SYSTEMS SHORTNOTE

System

orderly grouping of interrelated and interdependent components linked together according to a plan to achieve a specific objective.

Types of Systems

1. Open System

- Take inputs from the environment.
- Gives output to the environment.

2. Closed system

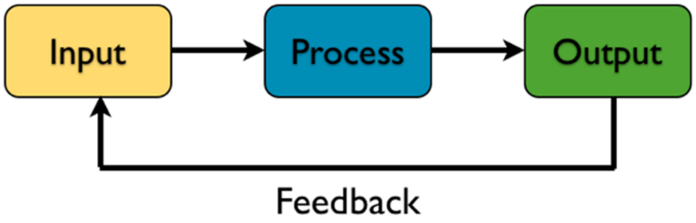
- It doesn't interact with its environment.
- It doesn't take input or give output to the environment.

Transportation System of a Country ← **Manmade System**
Human Blood Circulation System ← **Natural System**

Solar System ← **Nonliving System**
Human Respiratory System ← **Living System**

What is an Information system?

an integrated set of components for Collecting, Storing, Processing and Communicating Information.



Types of Information System

Transaction Processing Systems (TPS)

used to perform and record the daily routine transactions necessary to conduct a business and serve the operational level users of an organization. Used to make structured decisions

Eg: payroll, Retail point of sale system

Management Information Systems (MIS)

serve the functions of planning, controlling, and decision making by providing routine summary and exception reports for the management level users of an organization.

Eg: Inventory control System

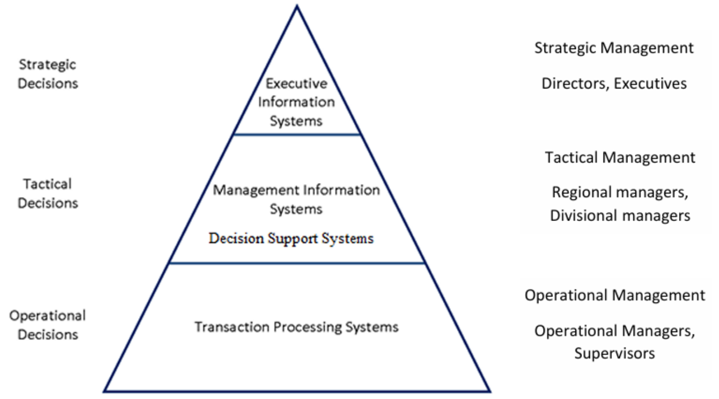
Decision Support Systems (DSS)

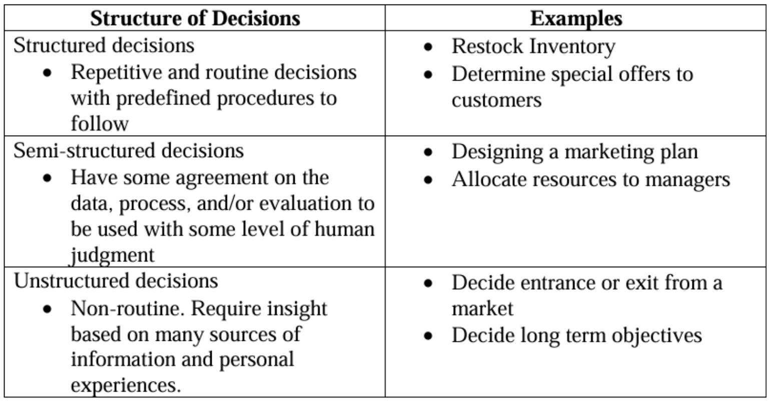
combine data and sophisticated analytical models or data analysis tools to support semi-structured and unstructured decision making for the management level users

Eg: Market analysis systems

Executive Support Systems (ESS)

Address unstructured decision making through advanced graphics and communications for the strategic level users of an organization. **Eg**: Financial Planning Systems.





