**UNIT – 3 Developing Proposal / System design**

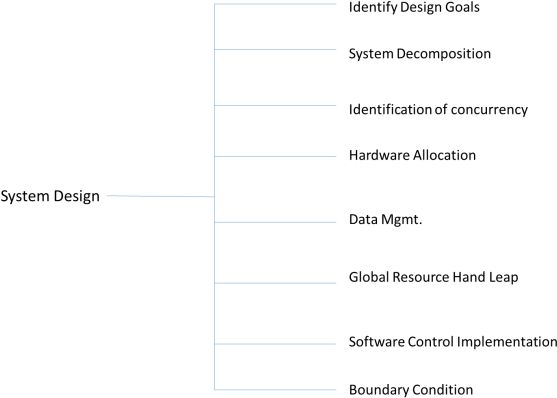
Systems Analysis and Design

**5. System Design**

**System design** is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. “*how to implement*?”

It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.

In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.



# Inputs to System Design

System design takes the following inputs:

* Statement of work
* Requirement determination plan
* Current situation analysis
* Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

# Outputs for System Design

System design gives the following outputs:

* Infrastructure and organizational changes for the proposed system.
* A data schema, often a relational schema.
* Metadata to define the tables/files and columns/data-items.
* A function hierarchy diagram or web page map that graphically describes the program structure.
* Actual or pseudocode for each module in the program.
* A prototype for the proposed system.

# Types of System Design

### Logical Design

Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system. It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

While preparing the logical design of a system, the system analyst specifies the user needs at level of detail that virtually determines the information flow into and out of the system and the required data sources. Data flow diagram, E-R diagram modeling are used.

### Physical Design

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

It consists of the following steps:

* Specifying the input/output media, designing the database, and specifying backup procedures.
* Planning system implementation.
* Devising a test and implementation plan, and specifying any new hardware and software.
* Updating costs, benefits, conversion dates, and system constraints.

### Architectural Design

It is also known as high level design that focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process.

### Detailed Design

It follows Architectural design and focuses on development of each module.

### Conceptual Data Modeling

It is representation of organizational data which includes all the major entities and relationship. System analysts develop a conceptual data model for the current system that supports the scope and requirement for the proposed system.

The main aim of conceptual data modeling is to capture as much meaning of data as possible. Most organization today use conceptual data modeling using E-R model which uses special notation to represent as much meaning about data as possible.

# File Organization

It describes how records are stored within a file. There are four file organization methods:

* **Serial:** Records are stored in chronological order (in order as they are input or occur). **Examples:** Recording of telephone charges, ATM transactions, Telephone queues
* **Sequential:** Records are stored in order based on a key field which contains a value that uniquely identifies a record. **Examples**: Phone directories
* **Direct (relative):** Each record is stored based on a physical address or location on the device. Address is calculated from the value stored in the record’s key field. Randomizing routine or hashing algorithm does the conversion
* **Indexed:** Records can be processed both sequentially and non-sequentially using indexes.

### Comparison

# File Access

One can access a file using either Sequential Access or Random Access. File Access methods allow computer programs read or write records in a file.

### Sequential Access

Every record on the file is processed starting with the first record until End of File (EOF) is reached. It is efficient when a large number of the records on the file need to be accessed at any given time. Data stored on a tape (sequential access) can be accessed only sequentially

## Direct (Random) Access

Records are located by knowing their physical locations or addresses on the device rather than their positions relative to other records. Data stored on a CD device (direct-access) can be accessed either sequentially or randomly.

### Types of Files used in an Organization System

Following are the types of files used in an organization system:

* **Master file**: It contains the current information for a system. For example, customer file, student file, telephone directory.
* **Table file:** It is a type of master file that changes infrequently and stored in a tabular format. For example, storing Zipcode.
* **Transaction file:** It contains the day-to-day information generated from business activities. It is used to update or process the master file. For example, Addresses of the employees.
* **Temporary file:** It is created and used whenever needed by a system.
* **Mirror files:** They are the exact duplicates of other files. Help minimize the risk of downtime in cases when the original becomes unusable. They must be modified each time the original file is changed.
* **Log files:** They contain copies of master and transaction records in order to chronicle any changes that are made to the master file. It facilitates auditing and provides mechanism for recovery in case of system failure.
* **Archive file:** Backup files that contain historical versions of other files.

# Documentation Control

Documentation is a process of recording the information for any reference or operational purpose. It helps users, managers, and IT staff, who require it. It is important that prepared document must be updated on regular basis to trace the progress of the system easily.

