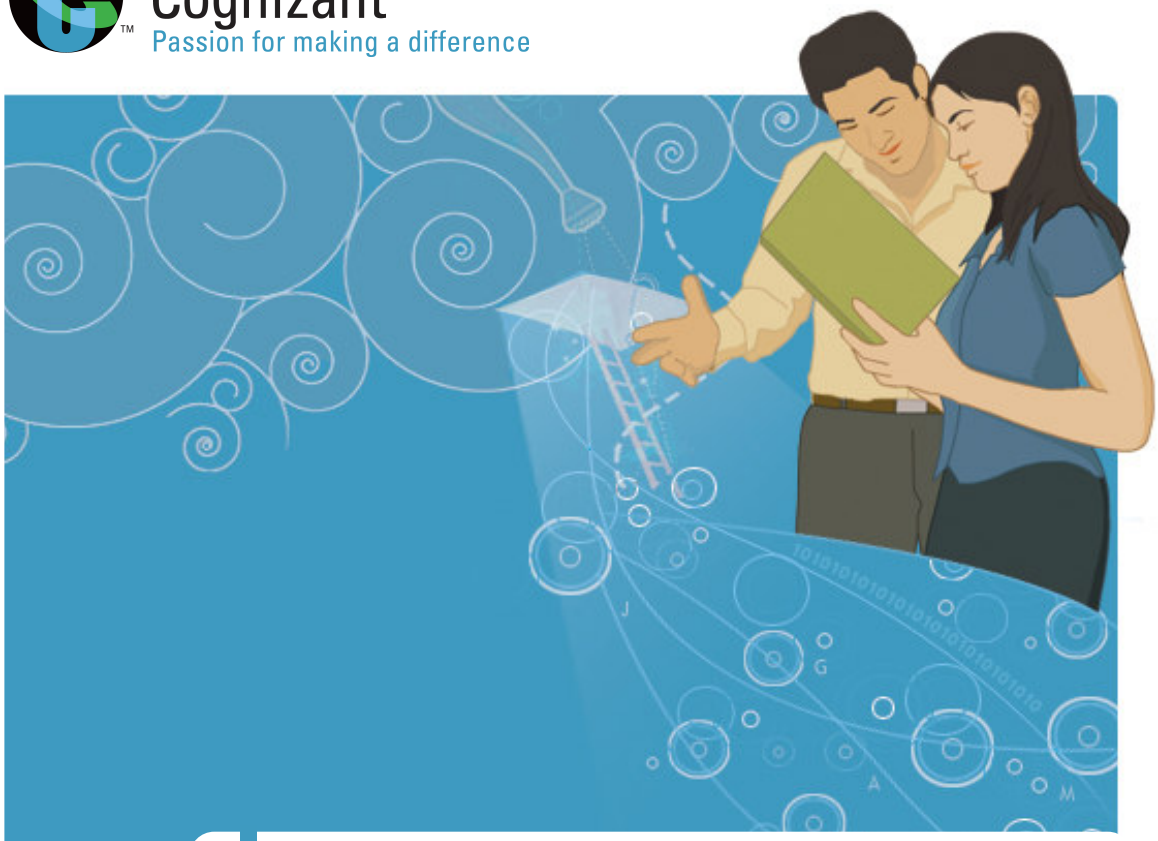
Cognizant
Passion for making a difference



Handout: System Analysis and Design

Version: SAD/Handout/0307/2.1
Date: 05-03-07

Cognizant
500 Glen Pointe Center West
Teaneck, NJ 07666
Ph: 201-801-0233
www.cognizant.com

TABLE OF CONTENTS

Introduction

About this Document

The document provides the basic knowledge of System analysis and Design that allows you to understand the information systems, how they can be designed, and developed. It also gives them a new perspective to look at information as a key resource in order to solve business problems.

Target Audience

In-Campus Trainees

Objectives

- Explain the basic concepts of Systems Analysis and Design
- Explain the various models for systems analysis and design
- Define Systems Analysis and Design
- Describe the model to design and develop an information system

Pre-requisite

Basic knowledge about data, information and data processing systems

Session 1: Information and Information Systems

Learning Objectives

After completing the session you will be able to:

- Appreciate the information needs of different levels of management
- Understand how information systems support the major business functions
- Differentiate between different types of information systems

Basic Definitions

Data are raw facts that are collected for any entity.

Information is processed data. Information attributes a meaning to the data.

Example: "E103" is merely a sequence of alphanumeric letters unless some meaning say employee Code is attributed to it, then only it becomes information. Hence it can be said:

Data: E103 Information: Employee Code E103

Management Structure and information needs

Information is today a key resource of any organization. Decision makers are aware that information is not just a byproduct of conducting business, rather it fuels business. Hence information systems are very critical to organizations. A successful information system can result in the success of the entire organization. Managers and staff, in all functional areas, not only use information systems, but also participate in their development.

Management in organizations exists on three broad horizontal levels: operational control, managerial planning and control (tactical), and strategic management. These levels can also be categorized as lower management, middle management and top management. Each level carries its own responsibilities and work in its own ways, towards achieving organizational goals.

Top level of the management includes the CEO, Managing Director and other functional heads of the organization. They are responsible for developing policy and for taking long term strategic decisions. Information such as trends in financial investment and human resource changes would be of interest to them. They need to take decisions when very little or no information may be available. They use their experience and this strategic information available to take appropriate decisions. They need summarized information. The information systems that support these types of managerial decision making are called ***Decision Support System (DSS)***.

The Middle level of management includes the divisional managers, departmental heads and key operational managers in an organization. Their decision making pertains to short term planning, control, and staffing. They need information about sales, weekly demand forecasts, distribution and transportation planning, production planning, materials requirement planning, cash flow projections, and annual financial statements. The information system that helps in decision making for operational control is called ***Management Information System (MIS)***.

The lower level of management includes line managers, sectional managers, sales officers, foreman, and accounts officers. They need operational information which is usually of short term and routine type. Examples are: daily employee attendance, current stocks available, overdue payments. The information system that supports day to day monitoring at field level is called ***Transaction Processing System (TPS).***

The organization process, information needs and the decision making characteristics for all the three levels can be summarized as in the table:

Table 1.1: Summary: Management levels and their characteristics

Management Level / Characteristics	Lower Management	Middle Management	Top Management
Decision Making	Structured Computational Routine	Semi structured	Unstructured Judgmental
Information	Operational Detailed, Low value Low risk	Management control Detail + summarized	Strategic planning Summarized High value, High risk
Organization process	Closed and static	Partially closed and partially static	Open and adaptive

Information Systems

Definition

A set of interrelated components that collect, manipulate, disseminate data and information, and provide feedback in order to meet a given objective. The purpose of Information system is to process input, maintain files of data about organization, and produce report and other outputs.

There are three types of information systems. Transaction Processing System (TPS)

- Management Information System (MIS)
- Decision Support System (DSS)

Transaction Processing System (TPS)

They are aimed at improving the routine business activities on which all organizations depend. They substitute computer based processing for manual procedure. TPS assists in carrying out the day-to-day, high volume activities / transactions of the organization. These transactions are processed using standard operating procedures. There are hardly any exceptions to these procedures. These routines are embedded in the computer programs that control the entry of data, processing of details, storage, and presentation of data and information.

Example: TPS includes Airline Reservation Systems, Banking Systems, job scheduling systems and queue monitoring system on a typical shop floor.

Management Information System (MIS)

MIS deals with supporting well-structured decision situations. MIS aims at improving operational efficiency. It utilizes transaction data obtained as the result of transaction processing, however MIS may also use other information that is developed internally and from outside the organization. This information is often represented as the tactical information. In a MIS the typical information requirement can be anticipated.

MIS helps in studying the decisions taken, the factors leading to a specific decision, and in developing reports that are useful in formulating future decisions (Demand reports, Exceptional Schedule reports).

Example: Payroll System, Sales Order System and Personnel management system

Decision Support System (DSS)

The DSS is sometimes referred to as Executive Information Systems (EIS). DSS are aimed at assisting managers who are faced with unique (non-recurring) and unstructured decision problems. DSS supports managers who are responsible for strategic planning and who take decisions based on long range considerations. DSS aids in decision-making in those situations where either information is not readily available or the information needs are difficult to predict.

DSS makes use of optimization models to predict / provide information. As information is acquired, it leads to the need of more related information. It provides interactive support for non routine decisions. The information provided by DSS facilitates decision making in high risk situations. It is impossible to pre-design report formats and contents in a DSS as it should be flexible enough to meet the changing needs of managers. The user must be able to request reports by defining their content and even by specifying how the information is to be produced. DSS is tool that aids in the process of decision-making but it cannot take decisions. A manager in addition to the information gained by DSS relies on his experience and intuition.

Example: A DSS that facilitates use of simulation and what if mechanism for forecasting customer preferences about a product in the forthcoming decade.

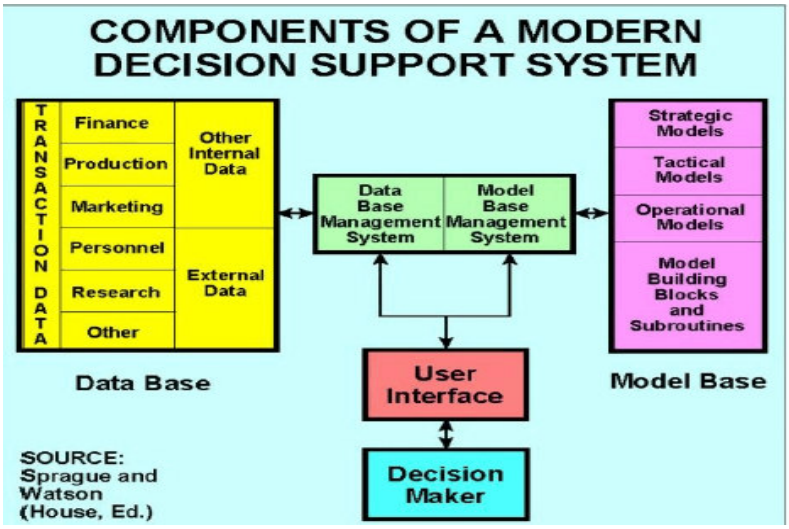


Fig. 1.1: Decision Support System

Summary

- There are three different types of information systems.
- These information systems are related to the different levels of management based on their information needs, organizational processes and their decision making style.