Structure of Decisions	Examples
Structured decisions • Repetitive and routine decisions with predefined procedures to follow	• Restock Inventory • Determine special offers to customers
Semi-structured decisions • Have some agreement on the data, process, and/or evaluation to be used with some level of human judgment	• Designing a marketing plan • Allocate resources to managers
Unstructured decisions • Non-routine. Require insight based on many sources of information and personal experiences.	• Decide entrance or exit from a market • Decide long term objectives

Office Automation Systems (OAS)

designed to increase the productivity of employees through automating information gathering, communication, and presentation processes. **Eg**: Work scheduling systems

Geographic Information Systems (GIS)

connect data with geography. GIS allow to map, model and analyze large quantities of data within a single database according to their location. **Eg**: Web based map services, Urban Planning and Transportation Planning applications

Knowledge Management Systems (KMS)

comprise a range of practices used in an organization for acquiring, creating, storing, distributing, applying, integrating knowledge.

Eg: Knowledge work systems (KWS), Customer Feedback systems

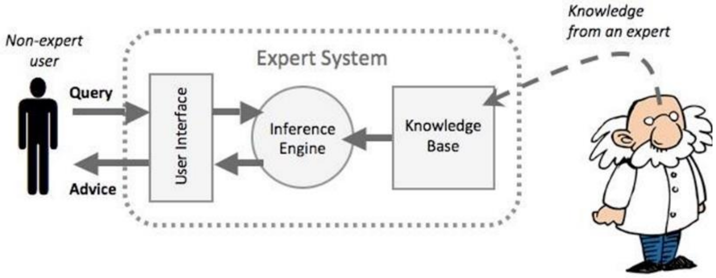
Content Management Systems (CMS)

support the creation and modification of digital content. It supports to multiple users working in a collaborative environment.

Eg: web-based publishing(WordPress), format management, history editing and version control.

Enterprise Resource Planning (ERP)
allow organizations to use integrated applications to manage the businesses. ERP software integrates product planning, development, manufacturing, sales and marketing. **Eg**: SAP ERP

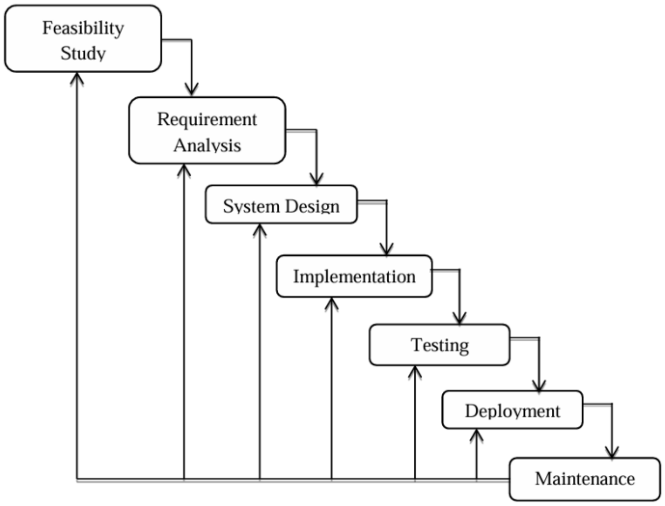
Expert Systems(ES)
emulate the decision-making ability of a human expert. To design an expert system, one needs a knowledge engineer, an individual who studies how human experts make decisions and translates the rules into terms that a computer can understand. **Eg:** MYCIN



Smart Systems
make decisions based on the available data in a predictive or adaptive manner, by means of sensing, actuating and controlling. **Eg**: Anti-lock Braking System (ABS)

System Development Lifecycle models

Waterfall Model

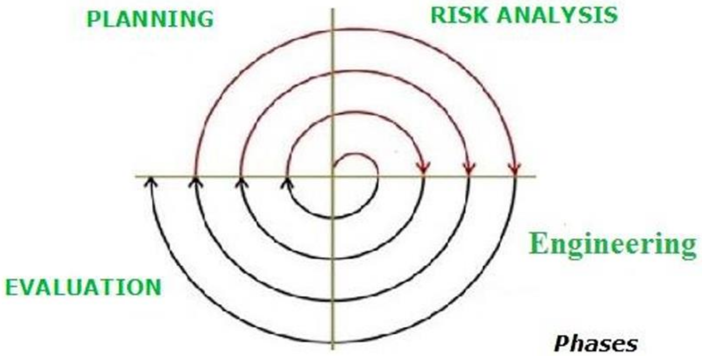


More suitable – when func. Requirements clear and stable, To develop short term simple systems,

Advantages - Easy to manage and control, Every phase has a defined start and end point, Phases are processed and completed one at a time, Works well for smaller projects where requirements are very well understood and stable.

Disadvantages - difficult to respond to changing customer Requirements, No working software is produced until late during the life cycle, High amounts of risk and uncertainty, Poor model for complex and ongoing projects.

Spiral Model



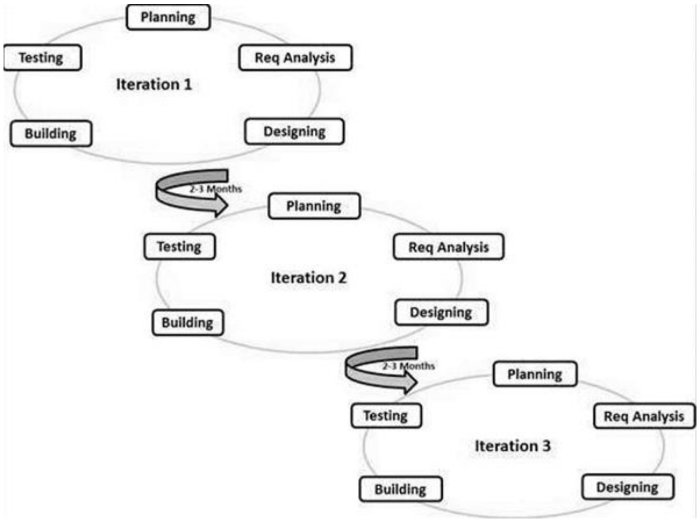
A combination of iterative development and sequential linear development processes.

Applicability – For systems need continuous risk evaluation, for expensive complicated projects, projects with continuous changes

Advantages - Fast development, more features can be added during the process, proper risk evaluation is involved

Disadvantages - Not suitable for small projects, spiral may go infinitely, cant understand the final product in the beginning, Management being complex, Costly

Agile Model



Applicability – When frequent changes are required, if continuous customer attraction is required, uses the existing methods and modifies them to suit the project requirements.

Advantages – Improved quality, Better control, high consumer satisfaction, reduced risk, Increased flexibility

Disadvantages – Can't predict effort in the beginning, difficult to measure the progress, difficult to implement, may easily fall out of track

Prototype Model

- A model is created before developed for demonstration.

Models – Throw-away Model, Evolutionary Model, Incremental Model

Advantages- active user involvement, better understanding About the system in the beginning, Quicker user feedbacks

Disadvantages- risks due to lesser req analysis, confusions may occur between prototypes and final products.

Rapid Application Development (RAD)

- Minimal planning & more focused on development

Phases

1. Business Modelling
2. Data Modelling
3. Process Modelling
4. Application Generation
5. Testing & Turnover

Advantages - Reduced development time, Encourages customer feedback, suitable for Changing requirements. Integration from very beginning solves a lot of integration issues.

Disadvantages -Only modularized systems can be built using RAD, Requires highly skilled developers/designers for modeling, High dependency on modeling skills.

System development methodologies

i. Structured methodology - Provides a framework (structure) with a set of well-defined guidelines through steps of tasks

ii. Object Oriented methodology - Models system as a collection of objects that work interactively.

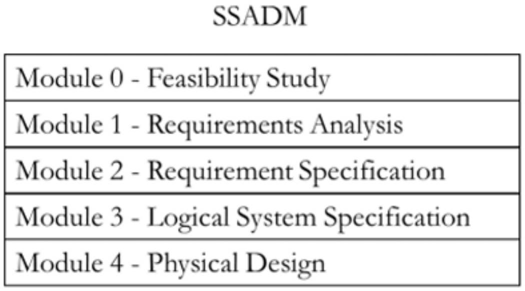
Structured System Analysis & Design Methodology (SSADM)

involves study the present system and sketches a blueprint to develop a new system or to modify the existing system.

Parts of a SSADM - Modules, Stages, Steps, Tasks

Modules of SSADM

- **Feasibility Study** - The business area is analyzed.
- **Requirements Analysis -** The requirements of the system to be developed are identified.
- **Requirements Specification** - Detailed func & non-func requirements are defined and new techniques are introduced to define the required processing and data storage.
- **Logical System Specification** - System to be developed is specified logically
- **Physical Design** - Logical design is transformed into a physical design taking technical constraints into consideration.