After the implementation of system if the system is working improperly, then documentation helps the administrator to understand the flow of data in the system to correct the flaws and get the system working.

Programmers or systems analysts usually create program and system documentation. Systems analysts usually are responsible for preparing documentation to help users learn the system. In large companies, a technical support team that includes technical writers might assist in the preparation of user documentation and training materials.

### Advantages

* It can reduce system downtime, cut costs, and speed up maintenance tasks.
* It provides the clear description of formal flow of present system and helps to understand the type of input data and how the output can be produced.
* It provides effective and efficient way of communication between technical and nontechnical users about system.
* It facilitates the training of new user so that he can easily understand the flow of system.
* It helps the user to solve the problems such as troubleshooting and helps the manager to take better final decisions of the organization system.
* It provides better control to the internal or external working of the system.

# Types of Documentations

When it comes to System Design, there are following four main documentations:

* Program documentation
* System documentation
* Operations documentation
* User documentation

### Program Documentation

* It describes inputs, outputs, and processing logic for all the program modules.
* The program documentation process starts in the system analysis phase and continues during implementation.
* This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

### Operations Documentation

Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.

It includes the following information:

* Program, systems analyst, programmer, and system identification.
* Scheduling information for printed output, such as report, execution frequency, and deadlines.
* Input files, their source, output files, and their destinations.
* E-mail and report distribution lists.
* Special forms required, including online forms.
* Error and informational messages to operators and restart procedures.
* Special instructions, such as security requirements.

# User Documentation

It includes instructions and information to the users who will interact with the system. For example, user manuals, help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

The users, system owners, analysts, and programmers, all put combined efforts to develop a user’s guide.

A user documentation should include:

* A system overview that clearly describes all major system features, capabilities, and limitations.
* Description of source document content, preparation, processing, and, samples.
* Overview of menu and data entry screen options, contents, and processing instructions.
* Examples of reports that are produced regularly or available at the user’s request, including samples.
* Security and audit trail information.
* Explanation of responsibility for specific input, output, or processing requirements.
* Procedures for requesting changes and reporting problems.
* Examples of exceptions and error situations.
* Frequently asked questions (FAQs).
* Explanation of how to get help and procedures for updating the user manual.

# System Documentation

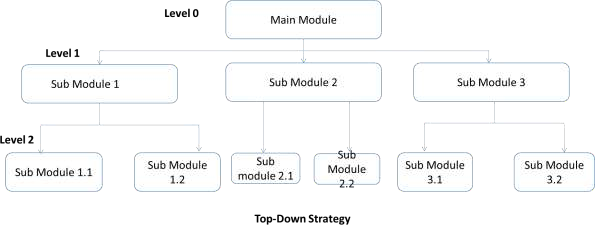
System documentation serves as the technical specifications for the IS and how the objectives of the IS are accomplished. Users, managers and IS owners need never reference system documentation. System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.

* It describes each program within the IS and the entire IS itself.
* It describes the system’s functions, the way they are implemented, each program's purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.
* It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project
* Most of the system documentation is prepared during the system analysis and system design phases.
* During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.

# Top-Down Strategy

The top-down strategy uses the modular approach to develop the design of a system. It is called so because it starts from the top or the highest-level module and moves towards the lowest level modules.

In this technique, the highest-level module or main module for developing the software is identified. The main module is divided into several smaller and simpler submodules or segments based on the task performed by each module. Then, each submodule is further subdivided into several submodules of next lower level. This process of dividing each module into several submodules continues until the lowest level modules, which cannot be further subdivided, are not identified.

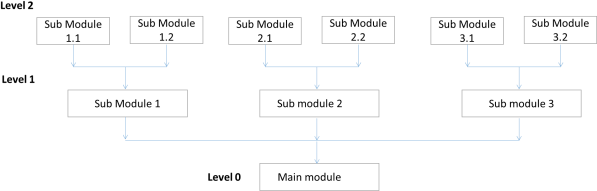


# Bottom-Up Strategy

Bottom-Up Strategy follows the modular approach to develop the design of the system. It is called so because it starts from the bottom or the most basic level modules and moves towards the highest level modules.