Test your Understanding

1. What type of information is required by the top level of management, and why?

2. What are Management Information Systems?

3. Match the following terms to the appropriate definitions.

___summary report	a. system designed to mimic human experts
___exception report	b. set of models
___on-demand report	c. information out of normal range
___database	d. set of related data
___model base	e. helps people make decisions
___decision support system	f. condensed information about all activities
___expert system	g. data when needed

Answer:

3. f, c, g, d, b, a, e

Session 2: Systems Analysis Fundamentals

Learning Objectives

This session will enable you to:

- Understand the basic concepts of Systems Analysis and Design
- Appreciate the characteristics of a systems analyst
- Learn different methods of information gathering

Systems Concept

A ***system*** is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective.

- A system must be designed to achieve a predetermined objective
- Interrelationships and interdependence must exist among the components.
- The objectives of the organization as a whole have a higher priority than the objectives of its subsystems

The key elements of a system are input, output, process(s), control, feedback and environment

Example: A manufacturing organization has subsystems: production, marketing, finance, administration, personnel, distribution, purchase, research and development.

Business process

A ***business process*** is a set of linked activities that create value by transforming an input into a more valuable output. Both input and output can be facts and/or information and the transformation can be performed by humans, machines, or both.

A business process can be decomposed into several sub-processes, which have their own goals, but also contribute towards achieving the goal of the parent process. The analysis of business processes typically includes the mapping of processes and sub-processes down to activity level.

Example: A cash process in an organization that involves receipt of the customer's payment, crediting the customer's account, and depositing the amount in the organization's account.

A business process is usually the result of a ***business process design*** or ***business process reengineering*** activity. Business Process Reengineering is a management approach that examines aspects of a business and its interactions, and attempts to improve the efficiency of the underlying processes.

Systems Analysis and Design

Systems Analysis and Design is the process of examining a business situation with the intent of improving it through better procedures and methods. It must ensure that the proposed information system meets user needs, can be delivered on time, and can be updated inexpensively.

The process of system analysis commences with system study that is conducted to learn the details of the current business situation, and to suggest alternative design strategies along with the costs and benefits analysis. It is conducted by the systems analyst. The system study involves the management, employees, and other end users of the system. The outcome of system study is the selection of a strategy for system development. A plan is developed based on the recommended strategy, which then forms the basis for system design.

Analysis is the collection of notations, methodologies and tools used to gather details and analyze a problem situation for recommending improvements to information system

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

The process of systems analysis and design tells us how to:

- Build information systems
- Analyze information system needs
- Design computer based information systems
- Solve systems problems in organizations

Systems Analyst

A ***Systems Analyst*** is the person responsible for researching, planning and recommending software and systems choices to meet an organization's business requirements. His primary task is to elicit needs and resource constraints for a given business situation, and to translate these into a viable operation

Personal Characteristics

- Ability to identify, define and recognize problems
- Logical approach
- Knowledge of professional practices and standards
- Persistent in his efforts
- Flexible in his outlook
- Possess good communication skills

Technical Characteristics

- Well versed in project management
- Strong technological perspective- IT tools and methods, Ergonomics, Systems concepts, Operations research

Tasks of Systems Analyst

- **Interviews:** The systems analyst conducts interviews to ascertain what information processes and procedures are performed/needed within the organization. What information is being maintained? What needs do users have?
- **Learns:** He learns about the nature of the business of the organization and its goals in order to recommend changes and understand the information processing tasks that are done by members of the organization.
- **Recommends:** He recommends software, hardware and communication equipment purchases for the organization to support its information processing systems.
- **Builds:** He builds a graphical representation of any existing information system to depict the requirements of the given system.
- **Uses:** He uses the existing system for requirements Analysis.
- **Monitors:** He monitors the development and installation of the new information system and the effectiveness of the new system.
- **Specifies:** He specifies the format of the data files, interfaces, reports, procedures to be developed, software to be purchased, security, and control procedures.
- **Plays different roles:** He plays the role of a consultant, supporting expert, change agent, psychologist, salesperson, an investigator, an architect, a motivator, and a politician.

Information gathering

Information gathering or “fact finding” or “data collection” is the formal process of using research, interviews, meetings, questionnaires etc., to collect information in order to determine requirements for the organization’s information system.

Sources

The information sources can be external and internal. External data sources include data about the environment and internal data is about the organization, its policies and the operating procedures.

Internal sources include information about:

a) Communications: Records of all kinds of meetings, discussions, correspondence and associated documents such as operational manuals, user manuals and policy statements.
b) Processes: Information regarding office procedures and operating procedures
c) Data: Transaction data, source data files, output data files, logical database, source data records, output data records.
d) Support information: Entity usage information and support facilities (H/W and S/W).
e) Test data: Specification of test requirements, test plan, test operations and test logs.
f) Cost data: Development cost, equipment cost, operational cost and personnel cost.
g) Performance Data: Information about timing, volumes, growth, documentation control, copy control,
h) Any amendment list

Parameters to be considered before data collection:

a) Objectives: What are the objectives of the project?
b) Background: Any background information about the project in question.
c) Terms of reference: What are the terms of reference? In other words what is the scope of the project? This sets the upper and lower bound for the project.
d) Constraints: What are the limitations or constraints under which the project development will take place?
e) Resources: What are the existing resources – man, material and information?

Methods

Information is gathered by:

a) Observation

Information can be collected by on-site observation. The data thus collected is highly reliable and relatively inexpensive. The systems analyst can understand the physical environment better. The problem with on-site observation is that the respondents may not be offering the true picture. The Un-interrupted observation is not feasible as on-site people may be busy at routine operations.

b) Administering Questionnaires

This method of data collection permits the people to answer queries at their own convenience. It is also relatively inexpensive. Another important factor is that it allows anonymity to the person and hence an objective viewpoint could be provided. This method gets a committed response. Its disadvantages include the fact that the number of respondents may be low. There is no guarantee about reliability. It offers no flexibility. It is difficult to prepare. Any wrong entry cannot be immediately clarified.

c) Interviews

In this methodology there is one to one communication between the systems analyst and the person concerned who is in-charge of the entity about which the information is being gathered. More information can be sought and the questions can be adapted to suit the interviewee. Interviews can be structured as well as unstructured. The disadvantage of this approach is that it is very time consuming. Interviewer's skill plays an important part. It requires exhaustive preparation by the interviewer.

Summary

- Analysis specifies what the system should do
- Design states how to accomplish the objective
- System analyst should possess a whole range of personal and technical traits in order to be successful
- Information can be gathered internally from the organization as well from external sources using different methods

Test your Understanding

1. What are the various tasks that a systems analyst performs?

2. What are the disadvantages of "interview" method of information gathering?

Session 3: Systems Analysis

Learning Objectives

After completing the session you will be able to:

- Understand the process of systems analysis
- Know about the System Requirement Specification (SRS) document

Feasibility

Feasibility is the measure of how beneficial the development of information system would be to an organization. Feasibility analysis is a management-oriented activity, by which alternative solutions are obtained and a specific solution is recommended for optimal performance.

The objective of a feasibility study is to provide the management with enough information to decide:

- Whether the project can be done?
- Whether the final product will benefit its intended users?
- What are the possible alternatives solutions?
- What are the costs and the benefits associated with each of the alternatives?
- Is there a preferred alternative?

After a feasibility study, management makes a go/no-go decision.

During the feasibility study look at:

- Present system- users, policies, functions, objectives.
- Problems with the present system- inconsistencies, functional inadequacies,
- And performance issues.
  - Objectives and other requirements for the new system- what needs to be changed?
  - Constraints including nonfunctional requirements on the system
  - Possible alternatives
  - Advantages and disadvantages of the alternatives

Feasibility analysis includes study of technical feasibility, economic feasibility and operational feasibility.

Technical Feasibility: Whether the project can be done with the available equipment, technology and personnel. Technical feasibility analysis questions whether the hardware, software, and other technologies needed for the project to exist in the organization or it has to be acquired or it needs to be developed?

Example: If a project requires coding in a new programming language and none of the personnel knows the language, the organization must consider the feasibility of training existing personnel or hiring consultants. If these options prove too costly or take too long, the project may not be technically feasible

Economic Feasibility: Economic feasibility study analyzes the viability of a solution in terms of revenues and investments. The questions that are asked are:

- Is the project viable, given the resource constraints?
- Are the benefits that accrue from the new system worth the costs?
- What are the savings that result from the system, including tangible and intangible ones?
- What are the development and operational costs?

Operational Feasibility: Operational feasibility looks at the operability and acceptability of any solution. The questions that are asked are:

- If the system is developed, will it be used?
- Are there any people oriented issues such as manpower problems and labour objections?
- Is there any conflict with the organizational policies
- Are there any issues of social acceptability, legality and adherence to government regulations?

Operational feasibility can be analyzed using the ***PIECES*** framework.

PIECES FRAMEWORK:

- **P**erformance: Does the system provide adequate throughput and response time
- **I**nformation: Does the system provide the end users with timely, pertinent, usefully formatted information
- **E**conomy: Does the system offer adequate service level and capacity to justify costs
- **C**ontrol: Does the system offer adequate controls to ensure integrity and security of data
- **E**fficiency: Does the system optimize the use of all available resources
- **S**ervices: Is the service reliable, desirable, flexible and expandable

After study, a feasibility matrix is created:

Table 3.1: Feasibility Matrix Template

Alternatives / Parameters	Candidate System 1	Candidate System 2	Candidate System 3
Operational Feasibility			
Technical Feasibility			
Economic Feasibility			
Ranking (1, 2, 3…)			

The best candidate system based on the ranking is selected.

Systems Requirement Specification (SRS)

System requirements specification document specifies the information requirements to be provided. SRS is obtained after excessive discussions with the user. The narrative of requirements by users can be too long and imprecise and it needs conversion to precise specifications. SRS does not specify the design of the information system to be developed.

Systems Requirement Specification (SRS) should

- Be complete and unambiguous
- Specify operational and strategic information requirements
- Attempt to eliminate future disputes between users and Analyst
- Use Graphical aids understood by users

How to create a SRS document

- Step1: Analyze statement of requirements
- Step2: Identify physical entities.
- Step3: Identify documents which are received/sent by each office
- Step4: Draw a physical document.

SRS can be specified using graphical tools also.

