Unit 3: Structured Analysis

AGENDA

- Structured Analysis
- Structured English
- Decision Tables
- Methods of Performing Cost Benefit Analysis
- System Logical and Physical Design
- Selection of Hardware and Software,
- Criteria to Evaluate Hardware and Software.

Structured Analysis

- Analysis is the heart of the process. It is the key component of the first two phases of the cycle.
- In analysis the present system, the analyst collects a great deal of relatively unstructured data through interviews, questionnaires, on–site observations, procedures manuals, and the like.
- The traditional approach is to organize and convert the data though system flowcharts, which support future developments of the system and simplify communication with the user.
- But the system flowchart represents a physical rather than a logical system.
- It makes it difficult to distinguish between what happens and how it happens in the system.

Structured English

- Structured English is used to analyse decision process. It has a restricted vocabulary.
- It uses **action oriented verbs** like **add, multiply, and so on**. It does not use a subject and makes use of the terms defined in data dictionary.
- It is a tool that enables one to state the rules precisely.
- It uses logical constructs and imperative sentences that are instructive.
- It is a special language that uses procedural logic derived from structured programming. Keywords of many programming languages are derived from such strong verbs in assembly language or other languages sometimes using mnemonics.

Structured English

- Words, for example do, add, sum, etc., are used. Shortened form, such as EQ for equal is used in FORmula TRANslator (FORTRAN).
- Query languages make use of words of structured English, such as SELECT, JOIN, WHERE along with other combination of special characters, such as =, >, <, etc.
- Words, such as IF, THEN, ELSE, UNTIL, REPEAT and SO are used for decision-making. Structures show logical hierarchy.
- It is most useful if it is required to take into account loops and sequences in a program. It is also good for use in procedure manuals.
- When the decision structure is not complex, structured English can be used. By using this tool, structured instructions are grouped and nested in an organized manner.
- **To write structured English, the following points should be kept in mind**:

Structured English

1. Use sequential structures, iterations or decision structures while expressing logic.
2. Make use of keywords that exist already. Such keywords are IF, THEN, ELSE, DO, DO WHILE, etc.
3. Statement blocks should be properly indented to reflect hierarchical structure or nesting.
4. If phrases or words that you use are already defined in the data dictionary, underline them. This shows that they have special meaning.
5. Before starting the coding stage, clarify logical statements.

Structured English

- Two basic building blocks are logic or instructions that are organized as grouped procedure and are nested and strong verbs that are used for actions.
- These verbs are move, add, multiply, etc. Structured English is used for labelling decision trees and creating decision tables.
- Four major considerations for using structured English in decision trees are as follows:

1. Identifying conditions
2. Identifying outcomes for every decision
3. Identifying actions
4. Identifying rules

Structured English

- In decision tables, structured English words are used.
- They are needed when there are complex combinations, actions, rules and conditions on which actions are performed.
- Here, structured English provides a method that can help avoiding contradictions, ambiguity due to duplication and impossible situations.
- Decision trees are useful when every condition is not relevant for every action.
- Structured English is used in three basic constructs that are used in every programming language. These are as follows:

1. Sequence NOTES
2. Selection (if-else conditional)
3. Iteration (loops)

Structured English

Pseudocode

- It is used to express logic in the English language and has no conformation to any particular programming language.
- It is used while implementing physical design and even after it. It is not used for any coding.
- Pseudocode may be a good replacement of flowcharts used for program design.
- It is used to explain the program logic in combination with structured programming.
- This technique makes utilization of verbs, such as DO WHILE, UNTIL, PERFORM, ENDIF, and so on easier. It is structured but distorted English.

Structured English

Pseudocode

- Conditional statement: IF-THEN example
- **BEGIN IF**

  IF Customer Age > 65

  THEN Billing Rate = Senior Citizen Rate

  ELSE Billing Rate = Standard Rate

  END-IF
- Conditional statement: SELECT CASE example
- **SELECT CASE**

  CASE 1 (State = "AZ")

  Sales Tax = 0.075

  CASE 2 (State = "CA")

  Sales Tax = 0.05 END-CASE

Structured English

Pseudocode

- Conditional statement: DO-WHILE example

```
START
OPEN employee file
DO WHILE not end of file
        GET next record
        IF year of hire is greater than 2009
            DISPLAY name, date of hire fields
            ADD 1 to employee count
        ENDIF
ENDDO
CLOSE employee file
DISPLAY employee count
EXIT
```