In this technique,

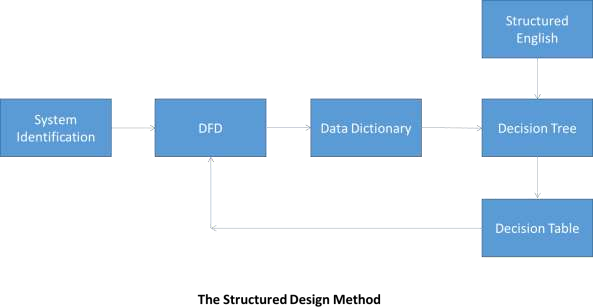
* The modules at the most basic or the lowest level are identified.
* These modules are then grouped together based on the function performed by each module to form the next higher-level modules.
* Then, these modules are further combined to form the next higher-level modules.
* This process of grouping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



# Structured Design

Structured design is a data-flow based methodology that helps in identifying the input and output of the developing system. The main objective of structured design is to minimize the complexity and increase the modularity of a program. Structured design also helps in describing the functional aspects of the system.

In structured designing, the system specifications act as a basis for graphically representing the flow of data and sequence of processes involved in a software development with the help of DFDs. After developing the DFDs for the software system, the next step is to develop the structure chart.



### Modularization

Structured design partitions the program into small and independent modules. These are organized in top down manner with the details shown in bottom.

Thus, structured design uses an approach called Modularization or decomposition to minimize the complexity and to manage the problem by subdividing it into smaller segments.

### Advantages

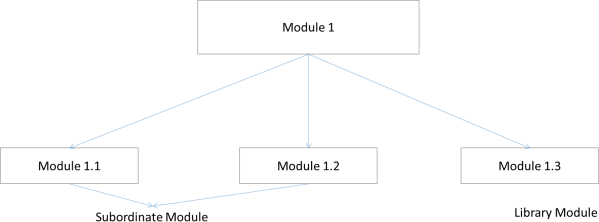
* Critical interfaces are tested first.
* It provide abstraction.
* It allows multiple programmers to work simultaneously.
* It allows code reuse.
* It provides control and improves morale.
* It makes identifying structure easier.

### Structured Charts

Structured charts are a recommended tool for designing a modular, top down systems which define the various modules of system development and the relationship between each module. It shows the system module and their relationship between them.

It consists of diagram consisting of rectangular boxes that represent the modules, connecting arrows, or lines.

* **Control Module:** It is a higher-level module that directs lower-level modules, called **subordinate modules**.
* **Library Module:** It is a reusable module and can be invoked from more than one point in the chart.



We have two different approaches to design a structured chart:

* **Transform-Centered Structured Charts:** They are used when all the transactions follow same path.
* **Transaction–Centered Structured Charts:** They are used when all the transactions do not follow the same path.

### Objectives of Using Structure Flowcharts

* To encourage a top-down design.
* To support the concept of modules and identify the appropriate modules.
* To show the size and complexity of the system.
* To identify the number of readily identifiable functions and modules within each function.
* To depict whether each identifiable function is a manageable entity or should be broken down into smaller components.

# Factors Affecting System Complexity

To develop good quality of system software, it is necessary to develop a good design. Therefore, the main focus on while developing the design of the system is the quality of the software design. A good quality software design is the one, which minimizes the complexity and cost expenditure in software development.

The two important concepts related to the system development that help in determining the complexity of a system are **coupling** and **cohesion**.

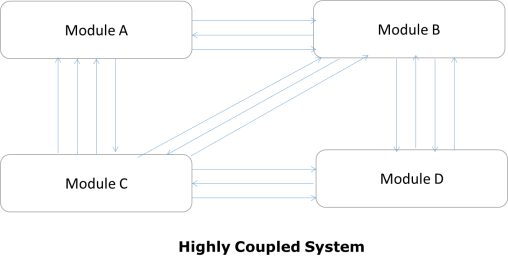
### Coupling

Coupling is the measure of the independence of components. It defines the degree of dependency of each module of system development on the other. In practice, this means the stronger the coupling between the modules in a system, the more difficult it is to implement and maintain the system.

Each module should have simple, clean interface with other modules, and that the minimum number of data elements should be shared between modules.

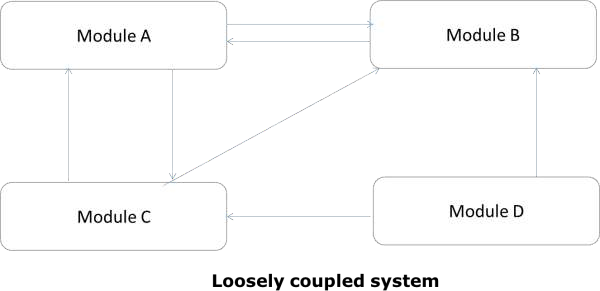
### High Coupling

These type of systems have interconnections with program units dependent on each other. Changes to one subsystem leads to high impact on the other subsystem.