Graphical Specification Tools for SRS

a) ***Physical- Document flow diagram*** describes the physical flow of document in a system. It depicts various entities or offices and documents generated/transmitted by these entities

b) ***Logical- Data Flow Diagram*** describes the functional logic of a system. It depicts the flow of data between various processes

Contents of a SRS document

- Introduction
  - Background, purpose, scope
- Current System
  - Overview, details, problems
- Proposed System
  - Overview, boundary, details, advantages, disadvantages, performance requirements, security considerations, portability, maintenance, fallback
- Hardware Environment
- Software Environment
- Change-over Plan
- Appendices: DFD, contents of data store, List of I/O with contents

Summary

- The systems analysis process involves identification and evaluation of the feasibility of alternative solutions, ranking alternatives, and preparing a system proposal for management approval. The most optimal alternative is recommended
- There are three types of feasibility- operational, technical and economic
- SRS document specifies the requirements of the system to be developed or enhanced

Test your Understanding

1. What do you understand by technical feasibility?

2. What is the PIECES framework for evaluating the operational feasibility?

Session 5: Data Flow Diagrams

Learning Objectives

After completing the session you will be able to:

- Understand how to depict the user's requirement in terms of a graphical model
- Provide an overview of what data a system processes, what transformations are performed on them, what data are stored in the system, what results are produced and what is the flow of data

Structured Analysis

Structured analysis is a method for modeling the components of a system using graphical symbols. ***Data Flow Diagrams*** (DFD) show the flow of data into the system and between ***processes*** and ***data sto***res. A data store is a repository for data and it can be manual, digital or temporary. In preparing the model, the analyst emphasizes what occurs, not how it is accomplished. Thus the focus is on logical rather than physical aspects of the system.

Process modeling is a graphical representation of the processes that capture, manipulate, store, and distribute data between a system and its environment and among system components.

Data-flow Diagram (DFD) is an example of process modeling. DFD is a graphical system model that shows all of the components of an information system in one diagram: ***inputs, outputs, processes and data stores***. It illustrates movement of data between external entities and the processes and data stores within a system.

Data Flow Diagram and Levels of Abstraction

DFD may reflect the processing at either a higher level (more general view of the system) or at a lower level (a more detailed view of one process). These different views of the system, higher level versus low level, create levels of abstraction. DFD is a modeling technique that breaks up the system into a hierarchical set of increasingly more detailed models. Higher level processes in a DFD can be decomposed into separate lower level DFDs.

Context Diagram

Context Diagram is a data-flow diagram to represent the scope of an organizational system. It shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system. It summarizes all processing activities within the system by a single process symbol. It describes highest level view of a system. All external agents and all data flows into and out of a system are shown in the diagram.

Features of Context Diagrams

- Shows system boundaries
- Represents the system scope within a single process
- Shows external agents that supply or receive data from the system and that are outside the system scope but not the data stores
- This highest level of DFD des not show any details of what takes place within the system

Data Flow Diagram

Data Flow

Data Flow depicts data that are in motion- moving as a unit from one place to another in the system. It is represented as an arrow. A data flow has only one direction of flow between symbols. A fork means that exactly the same data go from a common location to two or more processes, data stores, or external entities- sources or sinks. A data flow cannot go directly back to the same process it leaves. A data flow to a data store means update into database. A data flow from a data store means retrieve or use. A data flow has a noun phrase label.

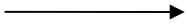


Fig. 5.1: Data Flow

Data Store

It depicts data at rest and may represent data in a file folder, computer-based file or a notebook. Data Store is represented as a rectangle with the right vertical line missing or with two horizontal parallel lines. Data store has a noun phrase label. The label should include name of the store as well as the number. Data can neither be moved from one store to another, nor from an outside source to a data store. A "join" means that exactly the same data come from any two or more processes, data stores or sources/sinks to a common location.

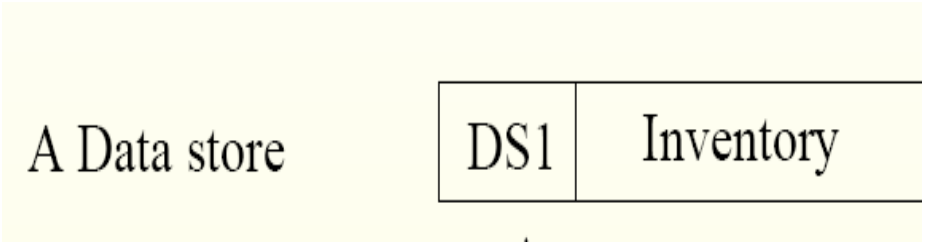
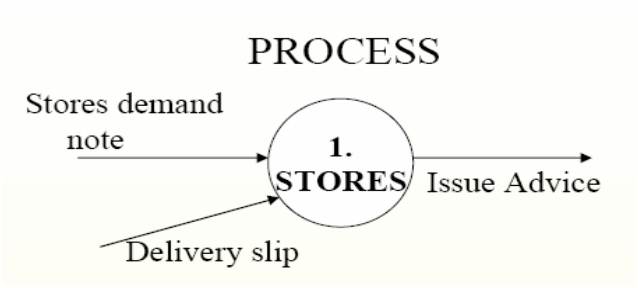


Fig. 5.2: Data Store

Process

A process depicts work or actions performed on data so that they are transformed, stored, or distributed. It defines rules- algorithms or procedures, for transforming inputs into outputs. It is drawn as a circle or rectangle with rounded corners. A process has a verb phrase label, and includes process number as well as name. No process can have only outputs (a miracle). No process can have only inputs (black hole).



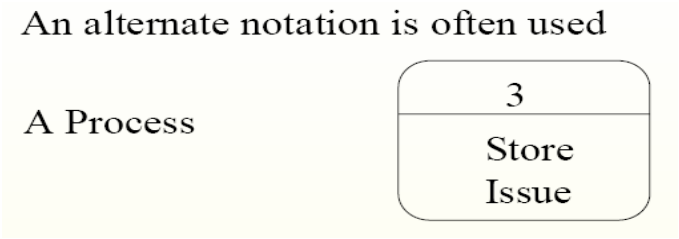


Fig. 5.3: Process

External Entity (Source/Sink)

It depicts the origin and/or destination of the data and is represented as a square symbol. A source/sink has a ***noun phrase label*** and represents a person or an organization. An external entity is outside the boundary of a system. It provides data outputs or accepts data inputs Data cannot move directly from a source to a sink. In some notation, it forms part of DFD while in some it is not a part of a DFD

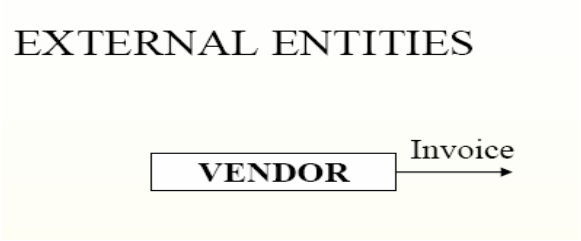


Fig. 5.4: External Entity

Guidelines for drawing DFDs

- Completeness:
  - DFD must include all components necessary for system
  - Each component must be fully described in the project dictionary or Computer Aided Software Engineering (CASE) repository
- Consistency:
  - The extent to which information contained on one level of a set of nested DFDs is also included on other levels
- Timing:
  - Timeframe is not represented on DFDs
  - Best way to draw a DFD is as though system is an ongoing one> It is not restricted by a specific time frame.
- Iterative Development:
  - Analyst should expect to redraw DFD several times before reaching the closest approximation to the system being modeled
- Top down development:
  - Start from context diagram and reach the lowest logical level of decomposition the primitive DFD
- Meaningful names should be chosen for process, flows, data stores and external entities
- Numbering does not imply sequencing of processes
- To avoid cluttering, data stores and data flows can be repeated on a diagram
- Each process on a DFD must be formally defined and numbered
- Each process can transform data but cannot create data
- Data Store can not create data. It can store previously created data

Example: The context diagram for a College Mess Management System that interacts with the external entities- students, vendors, mess secretary, chief warden and mess manager. Mess manager is responsible for deciding the items required for the day. The mess secretary decides the menu. Vendors supply the required items. Students pay the mess charges. Warden is responsible for the collection of overdue. The figure shows context diagram for a Mess Management system.

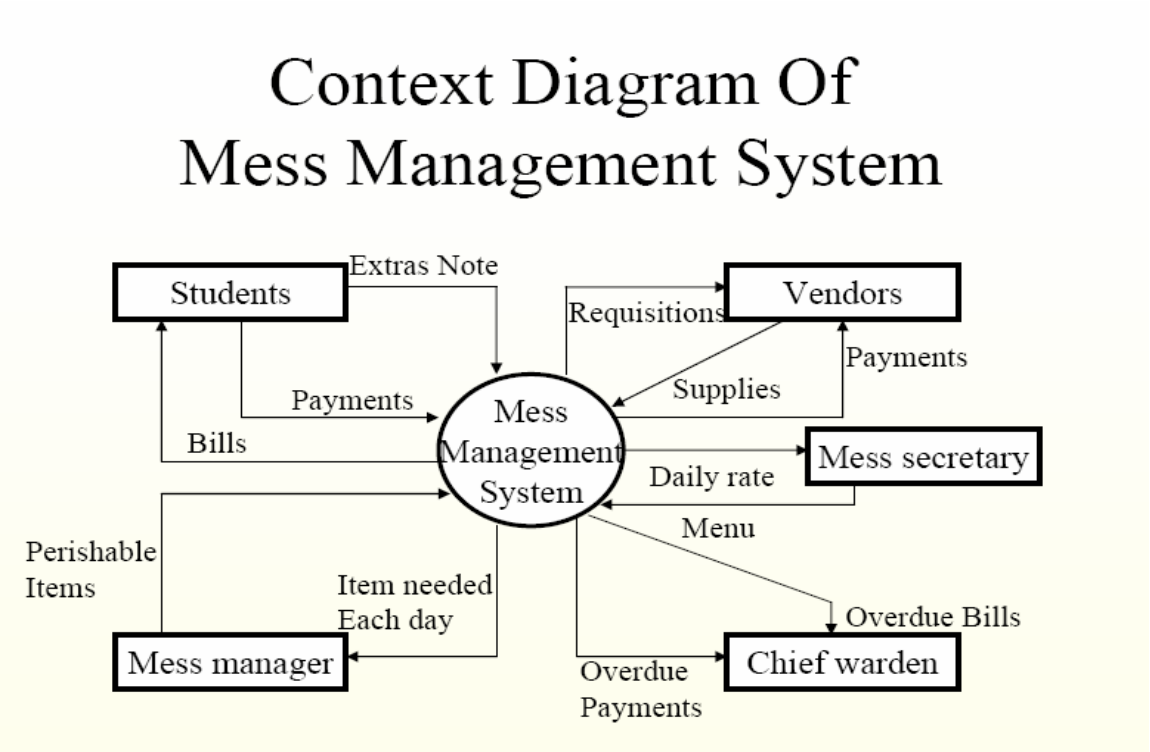
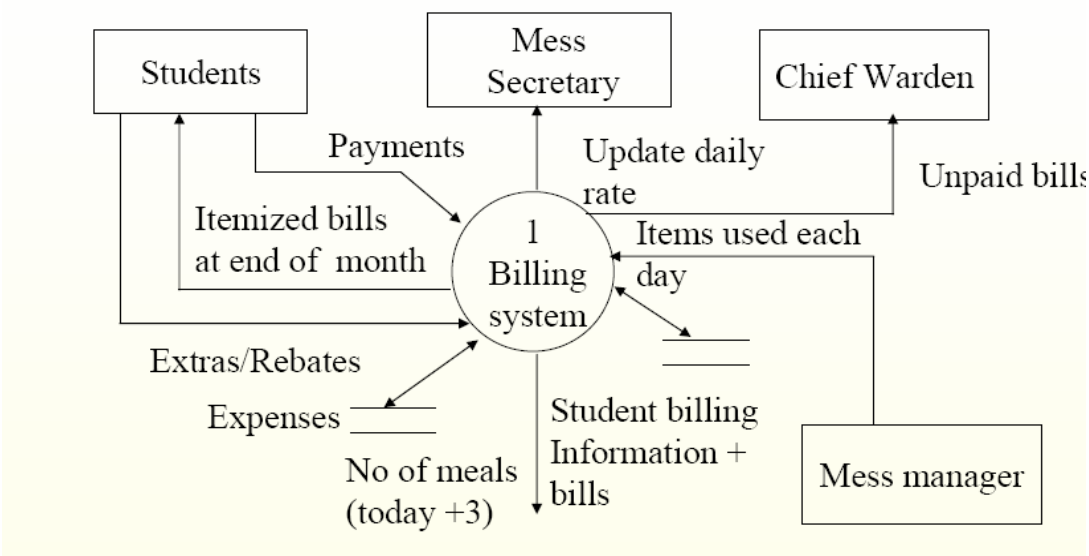


