**SYSTEM DESIGN**

**System Design** is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements

Basic understanding of system characteristics, system design, and its development processes. It is a good introductory guide that provides an overview of all the concepts necessary to build a system.

**System development** is systematic process which includes phases such as planning, analysis, design, deployment, and maintenance.

* System analysis
* System design

**System Analysis**

It is the process of gathering and interpreting data, finding faults, and breaking down a system into its constituent parts.

* A system analysis is carried out to investigate a system or its components in order to determine its goals.
* It is a problem-solving strategy that enhances the system and guarantees that all of the system's components work together to achieve their goals.

**System Design**

It is the process of specifying the components or modules of a new business system or replacing an existing system in order to meet certain needs.

Before you begin planning, you must first completely grasp the old system and identify how computers may be employed most effectively in order to run efficiently.

The goal of system design is to figure out how to achieve the system's goal.

The main focus of System Analysis and Design (SAD) is on

* Systems
* Processes
* Technology

**SYSTEM**

A system is an organized grouping of interdependent components linked together according to a plan to achieve a certain objective," according to the definition.

**Three fundamental limitation in system.**

* A system must have some structure and behavior that is intended to accomplish a specific goal.
* The system's components must be interconnected and interdependent.
* The organization's aims take precedence above the objectives of its subsystems.

**Properties of a System**

**Organization**

It is the arrangement of components that aids in the accomplishment of predetermined goals.

**Interaction**

It is determined by how the components interact with one another.

In a company, for example, the purchasing department must work with the manufacturing department and the payroll department with the personnel department.

**Interdependence**

The interdependence of a system's components refers to how they rely on one another. The components are coordinated and linked together according to a predetermined strategy for proper operation. The output of one subsystem is required as input by another subsystem.

**Integration**

The way a system's components are connected is referred to as integration. Even if each portion serves a separate function, the parts of the system work together within the system.

**System Design goals**

The system's goal must be in the center. It could be true or false. It's not uncommon for a company to declare one goal and then work toward another.

For a successful design and conversion, users must understand the fundamental goal of a computer program early in the analysis.

**Element (component) of systems**

Input/ Output

* A system's main goal is to create an output that is beneficial to its user.
* The information that enters the system for processing is referred to as inputs.
* The result of processing is output.

Processors

* The processor is the component of a system responsible for converting input into output.
* It is a system's operating component. Depending on the output requirements, processors may modify the input completely or partially.
* The processing changes when the output specifications vary. In other circumstances, the processor's input is also changed to allow it to handle the transformation.

Control

* The system is guided by the control element.
* It is the subsystem that makes decisions about the pattern of activities that govern input, processing, and output.
* The Operating System and software are in charge of a computer's behavior. Output Specifications establish what and how much input is required to maintain the system balanced.

**Software Design Levels**

**Logical Design**

Logical design refers to an abstract representation of a system's data flow, inputs, and outputs. It describes the inputs (sources), outputs (destinations), databases (data storage), and procedures (data flows) in a user-friendly format.

The system analyst specifies the user demands at a degree of detail that virtually determines the information flow into and out of the system, as well as the required data sources, while developing the logical architecture of a system. E-R diagram modeling and data flow diagrams are employed.

**Physical Design**

Physical design refers to the system's real input and output operations. It focuses on how data is entered into a system, validated, processed, and outputted.

It creates the working system by creating the design specification, which details the candidate system's functionality. It is involved with the design of user interfaces, processes, and data.

It is made up of the following steps:

* Choosing input and output media, creating a database, and deciding on backup strategies.
* Organizing the implementation of the system.
* Creating a test and implementation strategy, as well as identifying any additional hardware or software requirements.
* Costs, advantages, and conversions are all being updated.

**Architectural Design**

It's also known as high-level design, and it focuses on system architecture design. It explains the system's structure and behavior.

It identifies the software as a system with many components interacting with each other.

It establishes the structure of the system development process and the relationships between the various modules.

**Detailed Planning**

It adheres to architectural principles and concentrates on the development of each module.

The detailed planning consist of:

**Conceptual Data modeling**

It is a visual representation of organizational data that contains all of the important entities and relationships. System analysts create a conceptual data model for the current system that supports the new system's scope and requirements.

The basic goal of conceptual data modeling is to extract as much meaning as possible from data.