### Low Coupling

These type of systems are made up of components which are independent or almost independent. A change in one subsystem does not affect any other subsystem.



### Coupling Measures

* **Content Coupling:** When one component actually modifies another,then the modified component is completely dependent on modifying one.
* **Common Coupling:** When amount of coupling is reduced somewhat by organizing system design so that data are accessible from a common data store.
* **Control Coupling:** When one component passes parameters to control the activity of another component.
* **Stamp Coupling:** When data structures is used to pass information from one component to another.
* **Data Coupling:** When only data is passed then components are connected by this coupling.

### Cohesion

Cohesion is the measure of closeness of the relationship between its components. It defines the amount of dependency of the components of a module on one another. In practice, this means the systems designer must ensure that:

* They do not split essential processes into fragmented modules
* They do not gather together unrelated processes represented as processes on the DFD into meaningless modules.

The best modules are those that are functionally cohesive. The worst modules are those that are coincidentally cohesive.

### The worst degree of cohesion

Coincidental cohesion is found in a component whose parts are unrelated to another.

* **Logical Cohesion:** It is where several logically related functions or data elements are placed in same component.
* **Temporal Cohesion:** It is when a component that is used to initialize a system or set variables performs several functions in sequence, but the functions are related by timing involved.
* **Procedurally Cohesion:** It is when functions are grouped together in a component just to ensure this order.
* **Sequential Cohesion:** It is when the output from one part of a component is the input to the next part of it.

# Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well- designed input forms and screens have following properties:

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding:
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### Objectives for Input Design

The objectives of input design are:

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

### Data Input Methods

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

A system should prevent user from making mistakes by:

* Clear form design by leaving enough space for writing legibly.
* Clear instructions to fill form.
* Clear form design
* Reducing key strokes
* Immediate error feedback

Some of the popular data input methods are:

* Batch input method (Offline data input method)
* Online data input method
* Computer readable forms

Interactive data input

### Input Integrity Controls

Input integrity controls include a number of methods to eliminate common input errors by end-users. They also include checks on the value of individual fields; both for format and the completeness of all inputs.

Audit trails for data entry and other system operations are created using transaction logs which gives a record of all changes introduced in the database to provide security and means of recovery in case of any failure .

# Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### Objectives of Output Design

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end users requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions. Let us now go through various types of outputs:

### External Outputs

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

### Internal outputs

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

There are three types of reports produced by management information:

* **Detailed Reports:** They contain present information which has almost no filtering or restriction generated to assist management planning and control.
* **Summary Reports:** They contain trends and potential problems which are categorized and summarized that are generated for managers who do not want details.
* **Exception Reports:** They contain exceptions, filtered data to some condition or standard before presenting it to the manager, as information.

### Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

# Forms Design

Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading. For example, order forms, employment and credit application, etc.

* During form designing, the designers should know:
  + who will use them
  + where would they be delivered
  + the purpose of the form or report
* During form design, automated design tools enhance the developer’s ability to prototype forms and reports and present them to end users for evaluation.

### Objectives of Good Form Design

A good form design is necessary to ensure the following:

* To keep the screen simple by giving proper sequence, information, and clear captions.
* To meet the intended purpose by using appropriate forms.
* To ensure the completion of form with accuracy.
* To keep the forms attractive by using icons, inverse video, or blinking cursors etc.
* To facilitate navigation.

### Types of Forms Flat Forms

* It is a single copy form prepared manually or by a machine and printed on a paper. For additional copies of the original, carbon papers are inserted between copies.
* It is a simplest and inexpensive form to design, print, and reproduce, which uses less volume.

### Unit Set/Snap out Forms

* These are papers with one-time carbons interleaved into unit sets for either handwritten or machine use.
* Carbons may be either blue or black, standard grade medium intensity. Generally, blue carbons are best for handwritten forms while black carbons are best for machine use.

### Continuous strip/Fanfold Forms

* These are multiple unit forms joined in a continuous strip with perforations between each pair of forms.
* It is a less expensive method for large volume use.

### No Carbon Required (NCR) Paper

* They use carbonless papers which have two chemical coatings (capsules), one on the face and the other on the back of a sheet of paper.
* When pressure is applied, the two capsules interact and create an image.