Fig. 5.5: A context diagram for a Mess Management System

Example: The level 1 DFD for the College Mess Management System is shown in figure



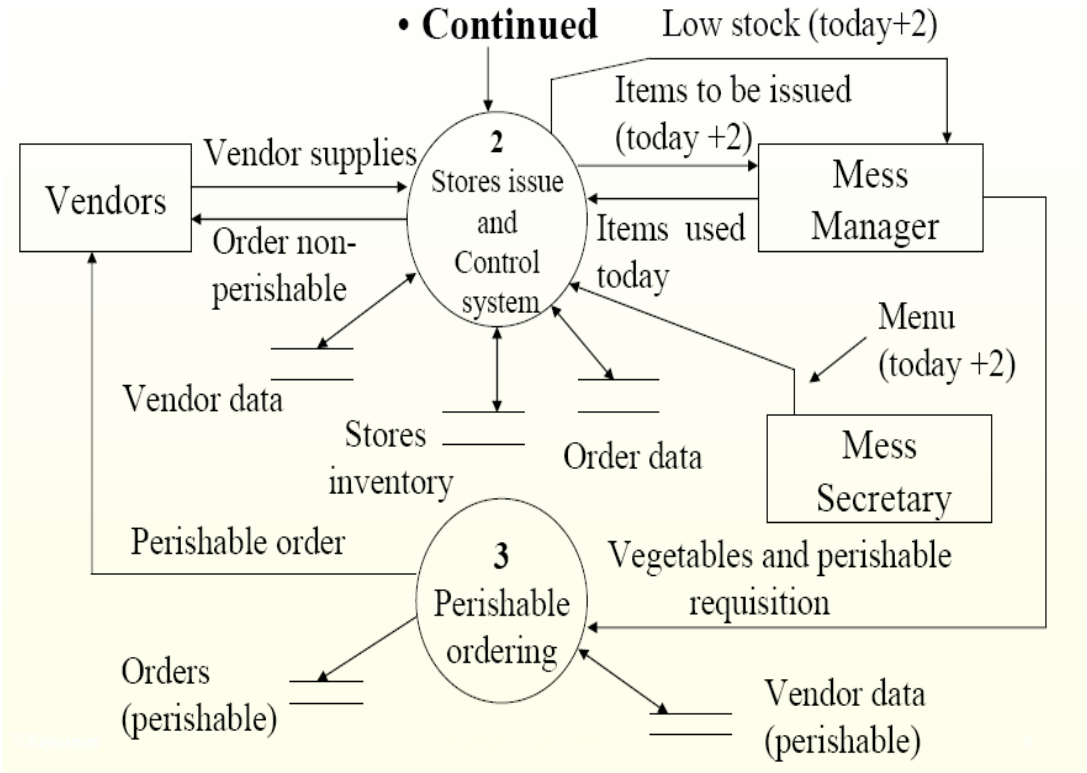


Fig. 5.6: A level 1 DFD for a Mess Management System

Example: The level 1 DFD for the College Mess Management System is shown in figure

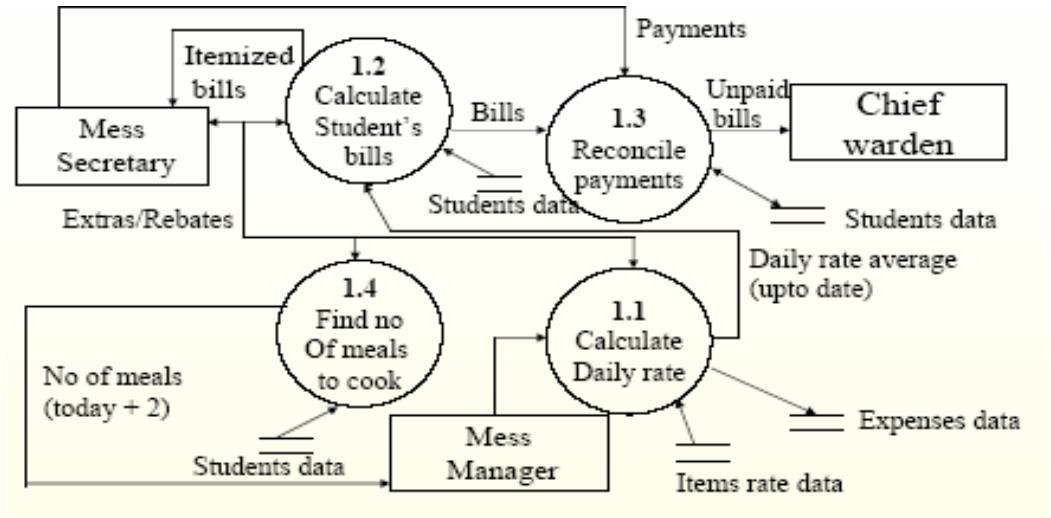


Fig. 5.7: A level 2 DFD for a Mess Management System

Decomposition of DFDs

Functional Decomposition: It is the act of going from one single system to many component processes. It is a repetitive procedure. The lowest level is called a primitive DFD.

Level-n Diagrams: A DFD that is the result of *n* nested decompositions of a series of sub processes from a process on a level-0 diagram

Balancing DFDs: When decomposing a DFD, you must conserve inputs to and outputs from a process at the next level of decomposition

Stop decomposition when:

- Each process has been reduced to a single decision, calculation or database operation
- Each data store represents data about a single entity
- The system user does not care to see any more detail
- Every data flow does not need to be split further to show that data are handled in various ways
- Each business transaction has been reduced to a single data flow
- There is a separate process for each choice on all lowest-level menu options

Gap Analysis: The process of discovering discrepancies between two or more sets of data-flow diagrams or discrepancies within a single DFD is called Gap Analysis. Inefficiencies in a system can often be identified through Data Flow Diagrams

DFD Tools

SmartDraw - A chart drawing program

ConceptDraw - Windows and MacOS X data flow diagramming tool

Dia - Open source diagramming tool with DFD support

MacAandD - Macintosh tool for multi-level data and control flow diagrams, data dictionary with verification and balancing reports

Microsoft Visio - Windows diagramming tool which includes very basic DFD support (Images only, does not record data flows)

WinAandD - Windows tool for multi-level data and control flow diagrams, data dictionary with verification and balancing reports

OmniGraffle - OS X tool similar to Visio,

QSEESuperLite - Windows integrated modeling environment.

Summary

- DFD is a graphical system model that shows all the main requirements for an information system in one diagram: *inputs, outputs, processes, and data stores*
- Functional Decomposition is the process of going from a single system to many component processes. The lowest level is called a primitive DFD

Test your Understanding

1. State True or False
   a. A process can be drawn as a circle or as a rectangle with rounded corners.
   b. A data store cannot move from an outside source to a data store.
   c. A data flow has bi-directional flow.

2. What are the rules for decomposition of DFDs?

Answers

1. a. True b. True c. False

Session 7: Approaches to System Development

Learning Objectives

After completing this session you will be able to:

- Learn the various approaches to systems development

Introduction

Organizations use different strategies for system development in different business situations. The three commonly used development strategies are "***Systems Development Lifecycle Method", "Systems Prototype Method"*** and ***"Spiral Method".***

The Systems Development Lifecycle Method, popularly called the Waterfall model, involves preliminary investigation of the problem, collection of data, determination of requirements followed by design and implementation of the system. This method has well defined stages. The major disadvantage is its long development time.

A prototype is a version of an information system having the essential functional features but not all the details of the user interface. In this strategy both system analyst and the user evaluate the results, identify deficiencies, missing features and adjustments required. After each iteration of development, the system is evaluated and enhancements are made. Thereby the final working system emerges. The Spiral Model was designed to include the best features from the waterfall and prototyping models and introduces a new component risk-assessment.

Systems development Life cycle (SDLC) Method

SDLC is classically considered as a set of activities that analysts, designers and users carry out to develop and implement an information system.