SSADM

Module 0 - Feasibility Study
Module 1 - Requirements Analysis
Module 2 - Requirement Specification
Module 3 - Logical System Specification
Module 4 - Physical Design

Advantages - More focus on analysis and design, Better quality system specification and documentation, Effective communication and user involvement, Flexible continuity when staff changes, Improved management control

---

Preliminary Investigation - focuses on recognizing the need for a new system and reaching a clear initial picture of what the physical system actually is.

Preliminary investigation is done in two phases.

1. Problem definition 2. Feasibility study

Problem definition - A preliminary survey of the system is carried out to identify the scope of the system.

Feasibility study - means whether the development of a new or improved system is practical and beneficial.

i. Technical feasibility - evaluates whether the developers have ability to construct the proposed system. whether the technology needed exists? how difficult system will be to develop? whether the developers have enough experience using that technology?

---

ii. Economic feasibility - studies cost and benefits to evaluate whether the benefits justify the investments in the system development.

---

iii. Operational feasibility - assesses the willingness and ability of the users to support and use the proposed system. Will the system be useful?

---

iv. Organization feasibility - determines the extent to which the proposed system supports the objectives of the organization's strategy

- End of the feasibility study → feasibility report is produced

---

Requirement analysis - process of studying and analyzing the user needs to arrive at a definition of the problem domain and system requirements. The main objective is to discover the boundaries of the new system & how system must interact within the new problem domain.

- Methods to identify requirements – Interviews, surveys, Questioners, observations, prototyping, sample document collection

Functional requirements: What activities that the system should carry out.(functionality/behavior of the system)
Non-Functional requirements: How well or with in what limits requirement should be satisfied

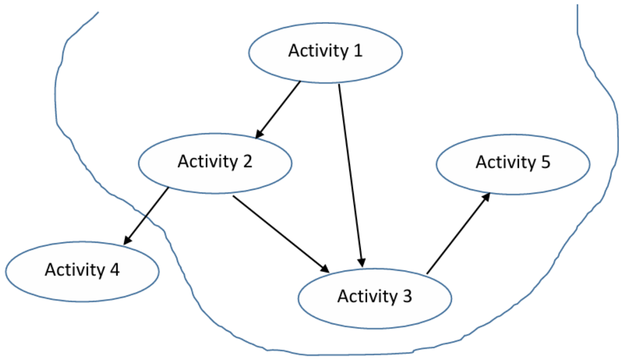
Func. Requirements	Non-Func. Requirements
Mandatory	Not mandatory
On user requirements	On user expectations
What product does	How product works
Product features	Product properties

Essential requirements are defined with **"Shall"** whereas nice to have requirements are defined with **"Should"**

Analytical Tools

- **Business Activity Modeling -** A start-off technique for understanding what's going on in a system. It is used to show the business activities that the acts in the system's environment.

Steps – represent activities, link them in flow, mention the system boundaries



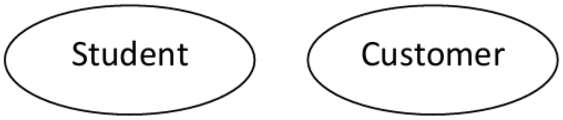
- **Data Flow Modeling (DFM)** - used to model data processing In the system. It is used to define partitions into sub systems. DFM consists of a set of Data Flow Diagrams (DFD).

3 Levels of DFD

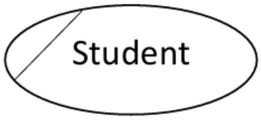
- DFD0 – Context Diagrams
- DFD1 – Level 1 DFD
- DFD2 – Level 2 DFD

Components in DFD

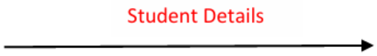
- External Entities



Representing duplicates of external entities

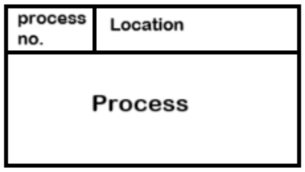


- Data Flows



* data flows between two external entities are shown by **dashed arrows** (------------------->)

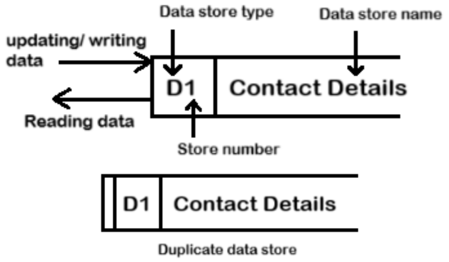
- Processes



- Data Stores

Four types:

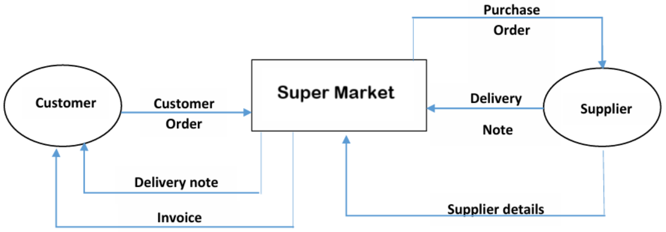
- Computerized (D)
- Temporary data (T) – Ex. temporary program file
- Manual (M) – Ex. filing cabinet
- Manual Temporary T(M) – Ex.in-tray, mail box



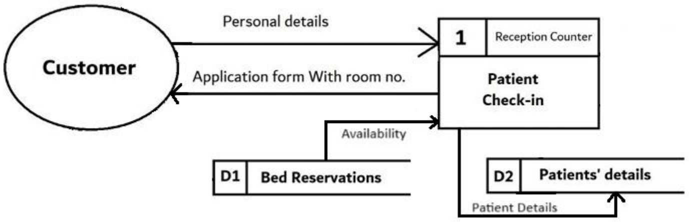
Data flow modelling Rules

- Data can flow directly between:
  - ✓ Two external entities
  - ✓ An external entity and a process
  - ✓ Two processes
  - ✓ A Process and a data store
- A direct data flow cannot exist between:
  - × An external entity and a data store
  - × Two data stores
- Inputs to Data Stores come only from Processes
- Outputs from Data Stores go only to Processes

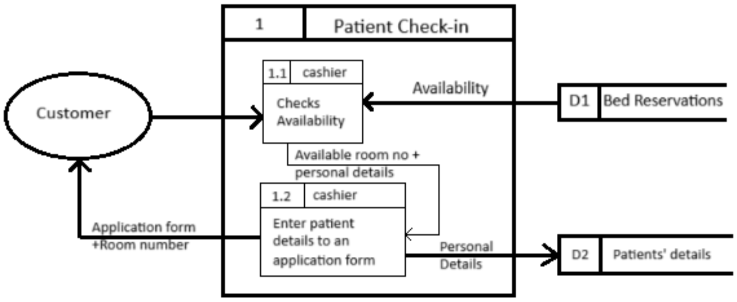
Context Diagram -The highest level of abstraction of the system.



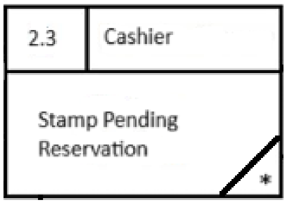
Level 1 DFD – provides a higher-level overview of the system's data processing, shows data movements among the major components of the system.



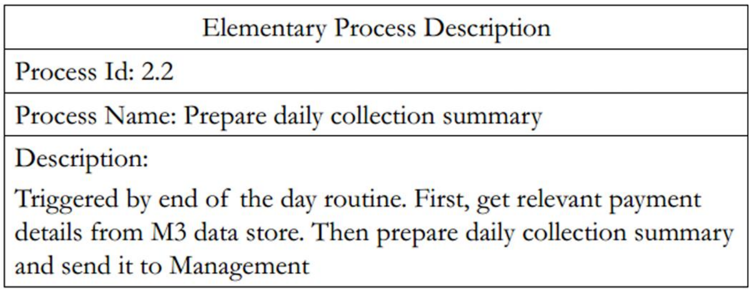
Level 2 DFD - Further decompose Processes into each and every task.



Elementary Process - A process that doesn't need any further Decomposition, Associates a textual description known as Elementary Process Description (EPD) Mentioned an asterisk (*) At the right below end of a process.