Structured English

- Things to remember:
- Capitalize keywords such as GET, IF, THEN, ELSE, END IF, CASE, DO WHILE, etc.
- Express actions as concise, simple commands. E.g.
  - GET the employee's Employee Record from Employee File
  - SEND the employee's Paycheque to Employee
- Express CASE or IF conditions as simple logical expressions that anyone can understand (e.g., use of "=", "<= 21", etc is fine)
  - If age<=21 THEN …
  - CASE 1 (Gender = Male)
  - CASE 1 (Weight < 100 pounds)
- For situations in which the IF … THEN logic will include a series of conditions, use ELSE and ELSEIF keywords (as seen in the example 6.1)

Structured English

- Things to remember:
- If process steps will be repeated, then use a repetition loop (e.g., DO WHILE … END DO WHILE)
- Try to avoid complex statements, if possible (see example below):
- The following conditional statement is readable, but some people may need to think twice to interpret the portion of the statement that is to the right of the "<=":

  IF Start Date <= Today's Date – 30 THEN …

- When appropriate, use indentation to enhance readability.

Decision Tables

- Decision table is an excellent tool for expressing complex logical relationships in a highly understandable and precise manner.
- A decision table is a matrix containing rows and columns which are used to define relationships.
- A decision table consists of two parts in which one is STUB and the other is ENTRY. The STUB is further divided into the upper and lower quadrant.
- The upper quadrant is known as condition STUB and the lower quadrant is known as action STUB.
- The ENTRY part is also subdivided into upper quadrant and lower quadrant.

Decision Tables

- The upper quadrant is known as condition entry and the lower quadrant is known as action entry. In a decision tree:
  - Y represents the existence of a condition.
  - N represents the condition which is evaluated and is not satisfied.
  - A blank space represents the condition involved that has been tested.
- There are three major components in a decision table. They are conditions, actions and decision rules.
- Conditions are events or facts that determine the course of action to be taken.
- Actions are the processes that get activated under certain conditions.
- Decision rules express the relationships between combination of conditions and courses of action.

Decision Tables

- Decision tables are useful in situations where the resulting actions depend on the occurrence of one or several combinations of independent conditions.

Conditions or causes	Condition rules (alternatives) represent the combinations of causes
Actions or effects	Action entries represent the combination of effects

Table 6.2 Quadrants of a Decision Table

- Conditions are put in quadrants at the upper left corner of the table and condition rules (alternatives) are placed in the upper right corner.
- The corner at quadrant on the lower left has actions needed and at the lower right there are rules for action.
- Conditions are also known as causes and actions are also known as effects.

Decision Tables

- Causes, values and combinations are considered as prime factors to summarize the decision table.
- **Example 1:** Give a discount of 5% if the customer pays advance or if the purchase is for Rs. 10,000 or more and the customer is a regular customer.

CONDITIONS	RULE1	RULE2	RULE3	RULE4
Advance payment made	Y	N	N	N
Purchase amt >=10,000	-	Y	Y	N
Regular Customer?	-	Y	N	-
ACTIONS				
Give 5% Discount	X	X	-	-
Give No Discount	-	-	X	X

Decision Tables

- A bank will grant loan under the following conditions:

1. If a customer has an account with the bank and had no loan outstanding, loan will be granted.
2. If a customer has an account with the bank but some amount is outstanding from previous loans then loan will be granted if special approval is given.
3. Reject all loan applications in all other cases.

Conditions	**Rule1**	**Rule2**	**Rule3**	**Rule4**
customer has a Bank Account	**Y**	**Y**	**Y**	**N**
Customer has no dues from previous account	**Y**	**N**	**N**	–
Management Approval is obtained	-	**Y**	**N**	–
Actions				
Approve Loan	**Y**	**Y**	-	-
Reject Loan	-	-	**N**	**N**

Decision Tables

- **Example 3:** In the following Table, there are ‘causes’ on the left upper corner and the number of causes are entered below it.
- Then below this head comes ‘effects’ and under it the possible effects, as relevant to the application are put.
- The Table also shows three causes (cause 1, cause 2 and cause 3). Under ‘effect’ only two effects are entered. Analyse it.