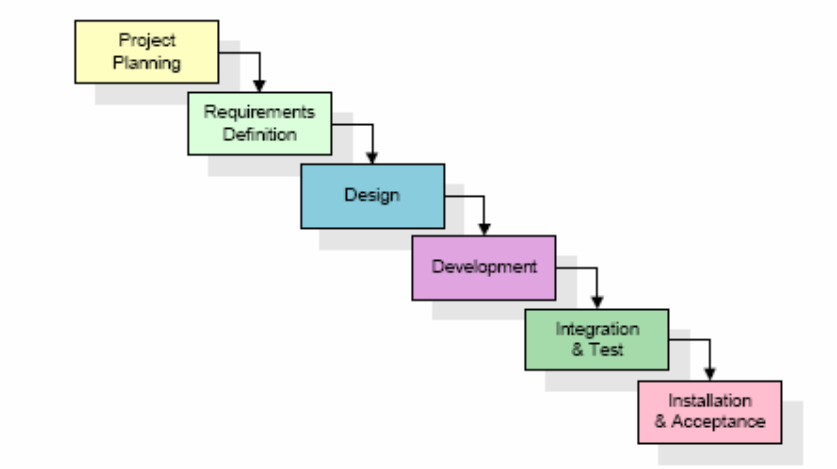


Fig. 7.1: SDLC

Software development lifecycle method comprises:

1. **Preliminary investigation (Project planning):** Preliminary investigation commences after a request for information system development or if any enhancement has been made. It involves the following activities:
   iii. Request Clarification: The project request is examined to determine precisely what the originator of the request wants. This activity ensures that the project request is clear and concise
   iv. Feasibility Study: This activity determines that the system requested for is feasible. It includes technical feasibility, economic feasibility and operational feasibility
      - Technical Feasibility: Whether the project can be done with the available equipment, technology and personnel or whether new technology will be required
      - Economic Feasibility: Are there sufficient benefits in creating the system to make the cost acceptable
      - Operational Feasibility: Whether it will be easily made operational or whether there will be resistance from the users

   iii. Request Approval: Those requests that are desirable and feasible are approved. The costs, completion time and personnel requirements are estimated and the schedule is prepared.

2. **Requirements Definition (Systems Analysis):** In this stage the entire system requirements determination is assessed in detail. Answers are sought for various key questions-what is being done? How is it being done? What is its frequency and volume? How is the quality? Does a problem exist and what is its cause? In order to elicit responses to these questions, the analyst has to interact with all the stakeholders namely managers, employees. This interaction can be through interviews, questionnaires or on site observations. Based on the information gathered, the analyst identifies the features the new system- information it should produce, operational features such as processing controls, response times and input and output methods.

3. **Systems Design:** This process includes specifications for meeting the system requirements. The reports and the display outputs are identified. The display is sketched using available automated tools. It also involves description of data to be input, calculated or stored. Individual data items and calculation procedures are written in detail. The storage devices are selected. The procedures to produce outputs are written. The design specification document is created that depicts information using charts, tables and special symbols. Hence, at the end of the design phase, the analyst presents the programmer with a clear and complete outline of the software specifications.

4. **Software Development:** Based on the cost involved, time available, programmers available, the software can be purchased and installed, purchased and modified or developed in-house. The programmers are responsible for documenting the program, providing explanation of how and why certain procedures are coded in specific ways. Documentation is very essential for testing and maintenance.

5. **Systems testing:** The software/ system are checked to ensure that it executes according to the pre-defined specifications and users expectations. Special test data are input for processing and the results are examined. The system is tested by a small subset of users to check for any unforeseen errors.

6. **Implementation (Installation and Acceptance):** It involves making the system operational– by installing the software, training the users and constructing any files that may be required. The software can be tested using a pilot or in a phased way or by completely stopping the old system and start using the new one. Implemented software/system has to be continually maintained – incorporating modifications and enhancements. The system is also evaluated continually to assess its strengths and weaknesses in the areas of operation, impact on organization and user-manager assessment.

Advantages

- Well defined stages
- Ensures thorough analysis of the existing systems problem
- Well structured development
- Availability of complete documentation
- Proper validation of all the input and outputs

Disadvantages

- Suitable only for applications having predictable requirements
- High transaction and processing volume
- It involves proper validation of input and other controls. It spans across several departments. It has a long development timetable

Prototyping Method

Prototyping is the process of building experimental system quickly and inexpensively for demonstration and evaluation. It overcomes the basic problem of SDLC i.e. long development time. It is basically a process of rapid software development to validate requirements.

A “Prototype” is a preliminary working version of information system for demonstration and evaluation. The focus is on ascertaining user reactions and expectations. The system analyst and the end user both use the prototype, evaluate the results, identify deficiencies and missing features and suggest changes. Hence, through successive iterations enhancements are made based on suggestions made after system evaluation and the final working system emerges. In this method, the user is given the opportunity to have realistic expectations from the system and influence changes at an early stage. This gives rise to improved system usability.

The steps in prototyping are:

- Identify user’s requirements
- Develop prototype
- Use prototype
- Revise and enhance prototype by expanding the list of system requirements based on the information gained by the users

There are two kinds of prototyping.

- **Evolutionary prototyping:** It is used to deliver a working system to end users. The development starts with those requirements which are best understood as shown in fig. 5.1

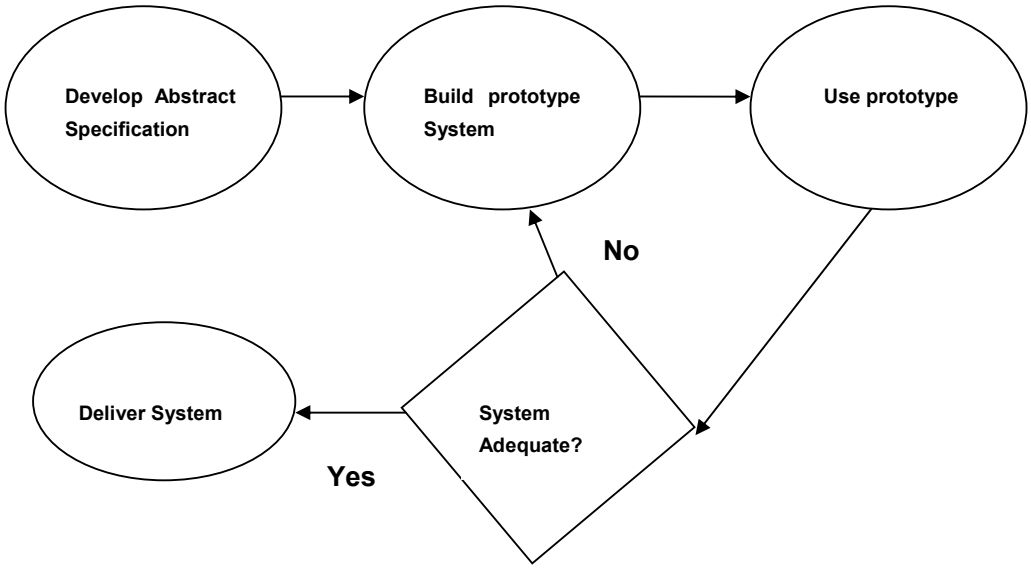


Fig. 7.2: Evolutionary Prototyping

- **Throw Away:** This kind of prototyping is to validate or to derive the system requirements. The prototyping process starts with those requirements which are poorly understood. The throw-away prototype should not be considered as a final system as some system characteristics may have been left out. There is no specification for long-term maintenance. The system is poorly structured and difficult to maintain

Advantages

- Used when it is not possible to predict user's requirements in advance
- A working system is available early in the process
- Misunderstandings between software users and developers are identified at the early stages of development
- Serves as a basis for deriving a system specification
- Developers receive quantifiable user feedback
- System implementation is easier since users know what to expect
- Has a better quality of design and it is easier to maintain.

Disadvantages

- Can lead to insufficient analysis
- User may perceive prototype as the final system
- May get a false impression that the final system can be easily and quickly delivered
- Can also lead to incomplete documentation.

Spiral Method

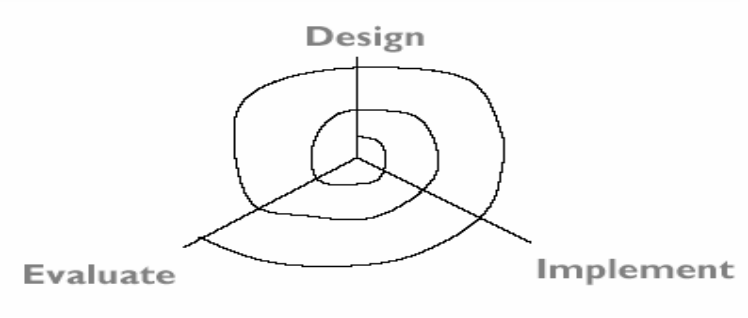


Fig 7.3: Spiral Model

The Spiral Model includes the best features from the SDLC and Prototyping models. It introduces a new component, risk-assessment. The term “spiral” is used to describe the process that is followed as the development of the system takes place. Similar to the prototyping model, an initial version of the system is developed, and then repetitively modified based on inputs received from customer evaluations. Unlike the Prototyping Model, however, the development of each version of the system is carefully designed using the steps involved in the SDLC Model. With each iteration around the spiral (beginning at the center and working outward), progressively more complete versions of the system are built. Risk assessment is done at each step in the development process as a means to evaluate each version for further development. If the user decides that the identified risks are too great, the project may also be scrapped. For example, if a substantial increase in cost or increase in project completion time is identified during one phase of risk assessment, the user or the developer may decide that it does not make sense to continue with the project, since the increased cost or longer timeframe may render the project infeasible.

The Spiral Model is made up of the following steps:

- **Project Objectives:** Objectives are determined, possible obstacles are identified and alternative approaches are weighed.
- **Risk Assessment:** Possible alternatives are examined, associated risks/problems are identified, and risks weighed in the consideration of project continuation.
- **Engineering and Production:** Detailed requirements are determined and the software is developed.
- **Planning and Management:** The user is given an opportunity to analyze the results of the version created in the engineering step and to offer feedback to the developer

Advantages

- Iterative design
- Early focus on users and tasks
- User analysis: who the users are
- Task analysis: what they need to do
- Involving users as evaluators, consultants and sometimes designers
- Constant evaluation
- Users are involved in every iteration
- Every prototype is evaluated

Summary

- SDLC is a model having well defined phases for structured development of software
- A prototype can be used to give end-users a concrete impression of the system's capabilities. There are two types of prototyping - throw-away prototyping and evolutionary prototyping
- Spiral model is an iterative model that emphasizes on the assessment of software risk

Test your Understanding

1. State True or False
   a. Phases in SDLC are fixed and inflexible

   b. Prototype can be used as the final developed system

   c. Spiral model combines both SDLC and Prototyping

Answers

a. True
b. False
c. True

Session 8: System Design

Learning Objectives

After completing this chapter, you will be able to:

- List the importance of system design in system development
- Design output, input, and interfaces

Design Objective

- To translate performance specifications into design specifications
- To help attain all the functional objectives of the new system namely reliability, response time, throughput, turnaround time, expandability, maintainability, user friendliness etc
- To specify the performance standards to be attained by the new system
- To design systems that satisfy the user's needs

Designing Effective Output

The output of an information system

- Conveys information about past activities, current status or projections of the future.
- Signals important events, opportunities, problems or warnings
- Triggers an action
- Confirms an action

The term output applies to any information produced by an information system. The system output can either be a report or a document or a message. Output contents may originate from a data store, can be transmitted from a process or are retrieved from an input source. The usage of the output determines its content, form and media. The primary objective of system outputs is to present relevant information for purpose of understanding and decision making.