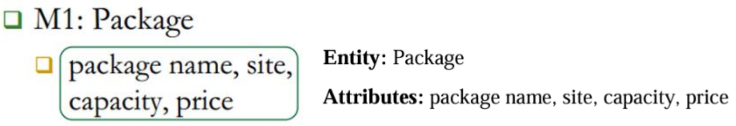


Elementary Process Description - Contains enough details for program specification, Written in plain English or pseudo code



Elementary Process Description
Process Id: 2.2
Process Name: Prepare daily collection summary
Description: Triggered by end of the day routine. First, get relevant payment details from M3 data store. Then prepare daily collection summary and send it to Management

Logical Data Modeling (LDM) models the systems data processed by the processes identified in DFM. It consists of a diagram called Logical Data Structure (LDS) and associated textual descriptions.



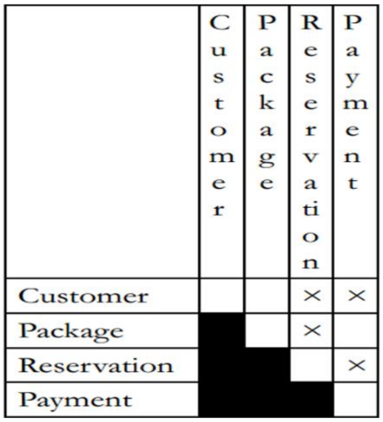
❑ M1: Package

❑ package name, site, capacity, price

Entity: Package

Attributes: package name, site, capacity, price

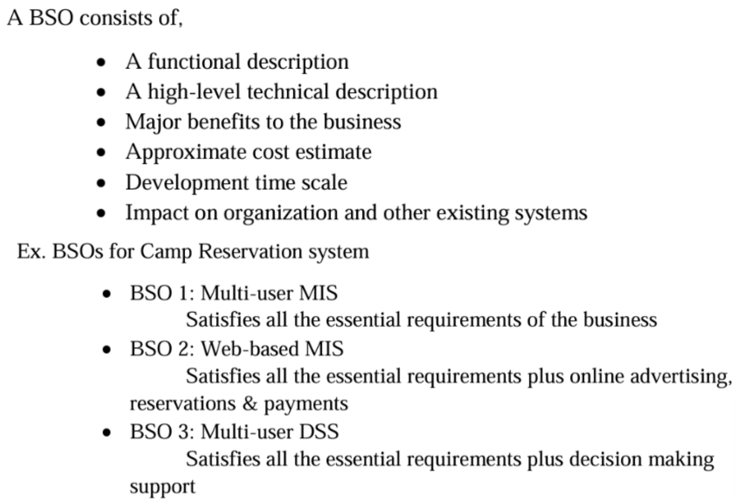
Entity Matrix



	Customer	Package	Reservation	Payment
Customer			×	×
Package	■		×	
Reservation	■	■		×
Payment	■	■	■	

- Helps to identify the relationships among entities.
- Provides all possible pairings of entities to identify relationships among them in the interested system.
- Associations between each pair of entities are checked for once.

Business System Options (BSO) - help to select the most appropriate option that satisfies the business requirements.



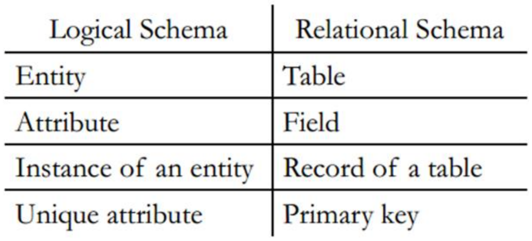
A BSO consists of,

- A functional description
- A high-level technical description
- Major benefits to the business
- Approximate cost estimate
- Development time scale
- Impact on organization and other existing systems

Ex. BSOs for Camp Reservation system

- BSO 1: Multi-user MIS
  Satisfies all the essential requirements of the business
- BSO 2: Web-based MIS
  Satisfies all the essential requirements plus online advertising, reservations & payments
- BSO 3: Multi-user DSS
  Satisfies all the essential requirements plus decision making support

Logical schema Vs Relational schema



Logical Schema	Relational Schema
Entity	Table
Attribute	Field
Instance of an entity	Record of a table
Unique attribute	Primary key

Data Dictionary - an integral part of database, Holds information about the database and the data that it stores (data about data - metadata)

Testing

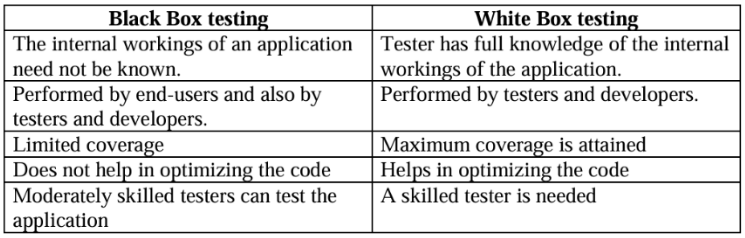
Test Cases - set of actions executed to verify a particular feature or functionality of a software application.

A test case consists of – test ID, tester name, steps, expected results, actual results etc..

Software testing techniques

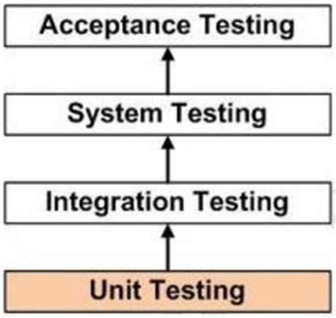
1. Black Box testing - behavior is examined by studying the inputs and outputs. No need technical knowledge. Used in system testing and user acceptance testing

2. White Box testing - detailed investigation of internal logic and structure of the code. Requires technical knowledge and used in unit testing and integration testing.



Black Box testing	White Box testing
The internal workings of an application need not be known.	Tester has full knowledge of the internal workings of the application.
Performed by end-users and also by testers and developers.	Performed by testers and developers.
Limited coverage	Maximum coverage is attained
Does not help in optimizing the code	Helps in optimizing the code
Moderately skilled testers can test the application	A skilled tester is needed

Software testing types



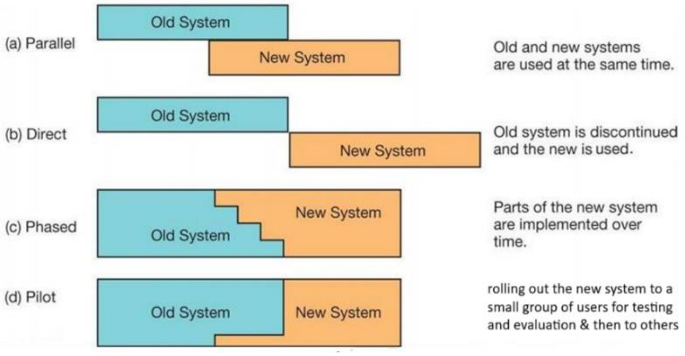
- Testing of individual program units **(Unit Testing)**

•Continues with the testing of the integration of these units **(Integration Testing)**

•testing of the system's functionality as a whole **(System Testing)**

•testing to see whether the system is acceptable to the users **(Acceptance Testing)**

Deployment Methods



Old and new systems are used at the same time.

Old system is discontinued and the new is used.

Parts of the new system are implemented over time.

rolling out the new system to a small group of users for testing and evaluation & then to others

Maintenance

•changes to hardware, software, and documentation to Support its operational effectiveness.

•Making changes to improve a system's performance, correct errors, deal with security issues, or address new user requirements.

Commercial-Off-The-Shelf (COTS) System

ready-made softwares and are readily available for purchase.

Advantages – Can be implemented in significantly less time. Less resources are required (labor, space & money). more configurable than custom developed software. Availability of wide range of alternatives in the market. More tested by users.

Disadvantages - can be highly complex and usually include many features that will be never used.

- Difficult to gain any competitive advantage from using COTS system as the competitors can also buy and use the same COTS.
- May not meet the expected organizational requirements
- Less contact and support by the developers

Custom-developed Softwares

designed specifically to the requirements and built to operate exactly as needed.

Advantages - Satisfy unique requirements, Can gain real competitive advantage, Can be modified, Can be integrated with existing systems of future systems, Provides all needed functions and none of not needed.

Disadvantages – Large initial Investment, Takes longer time to implement