**Entity Relation Model**

The E-R model, which uses special notation to represent as much meaning about data as possible, is used by the majority of organizations today for conceptual data modeling.

Model of Entity Relationships

It is a database design strategy that aids in describing the relationship between distinct organizational components

**Terms used in Entity Relation model.**

**Entity**

It specifies distinct real world items in an application. For example: vendor, item, student, course, teachers, etc.

**Relationship**

They are the meaningful dependencies between entities. For example, vendor supplies items, teacher teaches courses, then supplies and course are relationship.

**Attributes**

It specifies the properties of relationships. For example, vendor code, student name. Symbols used in entity relation model and their respective meanings

**Software Design Strategy (Approaches)**

The following are the approaches used software design

**Top down design strategy**

Top down strategy of software design is a modular design approach use to develop software which takes the whole software system as one entity and then decomposes it to achieve more than one sub-system or component based on some characteristics.

**In this approach**

* The first step involves identification of highest-level module or main module (generalized model) for developing the software
* The main module is divided into several smaller and simpler submodules or segments based on the task performed by each module.
* Each submodule is further subdivided into several submodules of next lower level.
* Process of dividing each module into several submodules continues until the lowest level modules, which cannot be further subdivided, are not identified
* It starts from the top or the highest level module (generalized model) of the system and keeps on defining the more specific part off the system up to the lowest level modules.
* When all components of the system are composed then the whole system comes into existence.
* Top-down design is more suitable when the software solution needs to be designed from scratch and specific details are unknown.

**Bottom-up strategy**

Bottom-up strategy follows the modular approach to develop the design of the system. Its starts from the bottom or the most basic level modules and proceeds towards the highest level modules by composing higher level of components by using basic or lower level components.

It keeps creating higher level components until the desired system is not evolved as one single component

Bottom-up strategy is more suitable when a system needs to be created from some existing system, where the basic primitives can be used in the newer system.

**In this approach:**

* Modules at the most basic or the lowest level are identified.
* These modules are then grouped together based on the function performed by each module to form the next higher-level modules.
* Then, these modules are further combined to form the next higher-level modules.
* This process of grouping several simpler modules to form higher level modules continues until the main module of system development process is achieved.

**NB:** Both, top-down and bottom-up approaches are not practical individually. Instead, a good combination of both is used

**Structured Design**

Structured design is a conceptualization of problem into several well-organized elements of solution. It is basically concerned with the solution design.

**Advantages of Structured design**

* It gives better understanding of how the problem is being solved.
* Structured design also makes it simpler for designer to concentrate on the problem more accurately.
* Structured design is mostly based on ‘divide and conquer’ strategy where a problem is broken into several small problems and each small problem is individually solved until the whole problem is solved.
* The small pieces of problem are solved by means of solution modules.
* Structured design emphasis that these modules be well organized in order to achieve precise solution.

NB: A good structured design has high cohesion and lowcoupling arrangements

Cohesion is the grouping of all functionally related elements while coupling is

communication between different modules.

**Coupling**

Coupling is a measure that defines the level of inter-dependability among modules of a program. It tells at what level the modules interfere and interact with each other. The lower the coupling, the better the program

**Cohesion**

Cohesion is a measure that defines the degree of intra-dependability within elements of a module.

**Software Analysis and Design Tools**

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code

**Software analysis tools includes:**

**Data flow diagram (DFD)**

Data flow diagram is a graphical representation of data flow in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

**NB:** Difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements

**Components of Data Flow Diagram (DFD)**

**Entities**

Entities are source and destination of information data. Entities are represented by rectangles with their respective names.

**Process**

Activities and action taken on the data are represented by Circle or Round-edged rectangles.

**Data Storage**

Data storage can be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.

**Data Flow**

Data flow is the movement of data indicated by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

**Data Dictionary**

A data dictionary lists all data items appearing in the DFD model of a system.

The data items listed include all data flows and the contents of all data stores appearing on the DFDs in the DFD model of a system.

A data dictionary lists the purpose of all data items and the definition of all composite data items in terms of their component data items

**Context Diagram**

* The context diagram is the most abstract data flow representation of a system. It represents the entire system as a single bubble.
* This bubble is labeled according to the main function of the system.
* The various external entities with which the system interacts and the data flow occurring between the system and the external entities are also represented.
* The data input to the system and the data output from the system are represented as incoming and outgoing arrows