Example – Printed report of list of sales made during the day Screen displaying details of employees, along with salary, education and experience Report cards of students Telephone Bills

Objectives

To:

- Determine the type of each output
- List specific outputs based on application design
- Specify necessary controls to protect the information provided in the output
- Design and prototype the report layout
- Design based on periodicity and volume of output
- Ensure timeliness in delivery

Output Characteristics

An output layout is the arrangement of items on the output medium. A layout is used as a mockup of the actual report or document as it would appear after the system is in operation. It shows the location of information. The most important items should be easy to find. All pages should have a title and page number and date. Print layout chart should read from left to right and from top to bottom. Columns are labeled and abbreviations avoided. The output design should take into consideration-

- Physical dimensions of the screen
- No of columns and rows
- Degree of resolution
- Number of colors available
- Methods of high-lighting
- Methods of intensity control

Output Specification contains all variable information and preprinted details.

Control Information: Summaries and totals and control breaks are indicated to emphasize specific points of information.

Internal Vs External Outputs

- Printed outputs are classified as internal and external outputs.
- Internal output is a printed report or document produced for use inside of the organization (includes types of reports discussed above)
- External outputs are printed documents, like statements, notices, form letters, and legal documents, produced for use outside of the organization (e.g. bank monthly statements you get in the mail). Some external outputs are called turn-around documents (they include a portion that is returned to the system as input, e.g. a bill with a payment stub you fill out and send back

Types of Output

Tabular: It is recommended to present details when only a few narrative comments or explanations are needed, when details can be described in discrete categories or when total and subtotals are appropriate. Certain information in the tabular formats should visibly stand out:-

- Exceptions to normal expectations,
- Major categories or groups of activities or entities
- Summaries

- Unique identification information
- Time dependent entities

Tabular formats are often appropriate for accounting information

Graphical: These are used for the display of business information. Graphics are used:

- To improve the effectiveness of output reporting for the targeted recipient
- To manage information volume
- To suit personal preferences

Types of Graphs

- **Pie charts** describe portions of a whole associated with a particular development or activity refer fig. 6.1
- **Area charts** show change in performance along a scale over multiple time periods. A horizontal scale indicates time and the vertical scale measures the units of interest. Several items overlaid on the same chart enable the reader to compare different items.
- **Curve chart** emphasizes the overall trend in expenditure
- **Step and Bar charts** show changes in categories. They do not connect individual data points instead they measure data points from the horizontal scale up to proper level on the vertical scale. In bar charts the individual periods are discrete and in step chart they are shown side by side. Refer fig 6.1
- **Maps** effectively show variations across geographical areas.
- **Icons:** They are the pictorial representation of entities described by the data. Properly selected icons communicate information immediately as they duplicate images that users are already familiar with. They eliminate the necessity for users to learn abbreviations, notation or special nomenclature. In contrast taking time to read labels or footnotes usually contributes to complexity. It can also substitute for tabular report and are especially useful in showing proportions or comparisons. Icons are commonly used in computer interfaces to represent documents, printers etc. One must use the same icon to represent identical concepts. One must avoid labeling the icons. One should use a layout that maintains space and avoids overcrowding between icons. Also one should maintain a common size among different types of symbols.

Example graphs

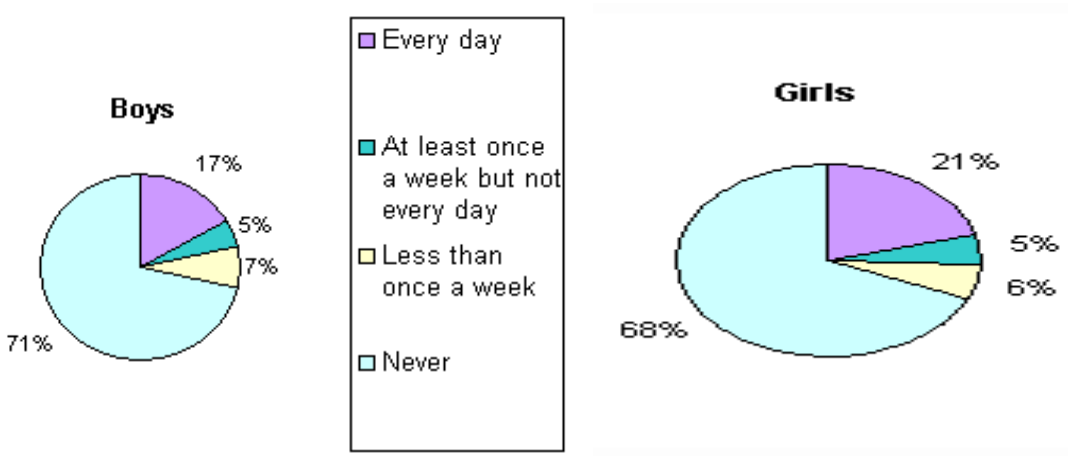
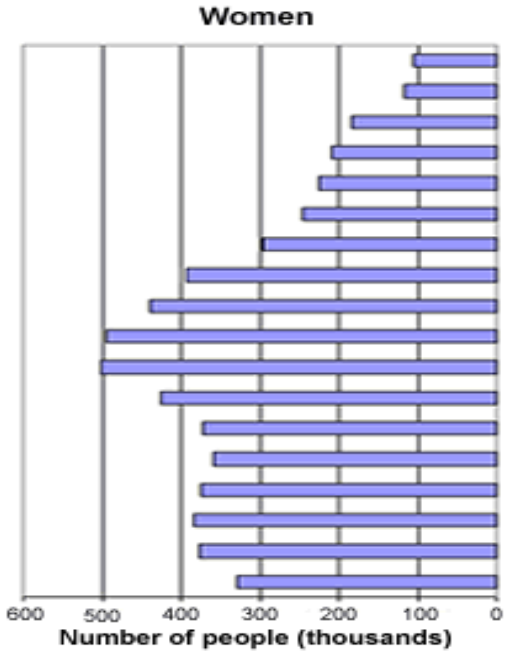


Fig 7.1: Smoking frequency of 15-year-olds on the secondary school track and field team depicted by a pie chart



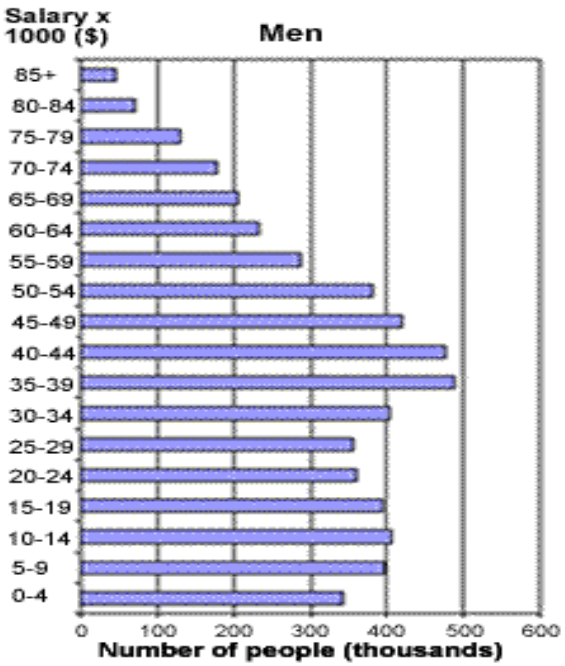


Fig 7.2: Earnings in a state categorized by sex depicted in a bar chart

Types of Reports

- Detailed reports
- Contain detailed information on business transactions
- A report may be for a single transaction or contain information about e.g. a particular account
- Summary reports
- Often used by middle management to track departmental or division performance
- Exception reports
- Conveys information about any extraordinary event
- Executive reports
- Normally used for strategic decisions by top management

Designing Effective Input

Objectives

- To choose a cost effective method of input
- To achieve the highest possible level of accuracy
- To ensure that the input is acceptable to and understood by the user

