitable method of developing the new system, methods of running it when it becomes operational and ways of implementing it.

Social Feasibility

This is also known as operational feasibility. It mostly deals with the effect of the system on the current society within the company. It is carried out on the following areas:

i. The reaction of individuals both inside and outside the company as a result of the new system.

ii. The effect of the system on the existing organizational structure.

iii. The effect of the system on the current working practices and management levels i.e. whether there would be any change required and if so the cost of the change socially.

iv. Redundancy or retrenchment, implication to the company as a result of the new system.

v. Implication of the system on existing staff development programmes.

The social feasibility is carried out along with technical feasibility such that the social implications of every alternative technical solution to the problem that emerges are evaluated. Areas to consider include:

- Group relationships
- Salary levels
- Job titles and job descriptions

- Social costs to be evaluated e.g. cost of user training, consultancy that may be engaged during development of the new system, job improvements and salary changes.

Legal Feasibility

The new systems legal implications should be evaluated e.g. if it requires that the computer should be insured or whether the stored data should be registered with the government registrar before use. The copyright implication for restriction should be assessed before the new system is implemented. Generally any legal aspects associated with the new system should be assessed, and adequate measures taken to protect the interest of the user company.

Economic Feasibility

Economic feasibility is aimed at determination of whether or not to continue with the project, depending on whether the project is economically viable. The systems benefits and estimated implementation cost should be determined before any further resources can be spent on the project.

A cost benefit analysis (CBA) is carried out to determine whether the new system is economically viable.

Cost Benefit Analysis

Benefit Analysis: Is obtained through comparison of the new system and the existing system. Benefits of the new system fall under two categories i.e. direct and indirect benefits as well as tangible and intangible benefits.

i. Direct (Tangible)–fall under two categories: measurable benefits and direct savings.

Measurable benefits are those that can be quantified in monetary terms e.g. increase in working capital as a result of purchasing of computer systems or reduction of delays in decision making which is obtained through improved procedures e.g. invoicing procedures and credit control procedures.

Direct savings are those costs, reduced or eliminated as a result of introduction of computerized system. They include reduction or elimination of clerical personnel and elimination of some specific costs e.g. stationery costs. Like measurable benefits direct savings can be quantified in monetary terms.

ii. Intangible benefits – they are benefits that cannot be quantified in monetary terms or those that are difficult or impossible to quantify in monetary terms.

They are clearly desirable but very difficult to evaluate in terms of money value e.g. improved customer satisfaction, better information, improved organizational image, increased staff morale, a competitive advantage to an organization etc.

Cost Analysis: Costs are expenses or expenditure which are incurred by a system. These may include equipment cost, development cost, operation cost and software cost. During cost analysis one should consider both the new and the existing system. The cost of retaining and operating the existing system should be compared to the cost of introducing and running the computerized information system.

These costs fall under the following categories:

a) The cost of running the existing system. This is calculated from the past records. The items to consider include:
    i. Man power cost – which is extracted from the budgets and payroll reports
    ii. Material cost – which includes consumables e.g. stationery, work in progress and current stock
    iii. Operational cost e.g. the equipment cost expressed in terms of unit rate. Others to consider include the duration the project takes to be completed and initial replacement cost.
    iv. Overhead costs – which are direct expenses incurred by the company on behalf of all departments e.g. rent, electricity, telephone bill etc. These can easily be extracted from departments or centres to which they are allocated.
    v. The intangible cost of existing system e.g. loss of sales or cost of sales as a result of inappropriate stock levels or loss of interest in bank as a result of improper credit control system.
b) The cost of operating the proposed system – this is likely to include all the areas covered above i.e. manpower, materials, overheads and the intangible costs. However there are additional costs associated with computerized systems e.g. service contracts for the computer system, insurance of the computer system, cost of data transmission, cost of consumables like printer cartridges, ribbons etc. All these costs should be evaluated or estimated as accurately as possible.
c) The cost of new system development – Includes the cost incurred for any consultancy services that may have been hired during development. Allowances given to the system development team members fall under this category. Overall effects of the system development and implementation should be determined and any cost associated established. These estimates are based on both time and

activities involved in the project. Staff training cost, recruitment costs and retrenchment costs should be considered under system development cost.

Cost benefit analysis should be conducted on each alternative solution to the problem. This enables the analyst to make recommendation on a suitable cost effective alternative solution to the problem.

Cost benefit analysis techniques

The techniques used in economical evaluation of a computer based information system are the same used in investment appraisal in other areas of commercial world. These techniques tend to produce contradictory results and none of them is universally accepted. These techniques are based on either marginal costing methods or life cycle costing method. Marginal costing methods deal with snapshots of systems performance at a given point in time. Life cycle costing methods deal with measuring system performance over its working life.

These techniques include:

- The ARR (Accounting Rate of Return)
- Pay Back Period
- Discounted Cash flow – Net Present Value (NPV)
- Internal rate of Return (IRR)

Some of the limitations of CBA are:

- Difficult to consider all factors that might contribute costs or benefits
- Difficult to quantify some costs and benefits.

Feasibility study report

After the feasibility study and the cost benefit appraisal, a report is prepared that gives recommendations on whether or not to commit any further resources on the project.

The contents of the feasibility study report include:

- Introduction – It gives general description of the existing system, the people contacted during the study and purpose of the report.

- Description of the alternative proposed systems in terms of the inputs, outputs, file processed, response time etc.
- Quantification to justify the cost of running the proposed system
- The recommendation by the analyst on the most cost effective alternative solution.
- The author of the report
- System analyst recommendations on the new system indicating whether to commit further resources.
- If the decision is to continue with the project, its development plan should be given.

The report is submitted to the management for approval. After approval a more detailed survey is conducted on the existing system mostly to establish its weaknesses and strengths. This is called fact-finding or fact gathering.

Fact finding/investigation

This involves collection of information about the existing system on which to base analysis in order to determine whether users current needs are being met. The following are some of the activities that are looked at:

a) Functional requirement – the requirements should be established

b) Determination of the proposed system requirements – this is necessary as it may suggest a change in the existing system requirement.

c) Establish any weaknesses or problems associated with the present system, working methods and procedures.

d) Determination of organizational growth rate – this will assist in determination of the growth of the volume of transactions to be processed.

e) Determination of the organization structure, objective and the cost associated with the present system.

Fact-finding comprises of the following activities:

i. Fact-gathering

ii. Fact-recording

iii. Fact-evaluation

Fact-finding techniques

a) Use of questionnaires

A questionnaire is a special document that allows the analyst to ask a number of standard questions set to be asked to a large number of people in order to gather information from them. It is used when:

- the system analyst is located at a considerably long distance from the respondent
- there is a large number of respondents such that interviewing them will be limited by time
- the questions to be asked are simple and straight forward and require direct answers
- limited information is required from a large number of people
- it is used as a means to verify facts found using other methods.

Advantages of using questionnaires are:

- They provide a cheap means of gathering information/data from a large number of people.
- They encourage individuals to provide response without fear, intimidation or victimization.
- The respondents can complete the questionnaire at their own convenience with minimal or limited interruption of their work.
- Questions are presented consistently to all without bias.

Disadvantages of using questionnaires are:

- Response is often too slow since the respondents complete and return the form at their own convenience.
- They don't provide an opportunity for respondents to obtain clarification of questions which may appear vague or ambiguous.
- Does not provide an opportunity for the analyst to observe respondents'

reactions.

- The design of the questionnaire requires an expert who may charge expensively and may not be economical when used for a small group of users.
- All forms may not be returned and also not all questions may be answered which leads to incomplete data for analysis.

Requirements for preparing a questionnaire include:

- Questions should be simple and clear.
- The questions should be objectively oriented and one should avoid leading questions.
- The questions should be logically organized.
- The form should be neat.

b) Interviewing

This is a direct face-to-face conversation between the system analyst (the interviewer) and users (interviewees). He obtains answers to questions he asks the interviewee. He gets the interviewee's suggestions and recommendations that may assist during the design of the proposed system.

Interviews serve the following purposes:

- Acts as a method of fact-finding to gather facts about the existing system.
- Used for verifying facts gathered through other methods.
- Used for clarifying facts gathered through other methods.
- Used to get the user involved in the development of the new system.

Interviews are used in the following circumstances:

- When the respondents are few e.g. corporate managers
- When the respondents are physically available and accessible
- When the main emphasis of the system investigation is people
- When the analyst wishes to seek direct answers, opinions, suggestions and detailed information

- When the analyst wishes to verify validity of facts collected through other techniques
- When immediate response is required

Interviews have the following advantages:

- The analyst can frame questions differently to individuals depending on their level of understanding. Thus it allows detailed facts to be gathered.
- The analyst can observe non-verbal communication from the respondents or interviewees
- The response rate tends to be high
- Provides immediate response
- The analyst can get detailed facts from each respondent

Disadvantages of interviews are:

- Costly and time consuming when used on a large number of people
- Success highly depends on the analyst human relation skills, expertise and experience
- May not be practical due to location of respondent
- May make the respondents to feel that they are being summoned or grilled by analyst
- Interviews can fail due to:
    - Ambiguous questions being asked
    - Personal questions being asked
    - Inadequate time allocation for the exercise
    - Lack of earlier preparation by both parties
    - When the analyst is biased on using technical jargon

c) Observation

Observation is the most effective fact-finding technique but requires the analyst to participate in performing some activities carried out by the user. He may choose to

watch them as they perform their activities and gather the facts intended.