		Combinations							
Causes	**Values**	1	2	3	4	5	6	7	8
Cause 1	*Y, N*	Y	Y	Y	Y	N	N	N	N
Cause 2	*Y, N*	Y	Y	N	N	Y	Y	N	N
Cause 3	*Y, N*	Y	N	Y	N	Y	N	Y	N
Effects									
Effect 1		X			X				X
Effect 2			X				X		X

Methods of Performing Cost Benefit Analysis

- A Cost-Benefit Analysis (CBA) is done to determine how well or how poorly a planned action will turn out.
- Although a CBA can be used for almost anything, it is most commonly done on financial questions.
- Since the CBA relies on the addition of positive factors and the subtraction of negative ones to determine a net result, it is also known as running the numbers.
- The benefits of a CBA are it finds, quantifies and adds all the positive factors. Then, it identifies, quantifies and subtracts all the negatives—the costs.
- Business-savvy organizations use CBA to identify the best choice making sure to include all the costs and all the benefits and properly quantify them.

Methods of Performing Cost Benefit Analysis

- A credible CBA provides hard facts that answer three fundamental questions:
    1. What must be accomplished to succeed?
    2. What are the pros and cons of the viable choices?
    3. Which choice is the best?
- Cost and Benefit Evaluation Methods:
    1. Net Benefit Analysis
    2. Present Value Analysis
    3. Net Present Value Analysis
    4. Cash Flow Analysis
    5. Payback Analysis

Methods of Performing Cost Benefit Analysis

- **Net Benefit Analysis**
- Net benefit is total benefit net of total costs or total benefit minus total cost. The steps to calculate the net benefit are as follows:
- 1. Find out the components of benefit (due to an economic decision/economic project/investment).
- 2. Measure and value the benefit in different components and aggregate them (add them suitably because there could be benefits of different types/components flowing over a period of time).
- 3. The same thing applies to all cost components: identify them, measure and value them and then aggregate them suitably.
- 4. Once the values of the total benefits and total costs are computed, it is simple to arrive at the net benefit as the difference between the two.
- The drawbacks of this method are as follows:
  - It does not involve the time value of money.
  - It does not discount future cash flow.

Methods of Performing Cost Benefit Analysis

- **Present Value Analysis**
- Usually in long-term projects, it is difficult to calculate and compare the cost incurred today with the complete value of tomorrow's benefits.
- Therefore, to overcome this problem, present value analysis calculates the costs and benefits in present day's terms, i.e. today's value of investment costs and then compares the cash flows at different times.
- Present value is the value on a given date of a future payment or series of future payments, discounted to reflect the time value of money and other factors such as investment risk.

Methods of Performing Cost Benefit Analysis

- **Present Value Analysis**
- Present value calculations are widely used in business and economics to provide a means to compare cash flows at different times on a meaningful ‘like to like’ basis.
- Present value analysis is the application of an appropriate discount rate to a stream of future cash flows. It allows differing payment streams to be compared.
- The formula for calculating the present value is

$$P = F/(1+i)^n$$

Where F is future value [$F=P(1+i)^n$] and I is the interest at the end of the year (n).

Methods of Performing Cost Benefit Analysis

- **Net Present Value Analysis(NPV)**
- NPV is the difference between the present value of cash inflows and the present value of cash outflows.
- It is used in capital budgeting to analyse the profitability of an investment or project.
- In other words, NPV is calculated by summing the rupee-valued benefits and then subtracting all of the rupee-valued costs, with discounting applied to both the benefits and the costs as appropriate.
- NPV analysis is sensitive to the reliability of future cash inflows that an investment or project will yield. The formula for calculating the NPV as follows:

$$NPV = \sum_{t=0}^{n} \frac{(\text{Benefits} - \text{Costs})}{(1+r)^t}$$

Where:

r = discount rate

t = year

n = analytic horizon (in years)

Methods of Performing Cost Benefit Analysis

- **Net Present Value Analysis(NPV)**
- The net present value is a measure of the value of a project:
  - If the project NPV is positive, the project creates value **and**
  - If the project NPV is negative, the project may not create value.