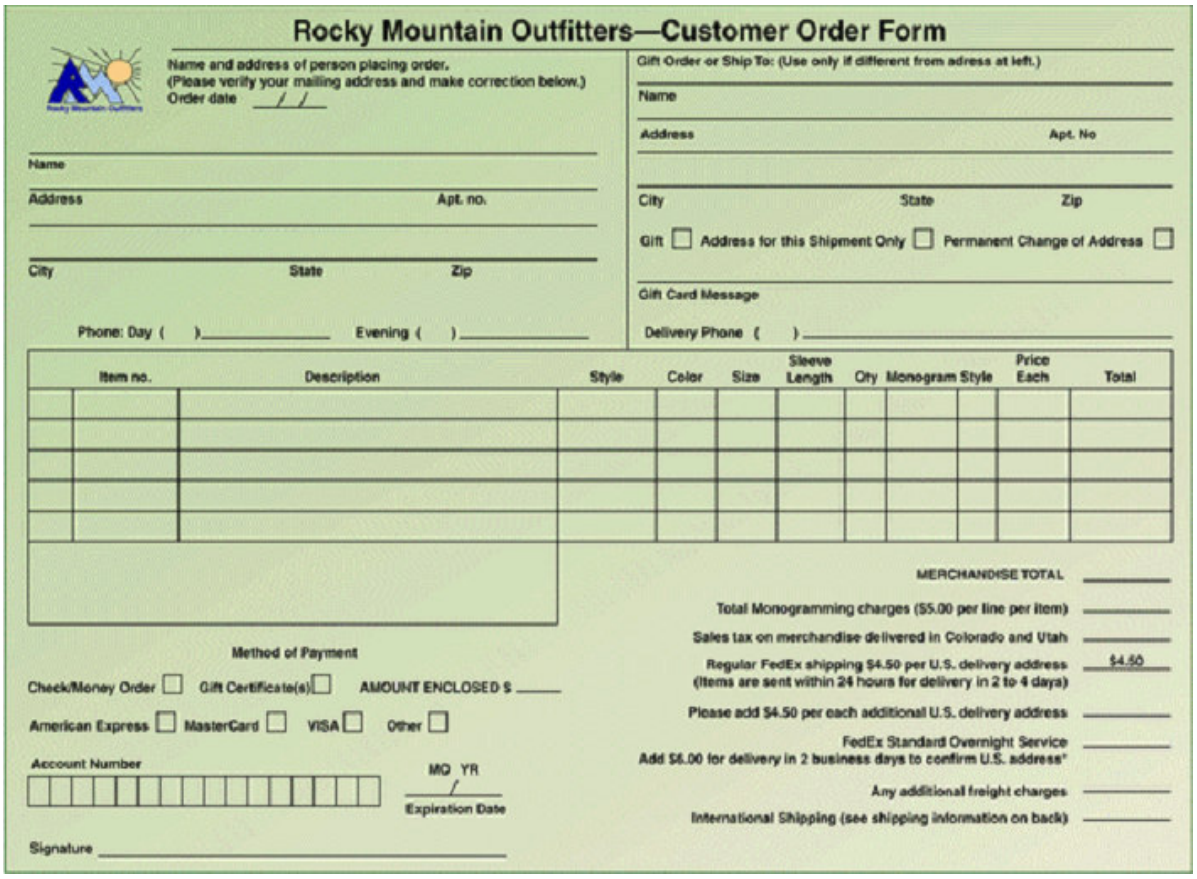
Input Design Stages

- Data Recording
  - Identify the devices and mechanisms that will be used to enter input
  - Capture the data as close to the originating source as possible
  - Use electronic devices and automatic entry whenever possible
  - Avoid human involvement as much as possible
  - If the information is available in electronic form anywhere, use it instead of reentering it
  - Validate and correct information at the time and location it is entered
  - Examples of input devices are magnetic card readers, bar code readers, optical character readers and scanners, touch screens and devices, electronic pens, digitizers etc.
- Data Identification and Conversion
  - Identify all system inputs, their content and transfer into a computer recognizable format.
  - Identify all information flows that cross the system boundary
- Data Control
  - Determine different controls necessary for each system input
- Actual Input Design
  - Design and prototype the electronic forms
- Data Transmission
  - Transmit or transport the data to computer
- Data Validation and Correction
  - Checking of the input data by the program at the point of entry and correcting the errors if any

Example – Application form for college admission, Attendance data for the employees Reservation form for ticket booking

Input Media:

- Source Document Conversion Devices: These devices convert the input to a computer acceptable form ex. Punch Card Reader, Key to tape, key to disk, key to cassette devices
- By- Product Data Capture devices: Capture data in a computer acceptable form as a by product of some essential operation. e.g. Billing machines, Cash register, Accounting machines
- Direct Data Capture Devices: These devices capture data without any conversion. For example, Optical mark reader, Magnetic Ink Character Reader, and so on.
- Online Data Entry Devices: Teletype writers, Visual Display Units, Audio response terminals, light pens, and so on.



Rocky Mountain Outfitters—Customer Order Form

Name and address of person placing order.
(Please verify your mailing address and make correction below.)
Order date ___/___/___

Name
Address Apt. no.
City State Zip
Phone: Day ( ) Evening ( )

Gift Order or Ship To: (Use only if different from adress at left.)
Name
Address Apt. No
City State Zip
Gift ☐ Address for this Shipment Only ☐ Permanent Change of Address ☐
Gift Card Message
Delivery Phone ( )

Item no.	Description	Style	Color	Size	Sleeve Length	Qty	Monogram Style	Price Each	Total

MERCHANDISE TOTAL
Total Monogramming charges ($5.00 per line per item)
Sales tax on merchandise delivered in Colorado and Utah
Regular FedEx shipping $4.50 per U.S. delivery address (Items are sent within 24 hours for delivery in 2 to 4 days) $4.50
Please add $4.50 per each additional U.S. delivery address
FedEx Standard Overnight Service Add $6.00 for delivery in 2 business days to confirm U.S. address*
Any additional freight charges
International Shipping (see shipping information on back)

Method of Payment
Check/Money Order ☐ Gift Certificate(s) ☐ AMOUNT ENCLOSED $ ____
American Express ☐ MasterCard ☐ VISA ☐ Other ☐
Account Number
MO YR
Expiration Date
Signature ___

Fig 7.2: Sample Input – Customer Order form

Designing Databases

Objectives

- Efficient Storage of Data
- Efficient Retrieval of Data
- Efficient Update of Data
- Ensuring Data availability
- Ensuring Data Integrity

Involves

- Structuring the data into stable structures, called normalized tables with minimum of redundancy
- Developing logical database design that reflects the actual data requirements
- Translate logical design into physical design

High Level Design

The High level design is also called the logical design. It is the design of the database schema. It involves identifying the data objects to be entered into the database, grouping them into records and grouping records into sets or tables. This is done through the process of normalization

Among the data items, the key that uniquely defines all the data items is chosen as the primary key. The primary key example

Employee table (Schema)

Primary key: ENo

Eno	Ename	Salary	Grade

Low Level Design

Low level design is also known as the physical design. It describes the actual physical storage / organization of the data in the database i.e. the file structure, indices and the access paths. It entails choosing the appropriate storage format for each element from the logical database model. It should ensure minimum storage space and maximum data quality. The techniques that are used are partitioning, chaining etc. It should specify the amount and type of secondary storage and processing speed required for efficient access and retrieval.

Designing User Interfaces

User Interfaces comprise the passages, messages, prompts and responses used to enable a conversation between system and user. It also determines the amount of information that must pass between system and user. The quality of interface is important as it guides the interaction between system and user. Awkward design impedes system usage. The interface strategy determines what information is entered and how responses are made.

Features of a good interface

- Strives for consistency
- Enables frequent users to use short cuts
- Offers informative feedback
- Designs dialogs to yield closure
- Offers simple error handling
- Permits easy reversal of actions

Interface Types

Menu driven: It presents the system with various alternatives.

- Menu Alternatives – The menu choices are presented in single word in keyword dialogue approach. In some systems pull down menu is used. In pull down menus, the drop down alternatives are displayed, when the keyword is pointed at by using the mouse. Nested menu are used when there are extensive set of alternatives to choose from. A selection from a set of choice leads to a subsequent decision about other alternatives. Decisions are made in top down fashion.

- Menu Interfaces - Menu dialogues can be designed using many interface devices such as keyboard, mouse, light pen, touch screens etc.
- Menu Placement Strategies -The menu of options can be positioned on the display screen in several ways. It can be a single column or double column display as used in transaction processing systems and reporting systems. On the other hand to facilitate retention of information on the display and at the same time to offer menu selections to users - the menu can be shown horizontally at the top or bottom of the screen

Keyword based: User invokes the processing activities by entering a command that the system understands. Three forms of keyword dialogue are

- Single command form- User enters the keyword that the system will associate with specific commands for certain processes
- Mnemonic command form– User enters the abbreviation based on longer phrases that are later correlated with the actual commands
- Natural Language form– The users can instruct the system with less rigid commands. Instead of conventional command syntax, the users can use their own vocabulary of words for different operations. It relies on the artificial intelligence and expert systems technology. The user's instructions in natural language are translated into computer oriented / understandable instructions.

Question Answer based: It relies on the presentation of the question to the user. The answer guides the resultant processing. It has two formats

- Yes / No answers
- Narrative responses: This strategy allows for presentations of more elaborate questions and alternatives than do the other strategies. It requires the analyst to anticipate every possible answer a user may provide for a difficult task. The analyst should take care to minimize the number of words used to speed user-system interaction and to avoid possible errors.

Based on the design specifications, the software and hardware selection and acquisition is made and eventually actual program development takes place.

Testing at Development Stage

No system design is ever perfect. Communications problems, programmer's negligence or time constraints can lead to errors that have to be eliminated before user acceptance testing. A system is tested for online response, volume of transactions, stress, recovery from failure and usability. The testing process involves creating a test plan for testing the key activities of the system created.

Integration testing is testing of all modules together. System testing is testing of the entire system together. The user acceptance testing at the development phase is performed using the simulated data.

Objective

- To confirm that the system satisfies the specified requirements
- To check whether the system produces the correct output

Testing can be are performed using various strategies. Testing can be performed in parallel with coding. The deliverables are test plan, test data and the results of program.

Unit testing

It is the individual program/ module testing with test data. Each unit is checked thoroughly and completely for any errors with the help of test data - both valid and invalid. The test data should be able to test possible maximum or minimum values as well as all variations in format and codes

Integration Testing

Integration or Link testing is performed to ensure that the programs/modules that are independent can actually work together as planned. The system is tested using normal transaction data to ensure that the coupled modules can actually handle the load.

System testing

It is testing of system as a complete entity. It also involves checking of the adequateness of documents - operational manual, user manual etc, ensuring that the manuals communicate effectively how data has to be prepared and also ensuring correctness and understandability of output.

Acceptance testing or Alpha testing

After the information system software is developed completely and before the start of installation, the software has to be tested for user acceptance. This user testing of a completed information system using simulated data is called Alpha Testing.

Summary

- The process of design translates end user performance specifications into design specifications that are in turn the foundation of the actual program
- System design involves specification of output, input, interface and database design
- Alpha testing is testing the system with the simulated data

Test your Understanding

1. What are the objectives of a good input design?

2. What are different types of outputs?

3. What is alpha testing?

Session 10: System Implementation

Learning Objectives

After completing this chapter, you will be able to:

- Manage the implementation of information systems
- Understand the importance of post-implementation review

Implementation

Whether a system is developed, leased, or purchased, successful implementation involves getting the right information to the right people in the right format at the right time. Systems implementation includes hardware acquisition, software acquisition or development, user preparation, hiring and training personnel, site preparation, data preparation, installation, testing, start-up, user acceptance and post-implementation review. Hardware can be purchased, leased or rented. Application service providers can not only provide software, but also the hardware and end user support Coding deliverables are the actual code and program documentation. In other words implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing manual or automated system or it may be a major modification. System developers may choose to pilot (test) the operation in only one area of the firm. Sometimes both old system and the new system are run simultaneously to compare results. In other cases the old system is stopped one day and the next day the new system is in place. Once installed the applications are used for many years. With the change in environment the requirements may change, users may seek modifications and enhancements. Hence the application has to be maintained continually. This is an ongoing process. Evaluation is performed to assess its strength and weaknesses. The actual evaluation may be operational i.e. to assess the functionality of the system namely ease of use, response time, suitability of information formats etc. Evaluation is performed for identification and measurement of benefits to the organization and to assess competitive edge the new system can provide.