This method is best used in the following circumstances:

- When the validity of facts gathered through other methods is questionable
- When complexity of certain aspects of a system prevent a clear explanation by the respondents or the user
- Used to confirm that the procedures specified in the manuals are being followed.
- When one needs to obtain first hand and reliable information

Guidelines when using the observation method include:

- There should be permission from concerned authorities before the exercise
- Gathered facts should be recorded
- Those to be observed should be notified and the purpose of the exercise explained
- The analyst should be objective and avoid personal opinion. He should have an open mind
- The analyst should also record ordinary events

Advantages of observation method include:

- Data gathered is highly reliable thus the method can be used to verify facts collected through other methods
- The analyst can see what is being done clearly including the tasks which are difficult to explain clearly in writing or in words
- Inaccuracy or inaccurately described tasks can easily be identified
- It allows the analyst to easily compare gathered facts through other methods and what actually happened on the ground
- Relatively cheap compared to other methods

Disadvantages of observation are:

- People feel uncomfortable when being observed and behave abnormally thus

influence the analyst's conclusions

- The exercise may take place at odd times thus inconveniencing those involved
- The analyst may observe exceptional activities, leaving some critical areas. His patience and expertise play a great role
- The tasks being observed may be interrupted and the analyst may gather wrong facts

d) Record inspection/Document review

This method involves perusing through literature or documents to gain a better understanding about the existing system. Examples of documents that are perused include sales orders, job descriptions, existing systems documentation, management reports, procedure manuals, organized structure charts, trade journals etc.

This method is best used when:

- The analyst needs to have a quick overview of the existing system
- The information required cannot be obtained through any other techniques

Advantages of this method are:

- It is comparatively cheap compared to other techniques
- It is a faster method of fact finding especially when documents to be considered are few

Disadvantages of this method are:

- Time consuming if the documents are many or if they are not within the same locality
- Unavailability of relevant documents makes this method unreliable
- Its success depends on the expertise of the analyst
- Most of the documents or information obtained may be outdated

e) Sampling

Sampling is the systematic selection of representative elements of a population. The selected elements are examined closely and the results assumed to reveal useful information about the entire population.

This method is used when the target population:

- Is too large and it is impractical to study every element of the population
- Contains homogenous elements (elements with similar characteristics)

Advantages of sampling are:

- It reduces the cost e.g. by avoiding to examine every document or talking to everyone in the organization to gather facts
- It speeds up fact finding process
- It improves effectiveness since one can concentrate on few people and fewer documents and get adequate accurate information
- May reduce biasness, if a representative sample is taken. All the elements of the population stand a chance of being selected.

Disadvantages include:

- The sample may not be representative enough which may lead to incorrect and bias conclusions
- The expertise of the analyst is required since sampling involves a lot of mathematical computation

TOPIC 6: SYSTEM ANALYSIS

A system analysis involves evaluation of the current system using the gathered facts or information. One should evaluate whether the current and projected user needs are being met. If not, he should give a recommendation of what is to be done. Analysis involves detailed assessment of the components of the existing system and the requirements of the system.

The objectives or aims of system analysis are:

- To determine information needs of an organization and the users of that information
- Determination of the current activities of the system i.e. functions involved in conversion of inputs to outputs
- Determination of the intended systems output
- Determination of the resources required for the intended system

- Determine capabilities required in the system to meet information needs of the organization

System analysis activities are:

i) Analysis of the organization environment. The analyst should evaluate in details information needs of the organization environment e.g. information needs of the consumers, suppliers, competitors, government departments etc.

ii) Analysis of the present system. The analyst should study the current system and identify its weaknesses and its strengths. He should establish the ability of current system in meeting the stated information needs. This guides a decision to be made on whether the existing system stands to be improved, changed or done away with altogether. Some aspects of the existing system that are examined includes input transactions, outputs or results, existing controls, files, user interaction, methods, procedures, functions and existing hardware and software.

iii) Requirement analysis – involves determination of user requirements e.g. task performed, output expected, proposed system development cycle and user goals. The following are also determined:

- Maximum, minimum and average levels of activities
- Duplicate procedures e.g. two people entering the same transaction at different times
- Labour intensive tasks – the tasks that are manual and can easily be computerized
- Activities or tasks that involve complex or repetitive computation
- Procedures that have become obsolete

Once all the facts are analysed and documented a formal report is written called statement of requirements.

The contents of statement of requirements are:

i) Description of the initial system goals and whether or not they are met and are still applicable

ii) Description of whether the existing system is cost effective

iii) Description of whether the output produced is adequate, timely and well controlled

iv) Description of whether files held are suitable for supporting current organization requirements

v) Description of current system inputs and whether or not they support current file maintenance activities

vi) Description of the existing system workflow efficiency

vii) Description of any constraints within the system

viii) Description of any existing system equipment, procedures and controls that can be transferred to the new system.

The importance of system analysis are:

- It helps the analyst or system developer to gain understanding of the existing system.
- It allows the analyst or system developer to record existing system information in a standard form to aid design of a new system. It also facilitates understanding of the system by the user staff.
- Enables the analyst or developer to define existing system procedure into a logical model
- Helps the analyst to write or produce statement of requirements, which guides the development team throughout subsequent stages of the development life cycle.

Systems analysis methods/approaches

(i) Model driven approach

Emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems.

Developed system model becomes the blueprint for designing and constructing an improved system.

A Model – is a representation of either reality or vision. Since “a picture is worth a

thousand words," most models use pictures to represent the reality or vision

There are 3 Types

a. Structured analysis:

- Focuses on the flow of data through processes
- Key Model: Data flow diagram

b. Information engineering:-

- Focuses on structure of stored data
- Key Model: Entity Relationship Diagram
- Illustrate and synchronize the system's data and processes

c. Object-oriented analysis:-

- Integrated data and process concerns into objects
- Illustrate the system's objects from various perspectives such as structure and behavior.

(ii) Accelerated System Analysis

Approaches emphasize the construction of prototypes to more rapidly identify business and user requirements for a new system.

A Prototype – is a small-scale, incomplete, but working sample of a desired system.

Types of accelerated system analysis

a. Discovery Prototyping
   Identify the users' business requirements by building a small-scale, representative or working model of the users' requirements in order to discover or verify them
b. Rapid Architected Analysis
   Derive system models from existing systems or discovery prototypes. • Similar to reverse engineering

System Analysis Tools

System analysis tools include:

- DFD'S
- flowcharts

- data dictionary
- ELH

Flowcharts

A flowchart is a diagrammatic representation that illustrates the sequence of operations performed to get to the solution of a problem. Flowcharts facilitate communication between system analysts, system designers and programmers.

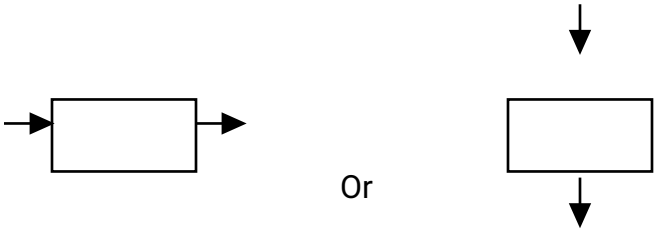
Guidelines for drawing a flowchart

Flowcharts are usually drawn using some standard symbols. Some standard symbols used in drawing flowcharts are:

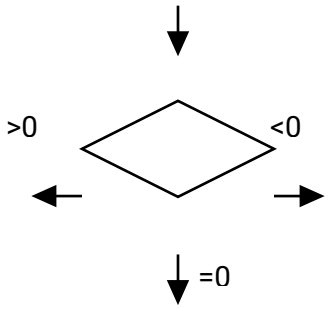
<Flowchart symbols>

The following are some guidelines in flowcharting:

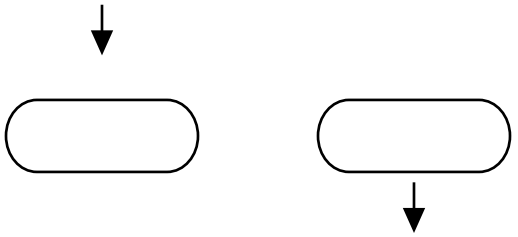
- In drawing a proper flowchart, all necessary requirements should be listed out in logical order
- The flowchart should be clear, neat and easy to follow. There should not be any room for ambiguity in understanding the flowchart
- The usual direction of the flow of a procedure or system is from left to right or top to bottom
- Only one flow line should come out from a process symbol



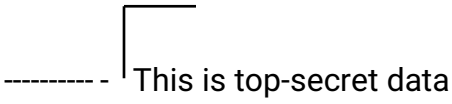
- Only one flow line should enter a decision symbol, but two or three flow lines, one for each possible answer, should leave the decision symbol



- Only one flow line is used in conjunction with the termination symbol



- Write within the standard symbols briefly. As necessary, you can use the annotation symbol to describe data or computational steps more clearly.



- If the flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines. Avoid the intersection of flow lines if you want to make it more effective and better way of communication.
- Ensure that the flowchart has a logical start and finish
- It is useful to test the validity of the flowchart by passing through it with simple test data.

Types of flowcharts

Flowcharts are of three types:

- System flowcharts
- Run flowcharts
- Program flowcharts

System Flowcharts

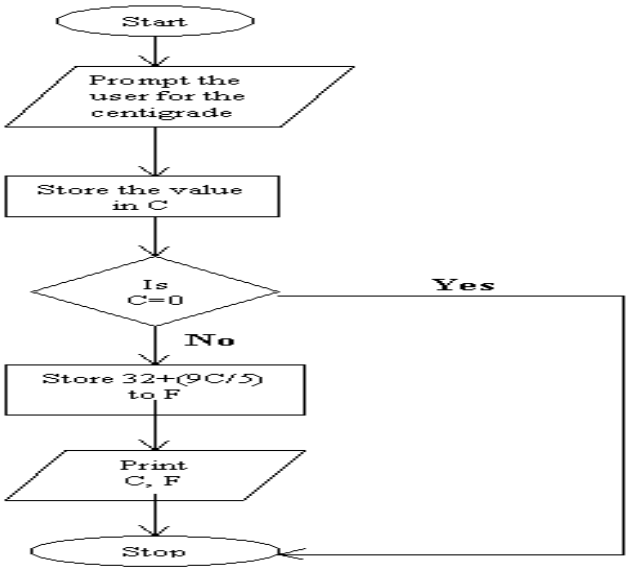
System flowchart describes the data flow for a data processing system. It provides a logical diagram of how the system operates. It represents the flow of documents, the operations performed in data processing system. It also reflects the relationship between inputs, processing and outputs. Following are the features of system flowcharts:

- The sources from which data is generated and device used for this purpose
- Various processing steps involved
- The intermediate and final output prepared and the devices used for their storage

The figure below is a sample of system flowchart for the following algorithm (step by step instructions on how to perform a certain task):

- Prompt the user for the centigrade temperature.
- Store the value in C

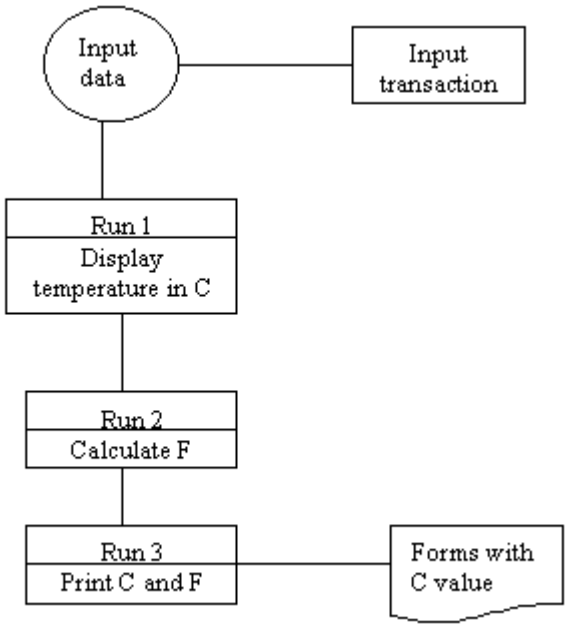
- Set F to 32+(9C/5)
- Print the value of C, F
- Stop



System Flowchart

Run Flowcharts

Run flowcharts are used to represent the logical relationship of computer routines along with inputs, master files transaction files and outputs.The figure below illustrates a run flowchart.

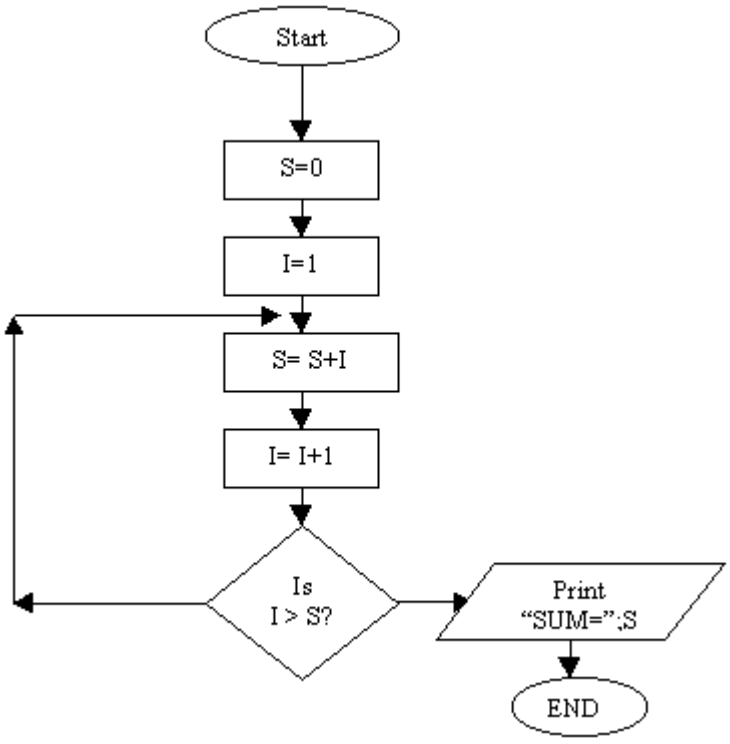


Run Flowchart

Program Flowcharts

A program flowchart represents, in detail, the various steps to be performed within the system for transforming the input into output. The various steps are logical/ arithmetic operations, algorithms, etc. It serves as the basis for discussions and communication between the system analysts and the programmers. Program flowcharts are quite helpful to programmers in organizing their programming efforts. These flowcharts constitute an important component of documentation for an application.

The figure represents a program flowchart for finding the sum of first five natural numbers (i.e. 1,2,3,4,5).



Program Flowchart

Advantages of using flowcharts

- Communication: Flowcharts are better ways of communicating the logic of a system to all concerned.
- Effective analysis: With the help of flowchart, problem can be analysed in more effective way.
- Proper documentation: Program flowcharts serve as a good program documentation, which is needed for various purposes
- Efficient coding: The flowcharts act as a guide or blueprint during the systems

analysis and program development phase.

- Proper debugging: The flowchart helps in the debugging process.
- Efficient program maintenance: The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part.

Limitations of using flowcharts

- Complex logic: Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
- Alterations and modifications: If alterations are required the flowchart may require redrawing completely.
- The essentials of what is done can easily be lost in the technical details of how it is done.

Data flow diagram

Data Flow Diagram is a graphical representation of the logical flow of the data.

- It helps in expressing the system's requirements in simple and understandable form
- It is also known as a bubble chart.
- Its aim is to clarify the system requirements and identify major transformations that will become programs in system design
- It decomposes the requirement specifications down to the lowest level of details

DFD contains four graphical symbols for:

a) Source/destination: its sign is a square. It represents source/destination of system data.
b) Data flow-its sign is an arrow. It identifies the data flow; it is a pipeline through which the data flows, i.e movement of data from one component to another.

c) Process: Its sign is a circle. It represents a process that transforms incoming data flow into outgoing flow.

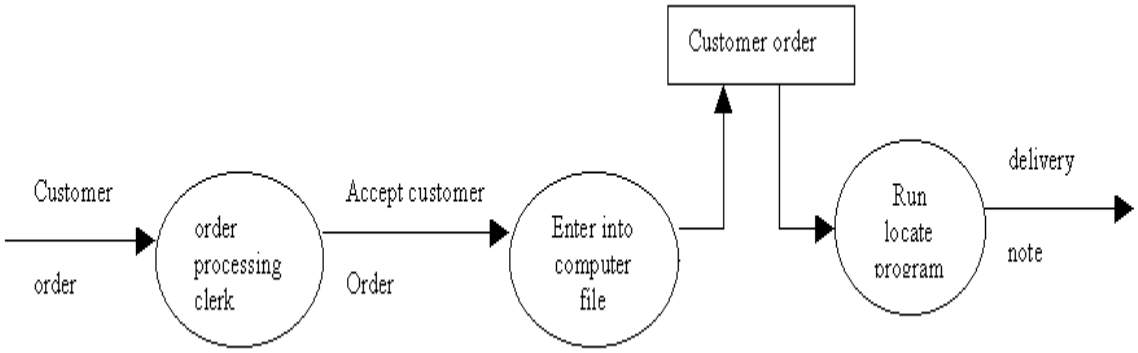
a) Data store: its sign is a rectangle. Represent stores of data within the system

Physical and Logical DFD

A **logical DFD** of any information system is one that models what occurs without showing how it occurs. An example is illustrated below.

It is clear from the figure that orders are placed, orders are received, the location of ordered parts is determined and delivery notes are dispatched along with the order. It does not however tell us how these things are done or who does them. Are they done by computers or manually and if manually who does them?

A **physical DFD** shows, how the various functions are performed? Who does them? An example is illustrated below:



The figure is opposite to that of the logical DFD, it shows the actual devices that perform the functions. Thus there is an "order processing clerk", an "entry into computer file" process and a "run locate program" process to locate the parts ordered. DFD(s) that shows how things happen, or the physical components are called physical DFD(s).

Typical processes that appear in physical DFDs are methods of data entry, specific data transfer or processing methods.

Difference between Flowcharts and DFD

The program flowchart describes boxes that describe computations, decisions, interactions and loops. It is important to keep in mind that data flow diagrams are not program flowcharts and should not include control elements. A good DFD should:

- Have no data flows that split up into a number of other data flows
- Have no crossing lines
- Not include flowchart loops of control elements
- Not include data flows that act as signals to activate processes.

Data dictionary

Data dictionary is an automated manual tool for storing and organizing information about the data maintained in a database. A data dictionary is a file which defines the basic organization of a database. It contains a list of all files in the database, the number of records in each file and the name and types of each field. All data elements contained in data dictionary are accompanied with a short description on what they are.

Its characteristics are:

(i) A query facility:- this is both for administrators and casual users. It helps users to perform searches on items like business definitions, user descriptions or even synonyms.

(ii) Automated input facilities:- this are to enable loading of records

(iii) Security features:- to help in protecting the information contained in the data dictionary

(iv) Comprehensive data reporting language for user designed reports.

(v) Language inter-phase, to allow, for example standard record layouts to be automatically incorporated into programs during the compile process.

(vi) Help facility – this helps to instruct users on how to use the data dictionary.

TOPIC 7: SYSTEM DESIGN

The objective system design is to put a logical structure of the real system in a form that can be interpreted by other people apart from the designer. The analyst should derive a logical model of the way the existing system works. There is an assumption that the existing system provides a good guide to what is required of a new system. It should be different from how the new system is to achieve the given requirement.

Limitations of system design include:

- There could be some requirements of the new systems that are not currently being satisfied by the current system. These requirements should not be taken into account.
- Inefficiency in the current system may be translated into a logical model and these will be transferred to the new system. Ideally, the models should reveal the logic of an efficient system and should be amended accordingly.
- It is likely that physical aspects of the existing system may be transferred to the logical analysis. The analysts should guard against that.

The above limitations can be dealt thus:

- Treatment of the new requirement: The new requirement can be estimated through interviews with management and users. It is important that the logical model be amended to reflect the new requirements. They are likely to lead to new processes that are added to higher-level design.
- Treatment of inefficiencies: The model should be adjusted through the decomposition of top level design tools e.g. DFDs. The lower level data flow diagram tend to be determined partly by what is done in existing system to fulfil a function.
- Treatment of physical aspects: Certain physical considerations may have shifted into a logical model e.g. a data store or file may contain extra information which may require amendment e.g. to incorporate separate files.

There are two types of design: logical design and physical design.

Logical Design

A logical design produces specifications of major features of the new system, which meets the system's objectives. The delivered product of the logical design is a blueprint

of the new system. It includes the requirements of existing system components:

- Outputs (reports, displays, timings, frequencies etc)
- Inputs (dialogs, forms, screens etc)
- Storage (requirement for data to be stored in databases)
- Procedures (to collect, transform and output data)
- Controls (requirements for data integrity, security, data recovery procedures)

Note: Logical design of the system is viewed in terms of what process i.e. procedures, inputs, outputs, storage and control the new system should have.

Physical Design

It takes the logical design blueprint and produces the program specification, physical files or database and user interface for the selected or targeted hardware and software. Physical design is mostly concerned with how the current system or future system works in terms of how the programs are written, files organized in storage media and how tasks are carried out for user interface with the system.

System design objectives

The designed system should meet the following criteria:

- User needs are met as cost effectively as possible
- One that is within the constraints laid down in the terms of reference
- Produce correct outputs by processing data accurately and efficiently
- Simple to operate i.e. easy to use
- One with sufficient built in controls and provide feedback to its user
- Should be reliable
- Should be functional

System design constraints

- The budget: A well-designed system incurs greater expenses. The total system cost of meeting the objectives must be considered in the light of the available budget.
- Time: Time taken to produce a very usable system would increase development cost and delay system delivery.
- Integration with other existing system: Existing and planned system may limit option and available features of the system.
- Skills: Limitation may arise from the range of skills and level of competence in both the design team and the system users.
- Standards: Standards may drive the design tasks in a specified direction.

System design tools

Decision tables

A Decision table represents conditions and the respective actions to be taken to address them, in a structured tabular format.

It is a powerful tool to debug and prevent errors. It helps group similar information into a single table and then by combining tables it delivers easy and convenient decision-making.

Creating Decision Table

To create the decision table, the developer must follow basic four steps:

- Identify all possible conditions to be addressed
- Determine actions for all identified conditions
- Create Maximum possible rules
- Define action for each rule

Decision Tables should be verified by end-users and can lately be simplified by eliminating duplicate rules and actions.

Example

Let us take a simple example of day-to-day problem with our Internet connectivity. We begin by identifying all problems that can arise while starting the internet and their

respective possible solutions.

We list all possible problems under column conditions and the prospective actions under column Actions.

	Conditions/Actions	Rules							
Conditions	Shows Connected	N	N	N	N	Y	Y	Y	Y
	Ping is Working	N	N	Y	Y	N	N	Y	Y
	Opens Website	Y	N	Y	N	Y	N	Y	N
Actions	Check network cable	X							
	Check internet router	X				X	X	X	
	Restart Web Browser							X	
	Contact Service provider		X	X	X	X	X	X	
	Do no action								

Table : Decision Table – In-house Internet Troubleshooting

Structured English

This is nothing but the description of what is required to code and how to code it. Structured English helps the programmer to write error-free code.

Other form of methods, which use graphs or diagrams, may are sometimes interpreted differently by different people. Here, both Structured English and Pseudo-Code tries to mitigate that understanding gap.

Structured English is the uses plain English words in structured programming paradigm. It is not the ultimate code but a kind of description what is required to code and how to code it. The following are some tokens of structured programming.

IF-THEN-ELSE,

DO-WHILE-UNTIL

Analyst uses the same variable and data name, which are stored in Data Dictionary, making it much simpler to write and understand the code.

Example

We take the same example of Customer Authentication in the online shopping environment. This procedure to authenticate customer can be written in Structured English as:

Enter Customer_Name

SEEK Customer_Name in Customer_Name_DB file

IF Customer_Name found THEN

  Call procedure USER_PASSWORD_AUTHENTICATE()

ELSE

  PRINT error message

  Call procedure NEW_CUSTOMER_REQUEST()

ENDIF

The code written in Structured English is more like day-to-day spoken English. It cannot be implemented directly as a code of software. Structured English is independent of programming language.

ERDS

An entity relationship diagram (ERD) is a picture which shows the information that is created, stored, and used by a business system. An analyst can read an ERD to discover the individual pieces of information in a system and how they are organizedand related to each other.

On an ERD, similar kinds of information are listed together and placed inside boxes called entities. Lines are drawn between entities to represent relationships among the data, and special symbols are added to the diagram to communicate high-level business rules that need to be supported by the system. The ERD implies no order, although entities that are related to each other are usually placed close together

Elements of an Entity Relationship Diagram

There are three basic elements in the data modeling language (entities, attributes, and relationships), each of which is represented by a different graphic symbol. There are

many different sets of symbols that can be used on an ERD. No one set of symbols dominates industry use, and none is necessarily better than another. We will use crow's foot in this book

Entity

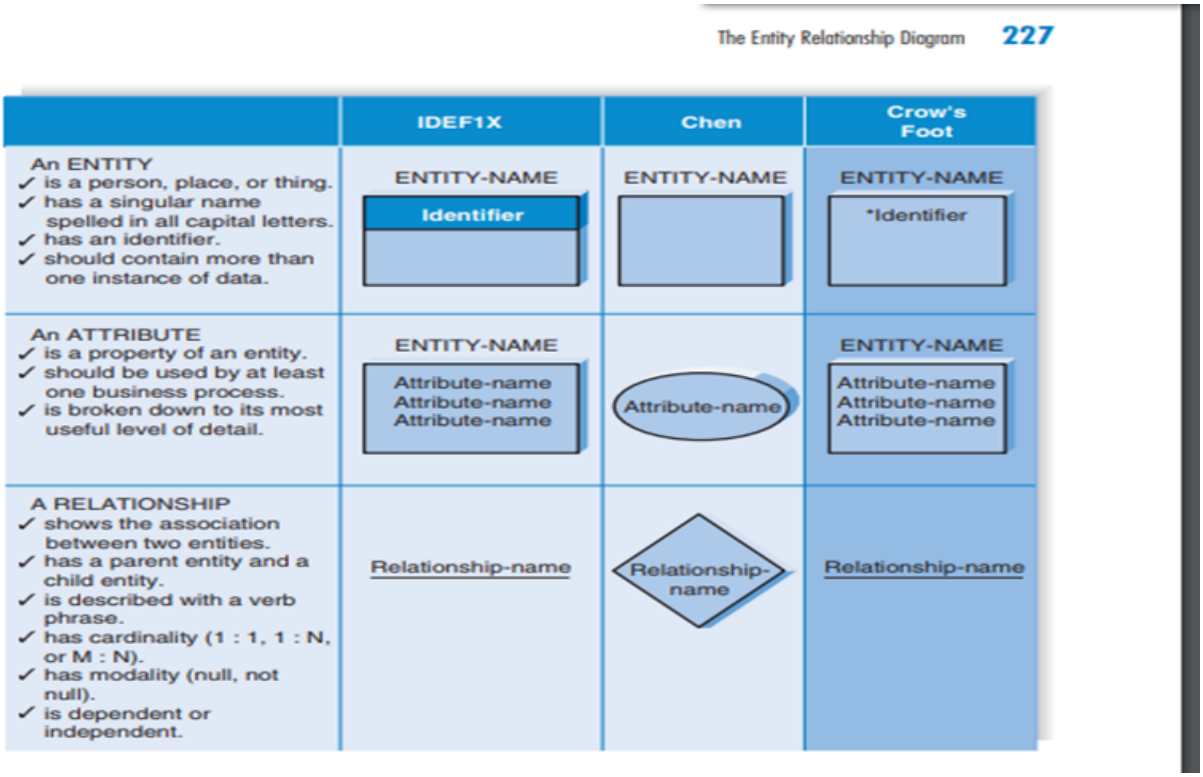
The entity is the basic building block for a data model. It is a person, place, event, or thing about which data is collected—for example, an employee, an order, or a product. An entity is depicted by a rectangle, and it is described by a singular noun spelled in capital letters. All entities have a name, a short description that explains what they are, and an identifier that is the way to locate information in the entity.

Attribute

An attribute is some type of information that is captured about an entity. For example, last name, home address, and e-mail address are all attributes of a customer. It is easy to come up with hundreds of attributes for an entity (e.g., a customer has an eye color, a favorite hobby, a religious affiliation), but only those that actually will be used by a business process should be included in the model. Attributes are nouns that are listed within an entity. Usually, some form of the entity name is appended to the beginning of each attribute to make it clear asto what entity it belongs

Relationship

Relationships are associations between entities, and they are shown by lines that connect the entities together. Every relationship has a parent entity and a child entity, the parent being the first entity in the relationship, and the child being the second. Relationships should be clearly labelled with active verbs so that the connections between entities can be understood.



	IDEF1X	Chen	Crow's Foot
An ENTITY ✓ is a person, place, or thing. ✓ has a singular name spelled in all capital letters. ✓ has an identifier. ✓ should contain more than one instance of data.	ENTITY-NAME Identifier	ENTITY-NAME	ENTITY-NAME *Identifier
An ATTRIBUTE ✓ is a property of an entity. ✓ should be used by at least one business process. ✓ is broken down to its most useful level of detail.	ENTITY-NAME Attribute-name Attribute-name Attribute-name	Attribute-name	ENTITY-NAME Attribute-name Attribute-name Attribute-name
A RELATIONSHIP ✓ shows the association between two entities. ✓ has a parent entity and a child entity. ✓ is described with a verb phrase. ✓ has cardinality (1 : 1, 1 : N, or M : N). ✓ has modality (null, not null). ✓ is dependent or independent.	Relationship-name	Relationship-name	Relationship-name

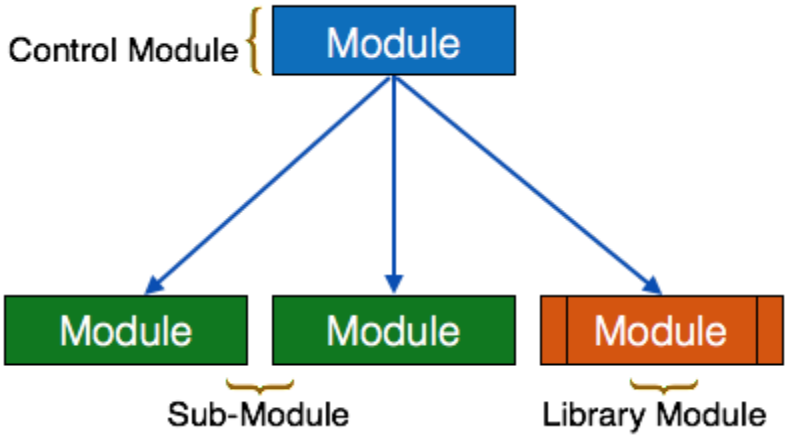
Structured charts

Structured chart is a chart derived from Data Flow Diagram. It represents the system in more detail than DFD. It breaks down the entire system into lowest functional modules, describes functions and sub-functions of each module of the system to a greater detail than DFD.

Structure chart represents hierarchical structure of modules. At each layer a specific task is performed.

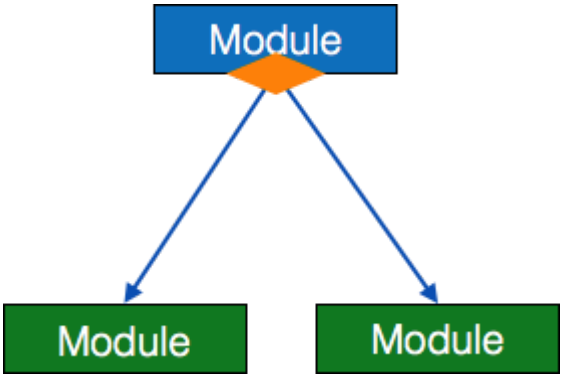
Symbols used in construction of structure charts are:

- **Module** - It represents process or subroutine or task. A control module branches to more than one sub-module. Library Modules arere-usable and invokable from

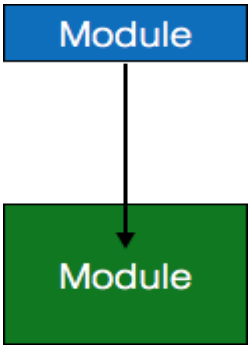


any module.

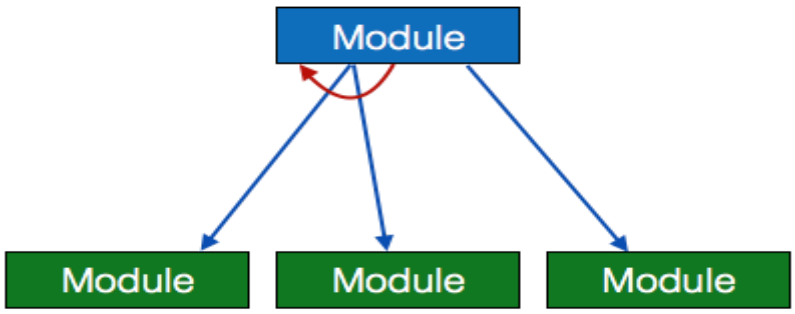
- **Condition** - It is represented by small diamond at the base of module. It depicts that control module can select any of sub-routine based on some condition.



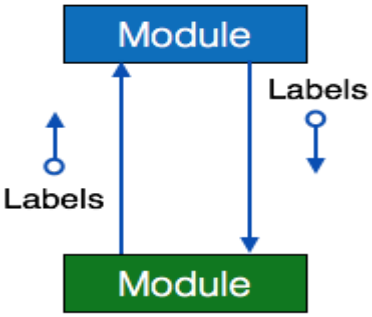
- **Jump** - An arrow is shown pointing inside the module to depict that the control will jump in the middle of the sub-module.



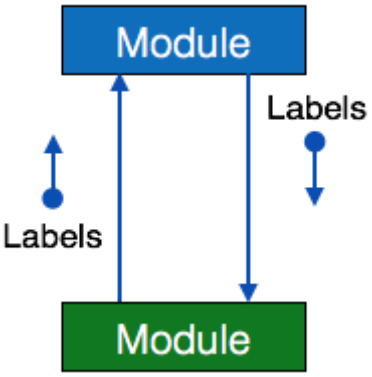
- **Loop** - A curved arrow represents loop in the module. All sub-modules covered by loop repeat execution of module.



- **Data flow** - A directed arrow with empty circle at the end represents data flow.



- **Control flow** - A directed arrow with filled circle at the end represents control flow.



TOPIC 8: SYSTEM IMPLEMENTATION

Implementation is the process of converting the manual or old computerized system with the new developed system and making it operational without distracting the functionality of the organization.

It's the final phase in the SDLC is the *implementation phase*, during which the system is actually built (or purchased, in the case of a packaged software design and installed). This is the phase that usually gets the most attention, because for most systems it is the longest and most expensive single part of the development process.

Significance of the implementation phase

- New system is made operational hence organizational processes are improved
- Improvement of employee skills through training
- Better organizational image

Procedure of system implementation

a) Hardware selection, acquisition and installation
b) User training
c) File conversion/creation
d) Changeover

System implementation techniques

There are four common techniques to implement a system:

(i) Direct change-over
(ii) Parallel change-over
(iii) Phased change-over

(iv) Pilot change-over

Direct

The old system ceases its operation and the new system commences operation the next day. The old system is made redundant in all its aspects. The method is applicable in the following circumstances:

- When the new system is small and simple
- When both the new and old system are substantially different
- When extra staff to oversee or undertake parallel running of both systems are unavailable
- When the management has complete confidence that the new system will work

The advantages of a direct changeover are:

- Relatively cheap
- Prevents the weaknesses of the old system from being passed over to the new system
- Reduces system implementation duration

Its disadvantages are:

- It is very risky especially if the new system fails. The cost of switching back to the old system will be high
- If not properly planned, it may interrupt user organization operations and bring confusion amongst staff members

Parallel

This is a method where new and old systems are allowed to run side by side or simultaneously until it is proved beyond reasonable doubt that the new system is working and all the benefits are realized. It is suitable when the new system is sophisticated and a very careful changeover is required or when the development team has little confidence in the new system and where there are more staff to cope with the operations of both system running in parallel.

Its advantages are:

- Users become familiar with the new system prior to the actual changeover which may enhance their efficiency
- The organization is exposed to less risks in case the new system fails
- There would be less interruption and inconveniences in the organization operations during the changeover period.

The disadvantages of this method are:

- It is an expensive method
- It might delay system implementation schedule or period

Phased

The method involves implementation of a system on step-by-step approach. This implies that only a portion of the system is implemented initially. Other portions are implemented in phases. For example if it has modules for finance, production and human resource management, then the finance module is implemented first, then the production and lastly the human resource management module.

Pilot

It involves installation of new system but using it only in one part of the organization on an experimental basis. E.g. a bank wishing to computerize its operations may install a computerized system on one branch on experimental basis. When the system is proved to be successful, it is transferred to another branch and after some time to another etc until the entire bank is computerized. Any refinement that ought to be done on the system should be done before it is installed in the next branch.

NB: The whole system is implemented on a section of the organization.

Both phased and pilot changeover methods have the following advantages and disadvantages.

Advantages are:

- Allows a new system to be implemented quickly with minimum costs
- Allow training of personnel on the new system during implementation
- They cause minimum interruption to company operations during system's implementation
- The peak demands are lighter on the end user and the operational environment
- They are less costly
- The risks associated with errors and system failure are minimized

The disadvantages include:

- Interfacing both the old and new system may usually bring problems
- There may be additional costs associated with running both systems at the same time

The change over plan should include the following:

(i) Time limit for the parallel run
(ii) Instructions on error handling procedures
(iii) Instruction on how to cope with major problems in the new system

TOPIC 9: MAINTENANCE AND REVIEW

As the last phase of the system development life cycle, it involves making use of feedback from end-users to make changes.

It involves changing part of the system according to the recommendations of the post implementation review team.

This also involves addressing any bugs that may still be present in the system.

System maintenance must continue to help make improvements to the now fully-implemented information system.

Causes of system maintenance include:

- Defects in the system after its delivery. This involves any errors or bugs in the newly implemented system e.g. use of wrong formula within a system
- Environment change e.g. a government tax policy may change which would influence a change of payroll system
- A change in user requirement. A business organization exist in a changing environment, therefore the user requirements change e.g. a payslip in a payroll system may initially be required to show the employee corporate share amount. Employees may feel that such information should not appear in the payslips and thus influence a change of the system.
- Poor documentation of the system. It makes it difficult for one to understand the system, and also to change it should there be a need to do so. A system may be changed and its documentation rewritten in order to improve it is maintainability.

System maintenance is carried to improve the system adaptability and flexibility.

Flexibility involves minor changes in a system in order to cope up with the growth in business transaction volume. Adaptability involves changing a system in order to benefit the user from advances in both software and hardware technology.

The process of the system maintenance should be controlled by the system analyst.

When a manager or a user suggests a change to the system regardless of the reasons:

a) The analyst should prepare diagrams and estimate the impact
b) The change control board decides whether or not to implement the change.
c) If change could take place, the analyst modifies all the documentation by merging the diagram and estimates into the existing problem and designs specification.
d) The programmers and testing teams are responsible for incorporating any change into the programs. They test the system to ensure that no errors or problem are introduced as a result of change.
e) Once the change is satisfied as default free, the revised system is adopted immediately.

Types of system maintenance

a) Corrective maintenance
b) Perfective maintenance
c) Adaptive maintenance
d) Preventive maintenance
e) Replacive maintenance

Corrective maintenance

It is usually a change effected in a system in response to detected problem or error. It is objective is to ensure that the system remains functional. It basically involves removal of errors on the already newly developed system. Example could be a failure in parts of the system.

Perfective maintenance

It is a change to perfect a system i.e. improve its performance in terms of response time to user request or to amend a system interface to make a system more user friendly.

Adaptive maintenance

Involves changing a system to take account of a change in its functional environment.

Preventive maintenance

Carried out on a system to ensure that it can withstand stress. It helps in ensuring data and software integrity.

Replacive maintenance

It is carried out on a system when a system becomes almost unmaintainable e.g. due to lack of documentation, poor design or age.

TOPIC 10: DOCUMENTATION

Software documentation is a description of software or system after its development. Software product therefore comprises of code and documentation. Documentation includes a wide range of technical and non-technical manuals, books, descriptions and diagrams relating to the use and operation of produced software. It is vital for software engineering to allocate adequate time to the software engineering particularly documentation throughout its development.

Documentation is produced for:

- System developer – who will depend on documentation from previous life cycle stages to guide continued development and subsequent maintenance of software or system.
- Management – who use documentation from past projects to plan and understand current projects
- System users – who learn how to use software or system from its narrative description or documentation.

Objectives of good documentation

The following factors should be considered when preparing a good documentation:

- Documentation should be complete – This implies that all known aspects or components of documents should be included.
- Documentation should be consistent – Inconsistency will destroy the reader's confidence in the documentation. The biggest challenge is not consistency in the original documentation but maintaining consistency through all the changes the software may undergo.
- Documentation should be targeted at the right levels – i.e. for its intended audience e.g. a training manual demands as much from its readers as design documentation to the programmer.

There are two major reasons why software engineers dislike producing documentation:

(i) They do not see the need for it because it may indicate that one is new to the profession and has not yet had time to appreciate the benefits of documentation. It may indicate also that one is so wrapped up in pressure of the moment that long-range goals have become absurd.
(ii) They do not feel capable of doing it. Although sometimes the feeling of inadequacy derives from inability to talk about technical subjects with non-technical people.

Importance of system documentation

(i) It guides the development team at various stages of the development life cycle
(ii) Can be used as a system backup copy to recover the system should something happen to its implementation
(iii) It aids or assists during system maintenance since it guides in identification of system modules to be changed
(iv) It can effectively provide a checklist of items to be covered during subsequent system audit or system maintenance
(v) It guides against loss of system understanding particularly when the author leaves the company or dies
(vi) It may act as a training guide or document to new programmers, analysts or users who may join after system implementation

Contents of system documentation

(i) Table of contents – acts as a document index
(ii) Introduction – indicates the system capabilities and constraints or limitations

(iii) System specification – it specifies the conceptual system in terms of process, data structures, files etc.

(iv) A list of files to be used by the system – used for reference should something go wrong when the system is live

(v) Test data – shows the data used to evaluate system functionality

(vi) Recover procedures – guides the user on how to recover the system should something go wrong when the system is running

(vii) Samples of input and output data – these helps the user to identify errors when they occur during system live-running

(viii) Back-up procedures – it advices the reader on how to make security copies of files for use to recover the system in case something goes wrong during system live-running

(ix) Contacts – Address or phone number to be used by the operator to seek help if other options fail

Types of Documentation

There are three fundamentally different types of user documentation: reference documents, procedures manuals, and tutorials.

Reference documents (also called the help system) are designed to be used when the user needs to learn how to perform a specific function (e.g., updating a field, adding a new record). Typically, people read reference information only after they have tried and failed to perform the function. Writing reference documents requires special care because users are often impatient or frustrated when they begin to read them.

Procedures manuals describe how to perform business tasks (e.g., printing a monthly report, taking a customer order). Each item in the procedures manual typically guides the user through a task that requires several functions or steps in the system. Therefore, each entry is typically much longer than an entry in a reference document.

Tutorials teach people how to use major components of the system (e.g., an introduction to the basic operations of the system). Each entry in the tutorial is typically longer still than the entries in procedures manuals, and the entries are usually designed to be read in sequence, whereas entries in reference documents and procedures manuals are designed to be read individually.

Regardless of the type of user documentation, the overall process for developing it is similar to the process of developing interfaces . The developer first designs the general structure for the documentation and then develops the individual components within it.

TOPIC 11: SYSTEM ACQUISITION

There are three primary ways to approach the creation of a new system:

a) Develop a custom application in-house;
b) Buy a packaged system and (possibly) customize it; and
c) Rely on an external vendor, developer, or service provider to build or provide the system.

Each of these choices has its strengths and weaknesses, and each is more appropriate in different situations. There may be obvious characteristics of the project that suggest the preferred acquisition strategy.

The following sections describe each acquisition choice in turn, and then present criteria you can use to select one of the three approaches for your project.

Custom Development

Many project teams assume that *custom development*, or building a new system from scratch, is the best way to create a system. For one, teams have complete control over the way the system looks and functions. Let's consider the purchasing process for Tune Source. If the company wants a Web-based feature that links tightly with its existing CD sales system, the project may involve a complex, highly specialized program. Alternatively, Tune Source might have a technical environment in which all information systems are built from standard technology and interface designs so that they are consistent and easier to update and support. In both cases, it could be very effective to create a new system from scratch that meets these highly specialized requirements.

In some situations, the challenges being addressed with the new system are so significant and demanding that serious systems engineering is required to solve them. In these cases, the developers really cannot find a packaged solution that is capable of meeting the project requirements, and a custom development project is the only real viable choice.

Custom development also allows developers to be flexible and creative in the way they solve business problems. Tune Source may envision the Web interface that takes customer digital music purchases as an important strategic enabler. The company may want to use the information from the system to better understand its customers who

buy digital music over the Web, and it may want the flexibility to evolve the system to incorporate technology such as data-mining software and geographic information systems to perform marketing research. A custom application would be easier to change to include components that take advantage of current technologies that can support such strategic efforts.

Building a system in-house also builds technical skills and functional knowledge within the company. As developers work with business users, their understanding of the business grows and they become better able to align information systems with strategies and needs. These same developers climb the technology learning curve so that future projects applying similar technology become much easier.

Custom application development, however, requires a dedicated effort that includes long hours and hard work. Many companies have a development staff that is already overcommitted. Facing huge backlogs of systems requests, the staff just does not have time for another project. Also, a variety of skills—technical, interpersonal, functional, project management, modeling—all have to be in place for the project to move ahead smoothly. IS professionals, especially highly skilled individuals, are quite difficult to hire and retain.

The risks associated with building a system from the ground up can be quite high, and there is no guarantee that the project will succeed. Developers could be pulled away to work on other projects, technical obstacles could cause unexpected delays, and the business users could become impatient with a growing timeline.

Packaged Software

Many business needs are not unique, and because it makes little sense to reinvent the wheel, many organizations buy *packaged software* that has already been written, rather than developing their own custom solution. In fact, there are thousands of commercially available software programs that have already been written to serve a multitude of purposes. Think about your own need for a word processor—did you ever consider writing your own word processing software? That would be very silly, considering the number of good software packages available for a relatively inexpensive cost.

Similarly, most companies have needs, such as payroll or accounts receivable, that can

be met quite well by packaged software. It can be much more efficient to buy programs that have already been created, tested, and proven, and a packaged system can be bought and installed quickly compared with a custom system. Plus, packaged systems incorporate the expertise and experience of the vendor who created the software.

Let's think about the needs that Tune Source will have in its Digital Music Download system. One requirement is to have a simple, fast, and flexible processing place to deliver the purchased tunes over the Internet to the purchaser. Server-side download management software programs are available that are designed to optimize the delivery of file downloads. Some of these products are available for free, and in some products, these tools are incorporated into a overall shopping-cart capability as well. Tune Source will certainly need to consider this type of option as it considers alternatives for the Digital Music Download system.

Packaged software can range from small single-function tools, such as the server-side download manager, to huge all-encompassing systems, such as *enterprise resource planning* (*ERP*) applications that are installed to automate an entire business. Implementing ERP systems is a popular practice in which large organizations spend millions of dollars installing packages by such companies as SAP, Oracle, and Infor and then change their businesses accordingly.

Installing ERP software is much more difficult than installing small application packages, because benefits can be harder to realize and problems are much more serious.

One problem is that companies utilizing packaged systems must accept the functionality that is provided by the system, and rarely is there a perfect fit. If the packaged system is large in scope, its implementation could mean a substantial change in the way the company does business. Letting technology drive the business can be a dangerous way to go.

Most packaged applications allow for some customization or for the manipulation of system parameters to change the way certain features work. For example, the package might have a way to accept information about your company or the company logo that would then appear on input screens. An accounting software package could offer a choice of various ways to handle cash flow or inventory control so that it could support

the accounting practices in different organizations. If the amount of customization is not enough and the software package has a few features that don't quite work the way the company needs them to work, the project team can create a *workaround.* A workaround is a custom-built add-on program that interfaces with the packaged application to handle special needs. It can be a nice way to create needed functionality that does not exist in the software package. However, workarounds should be a last resort, for several reasons. First, workarounds are not supported by the vendor who supplied the packaged software, so when upgrades are made to the main system, they may make the workaround ineffective.

Also, if problems arise, vendors have a tendency to blame the workaround as the culprit and refuse to provide support.

Although choosing a packaged software system is simpler than going with custom development, it also can benefit from following a formal methodology, just as if you were building a custom application. The search for a software package should be based on the detailed requirements identified during analysis.

Systems integration refers to the process of building new systems by combining packaged software, existing legacy systems, and new software written to integrate these. Many consulting firms specialize in systems integration, so it is not uncommon for companies to select the packaged software option and then outsource the integration of a variety of packages to a consulting firm.

The key challenge in systems integration is finding ways to integrate the data produced by the different packages and legacy systems. Integration often hinges on taking data produced by one package or system and reformatting it for use in another package or system. The project team starts by examining the data produced by and needed by the different packages and systems and identifying the transformations that must occur to move the data from one to the other. In many cases, this involves fooling the different packages or systems into thinking that the data were produced by an existing program module that the package or system expects to produce the data, rather than by the new package or system that is being integrated.

Outsourcing

The acquisition choice that requires the least in-house resources is *outsourcing*, which means hiring an external vendor, developer, or service provider to create or supply the system.

The term *outsourcing* has come to include a variety of ways to obtain IT services and products. Outsourcing firms called *application service providers* (ASPs) supply software applications and/or software-related services over wide area networks or the Internet. In this approach to obtaining software, the ASP hosts and manages a software application, and owns, operates, and maintains the servers that run the application.

The ASP also employs the people needed to maintain the application.

Organizations wishing to use a software application contract with the ASP, who makes it available to the customer via a wide area network or the Internet, either installed on client computers or through a browser. The customer is billed by the ASP for the application either on a per-use basis or on a monthly or annual fee basis.

Obtaining access to a software package through an application service provider has many advantages. There is a low cost of entry and, in most cases, an extremely short setup time. The pay-as-you-go model is often significantly less expensive for all but the most frequent users of the service. Investments in IT staff can be reduced, and investments in specialized IT infrastructure often can be avoided.

Outsourcing firms are also available that will develop a custom system on behalf of the customer. There can be great benefit to having others develop your system. They may be more experienced in the technology or have more resources, such as experienced programmers. Many companies embark on outsourcing deals to reduce costs, whereas others see it as an opportunity to add value to the business.

For example, instead of creating a program that handles the purchasing process or buying a preexisting package, Tune Source may decide to let a Web service provider provide commercial services for them.

For whatever reason, outsourcing can be a good alternative for a new system; however, it does not come without costs. If you decide to leave the creation of a new system in the hands of someone else, you could compromise confidential information or lose control over future development. In-house professionals are not benefiting from the skills that could be learned from the project; instead, the expertise is transferred to the

outside organization. Ultimately, important skills can walk right out the door at the end of the contract.

Most risks can be addressed if you decide to outsource, but two are particularly important. First, assess the requirements for the project thoroughly—you should never outsource what you don't understand. If you have conducted rigorous planning and analysis, then you should be well aware of your needs. Second, carefully choose a vendor, developer, or service with a proven track record with the type of system and technology that your system needs.

There are three primary types of contracts that can be drawn to control the outsourcing deal. A *time and arrangements* deal is very flexible because you agree to pay for whatever time and expenses are needed to get the job done. Of course, this agreement could result in a large bill that exceeds initial estimates. This arrangement works best when you and the outsourcer are unclear about what it is going to take to finish the job. You will pay no more than expected with a *fixed-price contract* because if the outsourcer exceeds the agreed-on price, he or she will have to absorb the costs. Outsourcers are very careful about clearly defining requirements up front, and there is little flexibility for change.

The type of contract gaining in popularity is the *value-added contract*, whereby the outsourcer reaps some percentage of the completed system's benefits. You have very little risk in this case, but expect to share the wealth once the system is in place.

Creating fair contracts is an art because you need to carefully balance flexibility with clearly defined terms. Needs often change over time, so you don't want the contract to be so specific and rigid that alterations can't be made. Think about how quickly technology like the World Wide Web changes. It is difficult to foresee how a project may evolve over a long period. Short-term contracts leave room for reassessment if needs change or if relationships are not working out the way both parties expected. In all cases, the relationship with the outsourcer should be viewed as a partnership in which both parties benefit and communicate openly.

Managing the outsourcing relationship is a full-time job. Thus, someone needs to be assigned full time to manage the outsourcer, and the level of that person should be appropriate for the size of the job. (A multimillion-dollar outsourcing engagement

should be handled by a high-level executive.) Throughout the relationship, progress should be tracked and measured against predetermined goals. If you do embark on an outsourcing design strategy, be sure to get more information.

Guidelines for outsourcing.:

a) Keep the lines of communication open between you and your outsourcer.
b) Define and stabilize requirements before signing a contract.
c) View the outsourcing relationship as a partnership.
d) Select the vendor, developer, or service provider carefully.
e) Assign a person to manage the relationship.
f) Don't outsource what you don't understand.
g) Emphasize flexible requirements, long-term relationships, and short-term contracts.

Summary of factors influencing selection of system acquisition method

When to Use When to Use a When to Use
Custom Development Packaged System Outsourcing

Factor	When to use custom development	When to use Packaged system	When to use outsourcing
Business need	The business need is unique.	The business need is common.	The business need is not core to the business.
In-house experience	In-house functional or technical experience exists	In-house functional experience exists	In-house functional and technical experience does not exist.
Project skills	There is a desire to build in-house skills.	The skills are not strategic.	The decision to outsource is a strategic decision.

Project management	The project has a highly skilled and a proven methodology.	The project has a project manager who can coordinate vendor's efforts.	The project has a highly skilled project project manager at the level of the organization that matches the scope of the outsourcing deal.
Time frame	The time frame is flexible.	The time frame is short.	The time frame is short or flexible.

TOPIC 12: ICT PROJECT MANAGEMENT

Terminologies

Project

- It is a temporary endeavor undertaken to create a unique project, service, or result"
- It is a planned undertaking that has a beginning and an end, and how well the organization applies its resources to its projects and other work
- It can also be defined as a temporary sequence of unique, complex and connected activities having one goal or purpose and that must be completed within a specific time, budget and according to specifications

Management

- It is the process of planning, organizing, staffing, directing and controlling an activity

Project management

- It is defined as the organizing and directing of other people to achieve a planned result within a predetermined schedule and budget. (Satzinger, Jackson & Burd).
- It is the application of knowledge, skills, tools and techniques to project activities in order to meet or exceed users' and stakeholders' needs and expectations from the project. This definition is recommended by Project Management Institute (PMI) Standards Committee

Software project management

Software Project Management is the collection of techniques used to develop and deliver various types of software products. This developing discipline traditionally includes technical issues such as:

a) The choice of software development methodology.

b) How to estimate project size and schedule.

c) Which programming development environment to use.

This discipline also includes management issues such as:

a) When to train personnel.

b) What are the risks to the project success, and
c) How to keep the project on schedule.

These choices are then embodied in a software project management plan. Software project management address both the process of software development and the desired functional characteristics of the final software development and the desired functional characteristics of the final software product. A complete software project management plan is the design, implementation, control and test strategy for a software development process.

Importance of ICT project management

ICT project management tools

There are a variety of tools and techniques which a project manager can use to ensure that a successful IT project is delivered on time and within budget.

Many of these tools focus on scheduling skills, resources, tasks and time management.

These tools are often software based and are known as 'project management tools'

Any software which helps the project manager to plan or monitor the project could be deemed to be a 'project management tool'. At the simplest level these tools range from a simple spreadsheet to keep a record of costs or a database to store lists of suppliers or even diary software to maintain a list of appointments.

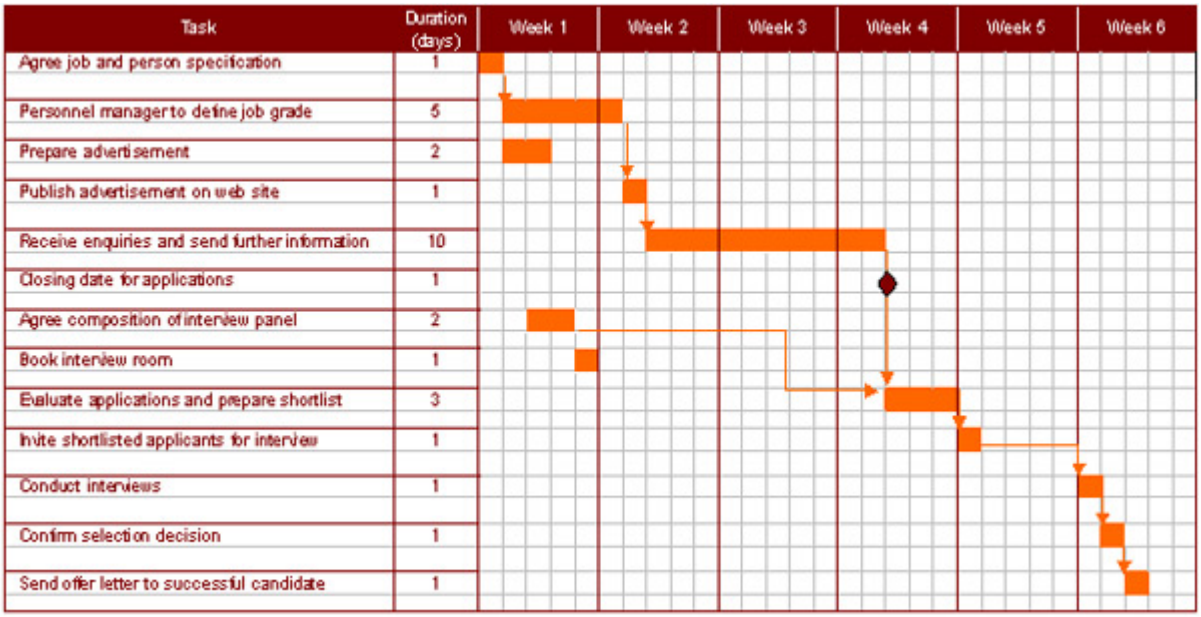
However, when we talk about 'project management tools and techniques' we tend to mean things such as:

- Gantt Charts
- Critical Path analysis (CPA)
- Program Evaluation Review Techniques (PERT)

- Project Management Software

Gantt Charts

A Gantt chart looks like a horizontal bar chart. It is used to help plan out and track specific tasks in a project.



Time is generally on the horizontal axis of the chart and the activities that need to be completed are arranged vertically from top to bottom in the order of their start dates.

Looking at the chart above you can see that the weeks are split into 5 days. The amount of days that each task will take are filled in as a block of colour.

You can see that task 1, 'agree job and person specification' will take 1 day to complete. That task has to be completed before task 2 and 3 can be started. Task 2, once started,

will take 5 days to complete. The arrow from task 2 to task 4 indicate that task 2 must be completed before task 4 can begin.

Task 3, task 7 and 8 can all be going on at the same time as task 2. This probably means that someone else is taking responsibility for them and so they can all run consecutively.

The purpose of a Gantt chart is to give the project manager a visual representation of what tasks will be happening at any given time, how long each task will take to complete and by when each task should be finished.

The four main features of a Gantt chart are:

- Milestones
- Resources
- Status
- Dependencies

Advantages

- Individual tasks can be shown
- The length of time required for each task is shown
- Tasks can be viewed against a calendar with their start, end and duration dates listed
- Links can be shown between dependent tasks i.e. what must be completed before the next task can begin
- They give a very quick visual view to what should be happening at any stage in the project
- Shows how long a project should take for completion
- Resources required for tasks can be linked to the tasks on the chart

Disadvantages

They can be too simplistic

They do not provide enough detailed information for complex projects

On a large project they can become so huge they are unmanageable

The time for each task must be estimated before the chart can be completed

Task dependencies can sometimes be hard to identify at the outset of a project

It is difficult to show where there is slack time in the project

Critical paths in the project might not be easily identifiable

Critical Path Analysis (CPA)

This is a useful Project Management Tool which enables managers to calculate the minimum length of time a project will need to be completed.

The idea behind it is that in order to complete a project some activities are dependent on other activities being finished in a certain order.

For example, if you are cooking Sunday lunch, you can't carve the meat up until you have cooked it and removed it from the oven.

CPA helps to identify every stage of a project and exactly how long each stage will take. The critical path of a project is the longest path from start to finish.

Once the critical path is known, the project manager can then plan staff and resources to ensure that the project does not fall behind schedule.

Working out the critical path

In order to calculate the critical path the following must be done:

- Identify all of the activities for the whole project

- Prioritise the activities in order of importance
- Identify which activities must be completed before another one can begin
- Identify the Earliest Start Time (EST) for activities
- Identify the Latest Finish Time (LFT) for activities
- Identify the FLOAT - those tasks which can be completed outside the critical path
- Identify the critical path

CPA - a worked example

The following is an example of how a CPA is constructed:

Example
XYZ Limited have decided to to create a questionnaire to ask customers about their products. They will use a mail shot to send out a similar survey to get the opinions of those that live outside of the area.

Each task (A - F) has been given an expected completion time (in weeks).

From: http://www.bizhelp24.com/small-business-portal/critical-path-analysis-3.html

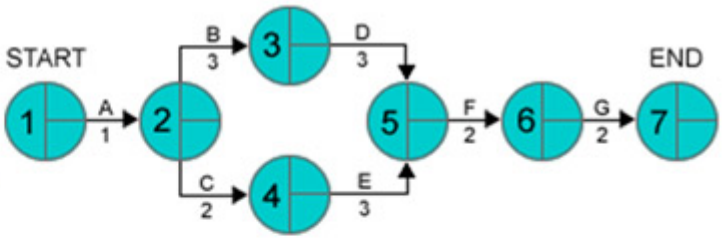
Task	Description	Order/Logic	Time
A	Plan Primary Research	To be completed first	1 wks
B	Prepare Mail Shot (Postal Survey)	Start when A is complete	3 wks
C	Prepare Questionnaire	Start when A is complete	2 wks
D	Send and Wait for Mail Shot Replies	Start when B is complete	3 wks
E	Issue Questionnaire	Start when C is complete	3 wks
F	Compile and Analyze Results	Start when D & E is complete	2 wks
G	Plan Selling Campaign	Start when D, E & F is complete	2 wks

Here is the beginning of a diagram for 'CPA'.

The numbers currently in the nodes (1 - 7) are only there to make following the diagram easier - nothing more.

The letters above the arrows represent the tasks (match with table)

Their completion times are shown below the arrow



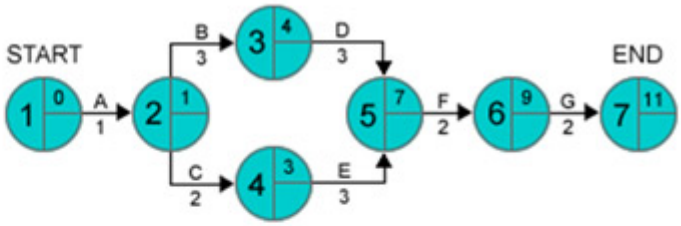
Task	Description	Order/Logic	Time
A	Plan Primary Research	To be completed first	1 wks
B	Prepare Mail Shot (Postal Survey)	Start when A is complete	3 wks
C	Prepare Questionnaire	Start when A is complete	2 wks
D	Send and Wait for Mail Shot Replies	Start when B is complete	3 wks
E	Issue Questionnaire	Start when C is complete	3 wks
F	Compile and Analyze Results	Start when D & E is complete	2 wks
G	Plan Selling Campaign	Start when D, E & F is complete	2 wks

First of all, we have to identify the **earliest starting time (est)** for each task.

This figure is then entered into the top right hand segment of the node

To work out the est's, we move from left to right on the diagram adding the previous node's est to the time the following task will take.

From:
http://www.bizhelp24.com/small-business-portal/critical-path-analysis-3.html



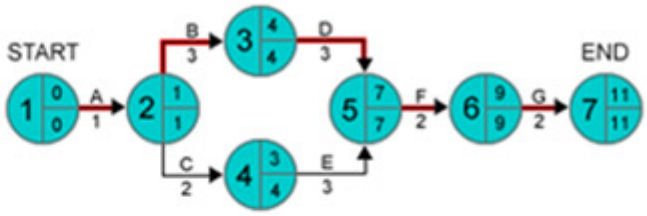
Task	Description	Order/Logic	Time
A	Plan Primary Research	To be completed first	1 wks
B	Prepare Mail Shot (Postal Survey)	Start when A is complete	3 wks
C	Prepare Questionnaire	Start when A is complete	2 wks
D	Send and Wait for Mail Shot Replies	Start when B is complete	3 wks
E	Issue Questionnaire	Start when C is complete	3 wks
F	Compile and Analyze Results	Start when D & E is complete	2 wks
G	Plan Selling Campaign	Start when D, E & F is complete	2 wks

Secondly, we have to work out the **latest finish time (lft)** for each task.

This is worked out similar to the est's except we subtract figures and work from right to left on the diagram.

The lft is then placed inside the bottom right hand segment of the node.

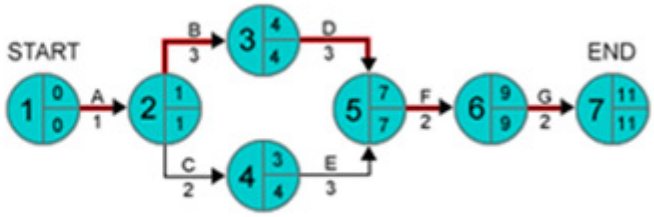
From:
http://www.bizhelp24.com/small-business-portal/critical-path-analysis-3.html



Task	Description	Order/Logic	Time
A	Plan Primary Research	To be completed first	1 wks
B	Prepare Mail Shot (Postal Survey)	Start when A is complete	3 wks
C	Prepare Questionnaire	Start when A is complete	2 wks
D	Send and Wait for Mail Shot Replies	Start when B is complete	3 wks
E	Issue Questionnaire	Start when C is complete	3 wks
F	Compile and Analyze Results	Start when D & E is complete	2 wks
G	Plan Selling Campaign	Start when D, E & F is complete	2 wks

Now the est's and the lft's have been calculated, we can determine the **critical path**.

This is found by recognizing those nodes where the **est = lft**.

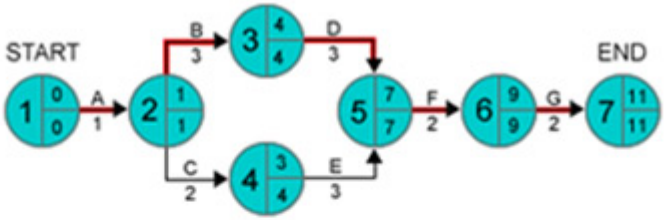


By looking at the above diagram, all the nodes have equal est and lft except for number 4.

By acknowledging this, we can identify the critical path as:

A - B - D - F - G (highlighted by the red line)

From: http://www.bizhelp24.com/small-business-portal/critical-path-analysis-3.html



There must be no delays in completing these tasks, otherwise the project completion time will be also be delayed beyond the expected 11 weeks.

This is not true for the tasks that do not lie on the critical path. A delay of a week for task 4 would not delay the whole project.

From: http://www.bizhelp24.com/small-business-portal/critical-path-analysis-3.html

CPA advantages

- Helps to increase the efficiency of how time and resources are used
- It enables time scales for the project to be planned
- It enables scheduling of tasks to take place in a logical order
- It means that one task isn't held up because its predecessor wasn't started on time
- It helps cash flow to be monitored and extra funds to be released when required
- It reduces waste

CPA disadvantages

- It needs skilled people to compile it in the first place
- Usefulness may be limited in very large scale or complex projects
- Accurate estimates of time are required

PERT

Short for Program Evaluation and Review Technique.

One of the weaknesses of the standard CPA technique is that it assumes that each activity will certainly be achieved in the stated time.

The PERT method extends the standard CPA method by trying to make a more realistic estimate of the time each activity will take.

This is based on first of all stating a *very* optimistic time for the activity, where the likelihood of achieving it in that time is less than 1%. Then consider a *very* pessimistic estimate of the time, where you are almost certain the activity will be achieved in that time. Then finally, state a time that you think is the *most likely* time it will take to complete. To summarize

- Most optimistic time (1% chance)
- Most pessimistic time (almost certain)
- Most likely time (high confidence)

Now use the formula below to work out the **estimated time** for the activity.

Estimated time = shortest time + (4 x likely time) + longest time

---

This is a weighted estimate formula that tends to push the estimate away from the shortest time. This estimated time is applied to each path in the CPA chart.

As the project progresses, the estimated time is replaced with the actual time.

This may cause the critical path to change due to an unexpected delay. In which case, if there is slack in the project, perhaps some of those resources can be deplayed to the critical path.

Criteria for evaluating ICT projects

A project is successful when:

- The resulting information system is acceptable to the customer
- The system is delivered on time
- The system is delivered within budget
- The system development process has a minimal impact on ongoing business operations

Signs of a failing ICT project

- Lack of User Input
- Incomplete Requirements & Specifications
- Changing Requirements & Specifications
- Lack of Executive Support
- Technical Incompetence

Reasons for ICT project failure

i. Failure to establish top-management commitment to the project
ii. Lack of organization's commitment to the system development methodology
iii. Taking shortcuts through or around the system development methodology
iv. Poor expectations management
v. Premature commitment to a fixed budget and schedule
vi. Poor estimating techniques
vii. Over-optimism
viii. Inadequate people management skills
ix. Failure to adapt to business changes
x. Insufficient resources
xi. Failure to manage the plan

Strategies for managing a failing ICT project

- Greater top management support
- More commitment from users
- More power and decisions making authority
- Greater financial control and flexibility
- Greater Control over staff resources
- Commitment to requirements and scope once specified
- More project management training
- Commitment to a stable project management method
- Alignment of IT project initiatives to business strategy
- Greater understanding of project management on the part of top management, project boards and clients
- Greater realism in setting targets. Several respondents railed against imposed rather than planned targets and deadlines
- Establishment of a supportive project/programme office.

TOPIC 13: EMERGING TRENDS IN SAD