Methods of Performing Cost Benefit Analysis

- **Cash Flow Analysis**
- Cash flow is the inflow and outflow of money from business to determine the business' solvency.
- Cash flow analysis is the study of the cycle of your business cash inflows and outflows, with the purpose of maintaining an adequate cash flow for your business and providing the basis for cash flow management.
- Cash flow analysis involves examining the components of your business that affect cash flow, such as accounts receivable, inventory, accounts payable and credit terms.
- By performing a cash flow analysis on these separate components, you will be able to identify more easily cash flow problems and find ways to improve your cash flow.

Methods of Performing Cost Benefit Analysis

- **Payback Analysis**
- Payback analysis is simply a calculation of how long it will take to recover your investment.
- Simple payback is a common economic analysis method and is understood by most business owners.
- Simple payback is the amount of time it will take to recover installation costs based on annual energy cost savings.
- The equation for simple payback is annual energy cost savings per year divided by the initial installation cost.
- The shorter the payback period, the sooner the organization will make profits and more attractive is the investment. He PA formula is:

$$\text{Payback Period (in years)} = \frac{\text{Initial Investment}}{\text{Annual Savings (Cash Flow)}}$$

Selection of Hardware and Software

- A major element in building system is selecting compatible hardware and software.
- Hardware and software selection begins with requirements analysis, followed by a request for proposal and vendor selection.
- **Steps of Selection Process are:**
  - Requirement Analysis
  - System Specifications
  - Request for proposal
  - Evaluation & validation
  - Vendor selection
  - Post-installation review

Selection of Hardware and Software

- **Requirements analysis**
- The first step in selection is understanding the user's requirements within the framework of the organization's objectives and the environment in which the system is being installed.
- Consideration is given to the user's resources as well as to finances.
- In selecting software, the user must decide whether to develop it in house, hire a software company or contract programmer to create it, or simply acquire it from a software house.
- The choice is logically made after the user has clearly defined the requirements expected of the software.
- Therefore, requirements analysis sets the tone for software selection.

Selection of Hardware and Software

- **System Specifications**
- Failure to specify system requirements before the final selection almost always results in a faulty acquisition.
- The specifications should delineate the user's requirements and allow room for bids from various vendors.
- They must reflect the actual applications to be handled by the system and include system objectives, flowcharts, input-output requirements, file structure and cost.
- The specifications must also describe each aspect of the system clearly, consistently and completely.

Selection of Hardware and Software

- **Request for Proposal**
- After the requirements analysis and system specifications have been determined, a request for proposal is drafted and sent to selected vendors for bidding.
- Bids submitted are based on discussions with vendors. At a minimum, the RFP should include the following

1. Complete statement of the system specifications, programming language, price range, terms and time frame.
2. Request for vendor's responsibilities for conversion, training and maintenance
3. Warranties and terms of license or contractual limitations.
4. Request for financial statement of vendor
5. Size of staff available for system support

Selection of Hardware and Software

- **Evaluation and validation**
- The evaluation phase ranks vendor proposals and determines the best suited to the user's needs.
- It looks into items such as price, availability and technical support. System validation ensures that the vendor can match his/her claims, system performance.
- True validation is obtained verified by having each system demonstrated. An outside consultant can be employed for consulting purpose.

Selection of Hardware and Software

- **Vendor selection**
- This step determines the winner – the vendor with the best combination of reputation, reliability, service record, training, delivery time, lease finance terms and conversion schedule.
- Initially a decision is made which vendor to contact.
- The sources available to check on vendors include the following:

1. Users
2. Software houses
3. Trade associations
4. Universities
5. Publications/Journals
6. Vendor software lists
7. Vendor referral directories
8. Published directories
9. Consultants
10. Industry contacts

Selection of Hardware and Software

- **Post- installation Review**
- Sometime after the package is installed, a system evaluation is made to determine how closely the new system conforms to plan.
- System specifications and user requirements are audited to pinpoint and correct any differences
- **Software selection**
- Software selection is a critical aspect for system development. There are 2 ways of acquiring the software.
  - Custom -made
  - Packages

Criteria for Hardware and Software Selection

- **Reliability** – It is the probability that the software will executed in a specific period of time without any failures. It is important to the professional user. It brings up the concept of modularity, or the ease which a package can be modified.
- **Functionality** – It is the definition of the facilities, performance and other factors that the user requires in the finished product.
- **Capacity** – Capacity refers to the capability of the software package to handle the users requirements for size of files, number of data elements, and reports. All limitations should be checked.
- ***Usability* –** This criteria refers to the effort required to operate, prepare the input, and interpret the output of a program.
- Additional points considered here are portability and understandability. Portability refers to the ability of the software to be used.
- Understandability is the purpose of the product.

Criteria for Hardware and Software Selection

- **Flexibility** – It is a measure of effort required to modify an operational program. One feature of flexibility is adaptability.
- **Security** – It is a measure of the likelihood that a system's user can accidentally or intentionally access or destroy unauthorized data.
- **Performance** – It is a measure of the capacity of the software package to do what it is expected to do. This criteria focuses on throughput or how effectively a package performs under peak load.
- **Serviceability** –This criteria focuses on documentation and vendor support.
- **Ownership** – Who owns the software ,and to consider whether he has the right to access the software, or he can sell or modify the software.
- **Minimal costs** – Cost is a major consideration in deciding between in-house and vendor software.

Criteria for Hardware and Software Selection

- **Evaluation process**
- There are three process for evaluating hardware and software

1. **Benchmark programs**:

- It is a sample program for evaluating different computers and their software. It is necessary because computers often uses the same instructions, words of memory or machine cycle to solve a problem. Benchmarking includes the following
  - Determination of the minimum hardware.
  - An acceptance test
  - Testing in an ideal environment to determine the timings and in the normal environment to determine its influence on other programs.

Criteria for Hardware and Software Selection

2. **Experience of other users**

- Benchmarking only validates vendors' claims. Experience of other users with the same system software is essential.

3. **Product reference manuals**

- These evaluate a system's capability. These reports elaborate on computer products, services and prices.

Criteria for Hardware and Software Selection

Evaluation of proposals

- After all proposals are evaluated, the final vendor is selected using any of the 3 methods
- 1. adhoc refers to the user's inclination to favor one vendor over others.
- 2. Scoring. In this method the characteristics of each system are listed and score is given in relation to the maximum point rating. Then each proposal is rated according to its characteristics.
- 3. Cost value approach. In this method a dollar credit method is applied to the proposal that meets the user's desirable characteristics.
- This credit is subtracted from the vendor's quoted price. The proposal with the lowest price is selected.

Logical and Physical Design

- Software implementation requires proper planning. A design is of no use if it can not be implemented.
- Implementation is a process that converts logical design into physical design that can be made operational such that users can take over to operate and evaluate it.
- Once the design phase is over, developers have to put these together and tested for operation involving users.
- In the design stage individual components are tested but the real task of putting these together so that it works as a system to the satisfaction of the user who wanted this system to behave the way he/she wanted.

Logical and Physical Design

- In the implementation stage, no changes are made.
- In all previous stages, the developers had freedom of making changes, if any, in the software design plan.
- You develop the logical design of an information system during the systems analysis phase.
- The logical design defines the functions and features of the system and the relationship among its components.
- The logical design includes the output that must be produced by the system, the input needed by the system, and the process that must be produced by the system without regard to how tasks will be accomplished physically.

Logical and Physical Design

- **Logical Design**
- A logical design defines what must take place, not how it is to be accomplished.
- Logical designs do not address the actual methods of implementation.
- 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 modelling are used.

Logical and Physical Design

- **Logical Design**
- Example: The Logical Design of a Customer Record System
- Describes the data that must be entered for each customer and specifies that records must be displayed in the order of customer number, and explains what information to produce for a customer status report.
- Specifications for the actual input, or entry, of data, the sorting method, the physical process of creating the report, and the exact format of the port are not part of the logical design.

Logical and Physical Design

- **Physical Design**
- The physical design of an information system is a plan for the actual implementation of the system.
- You develop the physical design during the systems design phase.
- The physical design is built on the system's logical design and describes a specific implementation, much like a working blue print describes the actual construction of a building.
- The physical design describes the actual processes of entering, verifying, and storing data; the physical layout of data files; the sorting procedures; the exact format of reports; and so on.
- Where as logical design is concerned with what the system must accomplish, physical design is concerned with how the system will meet those requirements.

Logical and Physical Design

- **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 −

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