Changeover

Changeover or conversion is the process of changing from old system to the new one. There are various ways of conversion or changeover. Depending on the size of the organization and the risk associated, system developers may choose any of the various methods available.

The working version of the system is implemented in one part of the organization to pilot (test) the operation in only one area of the firm. Based on feedback, changes are made and if it succeeds only then changeover should take place in the rest of the organization. It provides experience and live test before implementation. It may give the impression that the old system is unreliable and not error free. The changeover can be parallel i.e. the old system is being operational in tandem with the new system. Both systems are run simultaneously to compare results. This offers greatest security. The old system can take over, if errors are found in the new system or if usage problems occur. The disadvantages are that it doubles the operating cost. Also the new system may not get a fair trial.

In other cases, the old system is stopped one day and the next day the new system is in place. This kind of changeover is known as direct**.** The organization now relies fully on the new system. It forces users to make the new system work. There are immediate benefits from new methods and controls. The disadvantage is that there is no other system to fall back on if difficulties arise with the new system. It requires most careful planning.

The changeover may also take place in a phased manner. The system conversion is gradual across all depts. It allows some users to take advantage early. Also it allows training and installation without unnecessary use of resources. The disadvantage is that this method has a long phase-in period. It can cause user problems irrespective of whether the project is successful or not.

Acceptance Testing by user or Beta Testing

Testing of the software is done in an environment where it will be eventually used**.** This type of testing in which live data is used and organization audit it.

Maintenance

Definition

Maintenance includes all the software engineering activities that occur following delivery of a software product to the customer. The idea behind maintenance is to provide continuity of service, to support mandatory upgrades and to sustain correct operation of the installed system

Maintenance process comprises

- Making enhancements to software products
- Adapting products to new environments
- Correcting problems
- Improving performance of the installed products

Problems associated with maintenance

- Not a priority area for the management
- Management reluctant to spend software is developed and commissioned why we should spend more on it. Another problem could be the poor quality of documentation or no documentation at all. Yet another problem deals with the issue of user satisfaction. An unsatisfied user may have constant demand for enhancements and extensions.

The maintainability of the software is the ease with which the maintenance activity can be carried out. It depends on many attributes. The attributes are mean time to failure which is a measure of how often the system breaks down or the software fail, mean time to effect change i.e. how soon it is possible to make changes, clarity and modularity of the software and good internal / supporting documentation.

Types of Maintenance

Corrective: It deals with fault repair. In other words the modification initiated due to defects. These could be error due to design, logic code or any residual error. Undertaking this type of maintenance leads to increased program complexity. Corrective Maintenance can be avoided by developing higher quality of code, developing better test procedures, developing better documentation, adherence to standards and conventions

Adaptive: It deals with adapting software to changes in environment. These changes could be due to change in business rules, government policies, change in hardware or software. It does not lead to a change in system functionality. It can be avoided by anticipating changes during requirements engineering and design phase.

Perfective: In this type of maintenance changes are made to expand the existing requirements for example functional enhancements to products. It involves providing new functional capabilities, improving user displays and modes of interaction, upgrading external documents and internal documentation, upgrading the performance characteristics of a system. It can be avoided by fine tuning the user needs and by intensive user participation

Preventive: This type of maintenance is used to improve maintainability, to prevent malfunctions it involves making maintainable programs that can be easily corrected, adapted, enhanced, their code re-structured, code optimized and modularity improved.

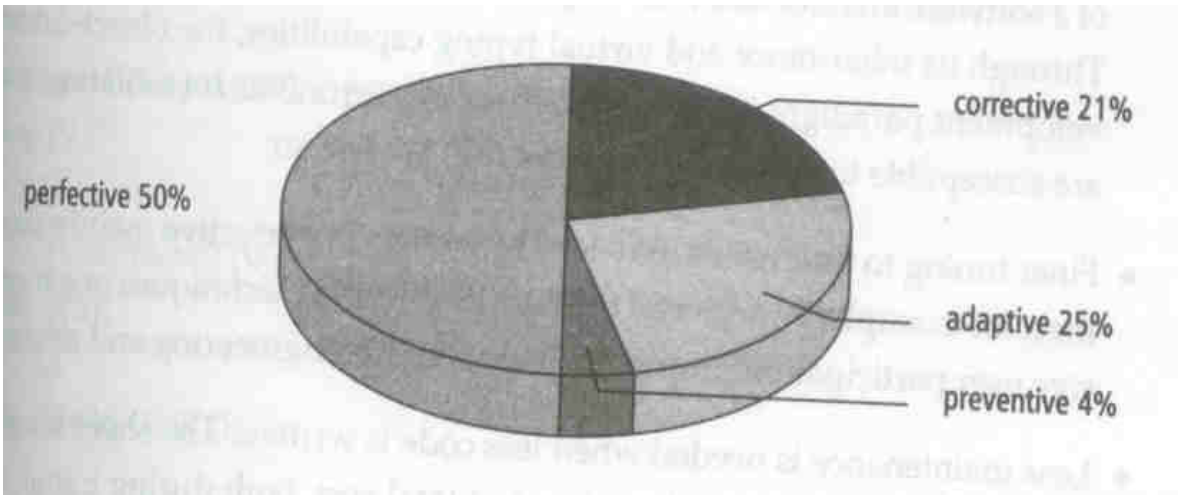


Fig. 10.1: Distribution of Maintenance activities

Review

The success of an MIS is generally measured after the system has been deployed and made operational by performing evaluation to identify the strengths and weaknesses of the system. The evaluation is performed, based on the following dimensions.

Operational evaluation: Detailed assessment of the manner in which the system functions is done. Functionality is assessed based on parameters namely ease of use, response time, suitability of information formats, overall reliability and level of utilization.

Organizational Impact: Evaluation may be for identification and measurement of benefits to the organization in areas such as financial concerns (cost, revenue and profit), operational efficiency and competitive impact. It includes impact on internal and external information flows

User-manager assessment also forms part of evaluation. The attitudes of the senior and user managers within the organization as well as the end user are evaluated. The actual performance is compared with the planned activity in order to determine whether the desired results are achieved or not and to assess whether to term MIS a success or not.

Summary

- System Implementation involves converting physical system specifications into working and reliable software, to provide associated documentation and to provide help for current and future users

Test your Understanding

1. What are the different types of changeover and conversion strategies

2. What are the different aspects of system review?

Glossary

Acceptance testing: The process whereby actual users test a completed information system, the end result of which is the users' acceptance of the system

Adaptive Maintenance: Changes made to a system to evolve its functionality to changing business needs or technologies

Alpha Testing: User testing of a completed information system using simulated data

Analysis: The process of study of the current system is studied and proposal of alternative replacement systems

Beta Testing: User testing of a completed information system using real data in the real user environment

Business Process re-engineering: The search for, and implementation of, radical change in business processes to achieve breakthrough improvements in products and services

Corrective Maintenance: Changes made to a system to repair flaws in its design, coding, or implementation

Computer Aided Software Engineering CASE: Software tools that provide automated support for some portion of the systems development process.

Context Diagram: An overview of an organizational system that shows the system boundary, external entities that interact with the system and the major information flows between the entities and the system

Conversion: The organizational process of changing over from the current information system to a new one

Data: Data are raw facts that collected for the entity in question

Database Management Systems: Software that is used to create, maintain, and provide controlled access to user databases

Data Flow Diagrams (DFD): Graphical depiction of data processes, data flows and data stores in a business system

Data Store: Data at rest, which may take the form of many different physical representations

Decision Support System (DSS): An interactive information system that supports the decision making process through the presentation of information designed specifically for decision maker's problem approach and application needs

Direct Conversion: Changing over from the old information system to a new one by turning off the old system when the new one is turned on.

Economic Feasibility: A process of identifying the financial benefits and costs associated with a development project

Gap analysis: The process of discovering discrepancies between two or more sets of data flow diagrams or discrepancies within a single DFD

Implementation: The phase of system development where the information system is coded, tested, installed, and supported in the organization

Information: Information is processed data. Information attributes a meaning to the data

Management Information System (MIS): A computer based system composed of people, hardware, software and procedures that share a common database to help users interpret data and apply to business

Maintainability: The ease with which software can be understood, corrected, adapted, and enhanced

Maintenance: Maintenance include all the software engineering activities that occur following delivery of a software product to the customer

Mean Time Between Failures MTBF: A measurement of error occurrences that can be tracked over time to indicate the quality of a system

Operational Feasibility: The process of assessing the degree to which a proposed system solves business problems or takes advantage of business opportunities

Parallel Conversion: Running the old information system and the new one at the same time until management decides the old system can be turned off

Perfective Maintenance: Changes made to a system to add new features or to improve performance

Preventive Maintenance: Changes made to a system to avoid possible future problems

Phased Conversion: Changing from the old information system to the new one incrementally, starting with one or a few functional components and then gradually extending the installation to cover the whole new system

Pilot Conversion: Trying out a new information system at one site and using the experience to decide if and how the new system should be deployed throughout the organization

Prototyping: An iterative process of systems development in which requirements are converted to a working system which is continually revised through close work between an analyst and users

Schema: The design of a database is called as schema

Systems Analyst: The organizational role most responsible for the analysis and design of information systems

Systems Analysis and Design; The complex organizational process whereby computer-based information systems are developed and maintained

Systems design: Phase of the SDLC in which the system chosen for development in systems analysis is first described independent of any computer platform (logical design) and is then transformed into technology-specific details (physical design) from which all programming and system construction can be accomplished

System Documentation: Detailed information about a system - design specifications, its internal workings, and its functionality

Transactions: Individual, simple events in the life of an organization that contain data about organizational activity

Transaction Processing System (TPS) A computerized information system developed to process large amounts of data for routine business transactions such as payroll and inventory

References

Books

- Systems Analysis and Design, Elias M Awad, Galgotia
- Analysis and Design of Information Systems, James A Senn, McGraw Hill International
- Analysis and Design of Information Systems, V Rajaraman, PHI 2002
- Modern Systems Analysis and Design, Jeffrey A Hoffer, Joey F George, Joseph S. Valacich, Pearson Education
- Systems Analysis and Design, Kendall and Kendall, PHI
- Systems Analysis and Design, Igor Hawryszkiewycz, PHI
- Davenport, Thomas (1993), Process Innovation: Reengineering work through information technology, Harvard Business School Press, Boston

STUDENT NOTES: