# UNIT 2 (Information Requirement Analysis )

**3. System Planning**

# What is Requirements Determination?

A requirement is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management.

Requirements determination involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made.

# Major Activities in requirement Determination

### Requirements Anticipation

* It predicts the characteristics of system based on previous experience which include certain problems or features and requirements for a new system.
* It can lead to analysis of areas that would otherwise go unnoticed by inexperienced analyst. But if shortcuts are taken and bias is introduced in conducting the investigation, then requirement Anticipation can be half-baked.

### Requirements Investigation

* It is studying the current system and documenting its features for further analysis.
* It is at the heart of system analysis where analyst documenting and describing system features using fact-finding techniques, prototyping, and computer assisted tools.

### Requirements Specifications

* It includes the analysis of data which determine the requirement specification, description of features for new system, and specifying what information requirements will be provided.
* It includes analysis of factual data, identification of essential requirements, and selection of Requirement-fulfillment strategies.

# Information Gathering Techniques

The main aim of fact finding techniques is to determine the information requirements of an organization used by analysts to prepare a precise SRS understood by user.

Ideal SRS Document should:

* be complete, Unambiguous, and Jargon-free.
* specify operational, tactical, and strategic information requirements.
* solve possible disputes between users and analyst.
* use graphical aids which simplify understanding and design.

# Feasibility Study

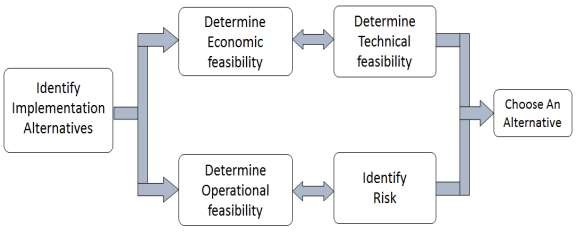
Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

* It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
* It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
* The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
* The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

# Steps Involved in Feasibility Analysis

The following steps are to be followed while performing feasibility analysis:

1. Form a project team and appoint a project leader.
2. Develop system flowcharts.
3. Identify the deficiencies of current system and set goals.
4. Enumerate the alternative solution or potential candidate system to meet goals.
5. Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
6. Weight the performance and cost effectiveness of each candidate system.
7. Rank the other alternatives and select the best candidate system.
8. Prepare a system proposal of final project directive to management for approval.



# Types of Feasibilities

### Economic Feasibility

* It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
* It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
* The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
* It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

### Technical Feasibility

* It investigates the technical feasibility of each implementation alternative.
* It analyzes and determines whether the solution can be supported by existing technology or not.
* The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
* It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

### Operational Feasibility

* It determines whether the system is operating effectively once it is developed and implemented.
* It ensures that the management should support the proposed system and its working feasible in the current organizational environment.
* It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits.
* It also ensures that the computer resources and network architecture of candidate system are workable.

### Behavioral Feasibility

* It evaluates and estimates the user attitude or behavior towards the development of new system.
* It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee’s job status on new ways of conducting business.

### Schedule Feasibility

* It ensures that the project should be completed within given time constraint or schedule.
* It also verifies and validates whether the deadlines of project are reasonable or not.

Analysts use various tools to understand and describe the information system. One of the ways is using structured analysis.

# What is Structured Analysis?

Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user.

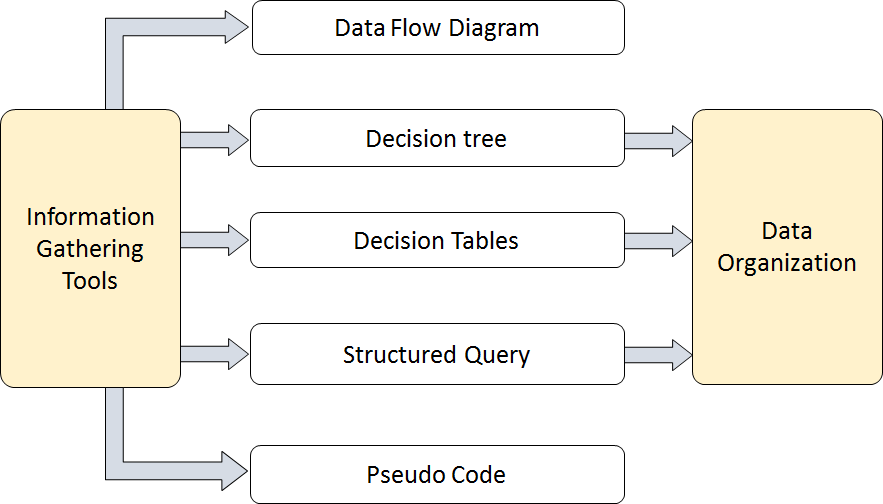
It has following attributes:

* It is graphic which specifies the presentation of application.
* It divides the processes so that it gives a clear picture of system flow.
* It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
* It is an approach that works from high-level overviews to lower-level details.

# Structured Analysis Tools

During Structured Analysis, various tools and techniques are used for system development. They are:

* Data Flow Diagrams
* Data Dictionary
* Decision Trees
* Decision Tables
* Structured English
* Pseudocode



# Data Flow Diagrams (DFD) or Bubble Chart

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

* It shows the flow of data between various functions of system and specifies how the current system is implemented.
* It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
* Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
* It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.

### Basic Elements of DFD

DFD is easy to understand and quite effective when the required design is not clear and the user wants a notational language for communication. However, it requires a large number of iterations for obtaining the most accurate and complete solution.

The following table shows the symbols used in designing a DFD and their significance:

|  |  |  |
| --- | --- | --- |
| **Symbol Name** | **Symbol** | **Meaning** |
| Square |  | Source or Destination of Data |
| Arrow |  | Data flow |
| Circle |  | Process transforming data flow |
| Open Rectangle |  | Data Store |

### Types of DFD

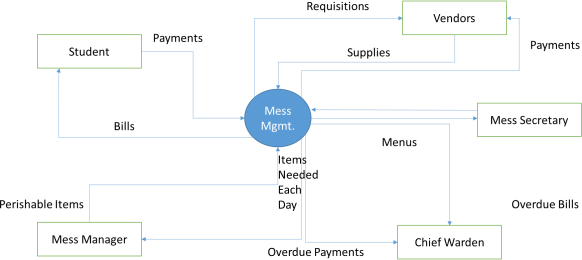
DFDs are of two types: Physical DFD and Logical DFD. The following table lists the points that differentiate a physical DFD from a logical DFD.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Physical DFD** | | | **Logical DFD** | | | |
| It is implementation dependent. It shows which functions are performed. | | | It is implementation independent. It focuses only on the flow of data between processes. | | | |
| It provides low level details of hardware, software, files, and people. | | | It explains events of required by each event. | systems | and | data |
| It depicts how the operates and how a implemented. | current system | system will be | It shows how business operates; not how the system can be implemented. | | | |

### Context Diagram

A context diagram helps in understanding the entire system by one DFD which gives the overview of a system. It starts with mentioning major processes with little details and then goes onto giving more details of the processes with the top-down approach.

The context diagram of mess management is shown below.



# Data Dictionary

A data dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements that is, details and definitions of data flows, data stores, data stored in data stores, and the processes.

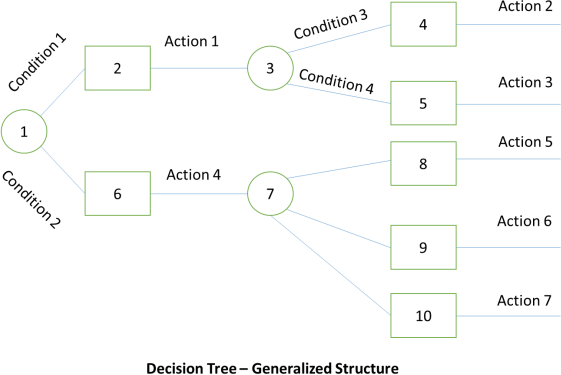
A data dictionary improves the communication between the analyst and the user. It plays an important role in building a database. Most DBMSs have a data dictionary as a standard feature. For example, refer the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Data Name** | **Description** | **No. of Characters** |
| 1 | ISBN | ISBN Number | 10 |
| 2 | TITLE | Title | 60 |
| 3 | SUB | Book Subjects | 80 |
| 4 | ANAME | Author Name | 15 |

# Decision Trees

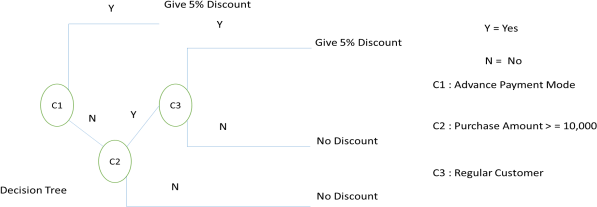
Decision trees are a method for defining complex relationships by describing decisions and avoiding the problems in communication. A decision tree is a diagram that shows alternative actions and conditions within horizontal tree framework. Thus, it depicts which conditions to consider first, second, and so on.

Decision trees depict the relationship of each condition and their permissible actions. A square node indicates an action and a circle indicates a condition. It forces analysts to consider the sequence of decisions and identifies the actual decision that must be made.



The major limitation of a decision tree is that it lacks information in its format to describe what other combinations of conditions you can take for testing. It is a single representation of the relationships between conditions and actions.

For example, refer the following decision tree:



# Decision Tables

Decision tables are a method of describing the complex logical relationship in a precise manner which is easily understandable.

* It is useful in situations where the resulting actions depend on the occurrence of one or several combinations of independent conditions.
* It is a matrix containing row or columns for defining a problem and the actions.

### Components of a Decision Table

* **Condition Stub:** It is in the upper left quadrant which lists all the condition to be checked.
* **Action stub:** It is in the lower left quadrant which outlines all the action to be carried out to meet such condition.
* **Condition Entry:** It is in upper right quadrant which provides answers to questions asked in condition stub quadrant.
* **Action Entry:** It is in lower right quadrant which indicates the appropriate action resulting from the answers to the conditions in the condition entry quadrant.

The entries in decision table are given by Decision Rules which define the relationships between combinations of conditions and courses of action. In rules section,

* Y shows the existence of a condition.
* N represents the condition, which is not satisfied.
* A blank - against action states it is to be ignored.
* X (or a check mark will do) against action states it is to be carried out.

For example, refer the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CONDITIONS** | **Rule 1** | **Rule 2** | **Rule 3** | **Rule 4** |
| Advance payment made | Y | N | N | N |
| Purchase amount = Rs 10,000/- | - | Y | Y | N |
| Regular Customer | - | Y | N | - |
| **ACTIONS** |  |  |  |  |
| Give 5%  discount | X | X | - | - |
| Give no discount | - | - | X | X |

# Structured English

Structure English is derived from structured programming language which gives more understandable and precise description of process. It is based on procedural logic that uses construction and imperative sentences designed to perform operation for action.

* It is best used when sequences and loops in a program must be considered and the problem needs sequences of actions with decisions.
* It does not have strict syntax rule. It expresses all logic in terms of sequential decision structures and iterations.

For example, see the following sequence of actions:

if customer pays advance then

Give 5% Discount

else

if purchase amount >=10,000 then

if the customer is a regular customer then Give 5% Discount

else No Discount

end if else No Discount

end if

end if

# Pseudocode

A pseudocode does not conform to any programming language and expresses logic in plain English.

* It may specify the physical programming logic without actual coding during and after the physical design.
* It is used in conjunction with structured programming.
* It replaces the flowcharts of a program.

# Guidelines for Selecting Appropriate Tools

Use the following guidelines for selecting the most appropriate tool that would suit your requirements:

* Use DFD at high or low level analysis for providing good system documentations.
* Use data dictionary to simplify the structure for meeting the data requirement of the system.
* Use structured English if there are many loops and actions are complex.
* Use decision tables when there are a large number of conditions to check and logic is complex.

Use decision trees when sequencing of conditions is important and if there are few conditions to be tested

### Entity Relationship Model

It is a technique used in database design that helps describe the relationship between various entities of an organization.

### Terms used in E-R 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 E-R model and their respective meanings:

The following table shows the symbols used in E-R model and their significance:

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
|  | Entity |
|  | Weak Entity |
|  | Relationship |
|  | Identity Relationship |

|  |  |
| --- | --- |
|  | Attributes |
|  | Key Attributes |
|  | Multivalued |
|  | Composite Attribute |
|  | Derived Attributes |
| **E1 R E2** | Total Participation of E2 in R |
| **E1 R E2** | Cardinality Ratio 1:N for E1:E2 in R |

Three types of relationships can exist between two sets of data: one-to-one, one-to-many, and many-to-many.