UNIT 4: REQUIREMENT ANALYSIS & DESIGN

1] CONCEPTS OF DATA MODELING

- Analysis modeling starts with data modeling
- The software engineer defines all the data objects that proceeds within the system and the relationships between data objects are identified.

DATA OBJECTS:

- The data object is the representation of composite information.
- The composite information means an object has number of different properties or attributes
- For example, Height is a single value so it is not a valid data object, but dimensions contain the height, the width and depth these are defined as an object.

DATA ATTRIBUTES

- Each of the data object has a set of attributes

Data Object has the following characteristics:

- Name an instance of the data object
- Describe the instance.
- Make reference to another instance in ot another table.

RELATIONSHIPS

- Relationships shows the relationship between data objects and how they are related to each other.

CARDINALITY

- Cardinality state the number of events of one object related to the number of events of another object.

The cardinality expressed as:

One to one (1:1)

One event of an object is related to one event of another object.
For example, one employee has only one ID.

One to many (1:N)

One event of an object is related to many events
For example, one college has many departments

Many to Many (M:N)

many events of one object are related to many objects of another ~~event~~ object.
For example, many customer place order for many products.

Modality

- If an event relationship is an optional then the modality is set to zero.
- If an event relationship is compulsory then modality of relationship is one.

2] DESIGN PROCESS AND DESIGN QUALITY

Introduction to Design Process

- The main aim of design engineering is to generate a model which shows firmness, delight and commodity.
- Software design is an iterative process through which requirements are translated into the blueprint for building the software.

2.1] Software Quality Guidelines

- A design is generated using the recognizable architectural styles and compose a good design characteristic of components and it is implemented in evolutionary manner for testing.
- A design of the software must be modular i.e the software must be logically partitioned into elements.
- In design, the representation of data, architecture interface and components should be distinct.
- A design must carry appropriate structure and recognizable data

- Design components must show the independent functional characteristic
- A design created an interface that reduce the complexity of connections between the components.
- A design must be derived using the repeatable method.
- The notations should be use in design which can effectively communicates its meaning.

2.2] QUALITY ATTRIBUTES

The attributes of design name as 'FURPS' are as follows:

1] FUNCTIONALITY:
It evaluates the feature set and capabilities of the program

2] USABILITY:
It is accessed by considering the factor such as human factor, overall as aesthetics, consistency and documentation.

3] RELIABILITY:
It is evaluated by measuring parameters like frequency and security of failure, output result accuracy, the mean-time-to-failure (MTFF), recovery from failure and the program predictability

4] PERFORMANCE

It is measured by considering processing speed, response time, resource consumption, throughput and efficiency.

5] SUPPORTABILITY

- It combines the ability to extend the program, adaptability, serviceability. These three term defines the maintainability
- Testability, compatibility and configurability are the terms using which a system can be easily instaued and found the problem easily.
- Supportability also consits of more attributes such as compatiability, extensibility, fault tolerance, modularity, reusability, robustness, security, probability and scalability.

3] DESIGN CONCEPTS

a) ABSTRACTION

- There are many levels of abstraction i.e At the highest level of abstraction, a solution is stated in broad terms using the language of the problem enviornment.
- At lower level of abstraction, a more detailed description of the solution is provided

→ There are different levels of abstraction

a) A procedural abstraction refers to a sequence of instruction that have a specific and limited function

b) A data abstraction is a named collection of data that describe a data object.

b) ARCHITECTURE

- Software architecture specify "the overall structure of the software & the ways in which that structure provides conceptual integrity for a system"

- one goal of software design is to derive an architectural rendering of a system.
- A set of architectural patterns enables a software engineer to solve common design problems.

c] PATTERNS

→ A design pattern describes a design structure that solves a particular design problem within a specific context & amid "forces that may have an impact on the manner in which the pattern is applied and used.

→ It provides the following details:

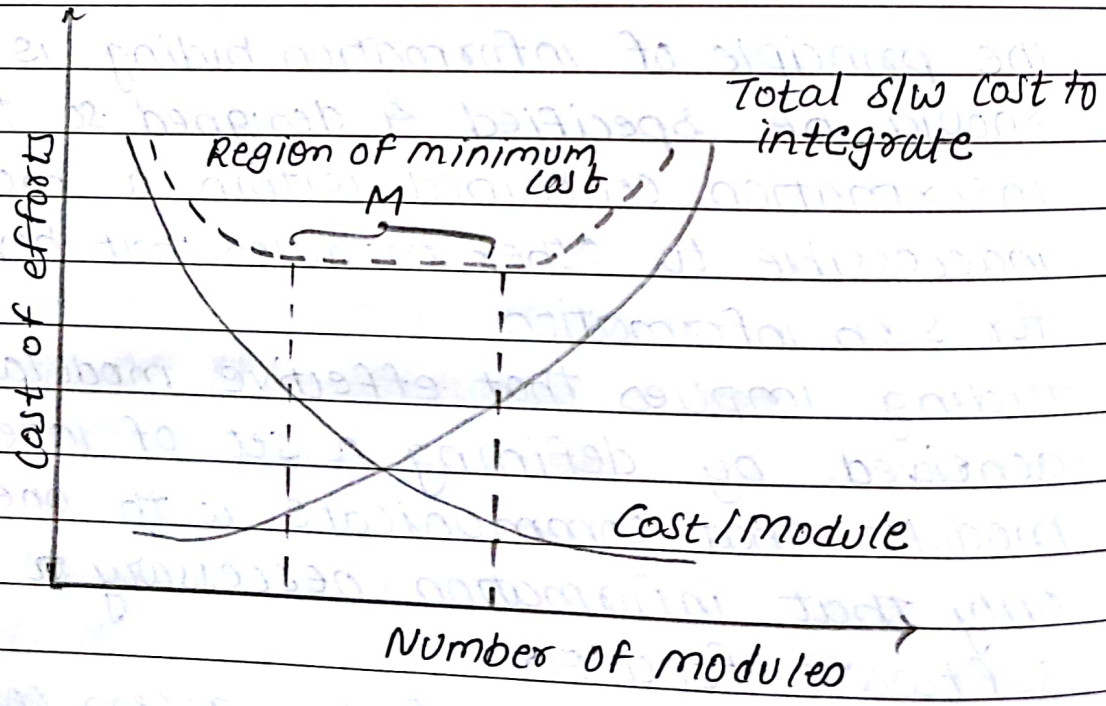
- Whether the pattern is applicable to the current work
- Whether the patter can be reused
- Whether the pattern can be served as a guide for developing a similar, but functionality different pattern

d] INFORMATION HIDING

- The principle of information hiding is, modules should be specified & designed so that the information contained within a module is inaccessible to other modules that have no need for such information.
- Hiding implies that effective modularity can be acheieved by defining a set of independent modules that communicate with one another only that information neccessary to achieve software function
- Hiding defines and enforces accen constraints to both procedural details within a module & any local data structure used by modules.

c] MODULARITY

- The software is divided into separately named and addressable components, called as modules
- Fig below The effort (cost) to develop an indivival software module does decrease as the total no. of modules increases. However, as the number of modules grows, the effort (cost) associated with integrating the modules also grows
- These characteristics lead to a total cost or effort curve shown in fig
- There is a ~~AUP~~ M number of a modules that would result in minimum development cost.



Modularity of S/W & cost

6] FUNCTIONAL INDEPENDENCE

- The functional independence is a concept of separation and related to the concept of modularity, abstraction and information hiding.
- The functional independence is accessed using two criteria i.e Cohesion and Coupling

Cohesion

- It is an extension of the information hiding concept
- A cohesive module performs a single task and it requires a small interaction with the other components in other parts of the program.

Coupling

- Coupling is an indication of interaction between modules in a structure of software
  - Coupling depends on the interface complexity between modules, the point at which entry or reference is made to a module, and what data pass across the interface.

9] REFINEMENT

- Refinement is actually a process of elaboration. The statement describe function or information conceptually but provides no information about the internal working of the function or the internal structure of the information.

- Then you need to elaborate on the original statement, providing more & more detail.
- Abstraction & refinement are complementary concepts. Abstraction enables you to specify procedure i.e about low-level details.
- Refinement helps you to reveal low-level details as design progress.

h) REFACTORING

- Refactoring is a reorganization technique that simplifies the design of a component without changing its function or behaviour.
- Refactoring is the process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure.

REQUIREMENT ENGINEERING

- Designing and building computer software is challenging, creative and just plain fun. The broad spectrum of task & techniques that leads to an understanding of requirement is called Requirement engineering

- Requirements Engineering builds a bridge to design & construction. It provides the appropriate mechanism for understanding what the customer wants, analyzing needs, assessing feasibility, negotiating a reasonable solution, validating the specification & managing the requirements as they are transformed into an operational system.

It has 7 distinct task:

a) **INCEPTION**

→ most projects begin when a business need is identified or a potential new market or service is discovered.

→ Stakeholders from the business community define a business case for the idea, try to identify the breadth & depth of the market, do a rough feasibility analysis & identify a working description of the project scope.

→ At project inception, you establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary

communication & collaboration between the other stakeholders & the software team

b) ELICITATION

→ It specify simply about asking the customer, the user and other what the objectives for the system or product are, what is to be accomplished, how the system or product fits into the needs of the business & finally how the system or product is to be used on a day-to-day basis

→ Problems of scope:

The customer/user specify unnecessary technical detail that may confuse, rather than clarifying overall system objectives

→ Problem of understanding

The customer/user are not completely sure of what is needed, having a poor understanding of capabilities & limitations of their computing enviornment.

→ Problem of volatility

The requirement changes overtime

ELABORATION

→ The information obtained from the customer during inception & elicitation is expanded and refined during elaboration

→ Elaboration is driven by the creation & refinement of user scenarios that describe how the enduser will interact with the system.

→ The attributes of each analysis class are defined and the services that are required by each class are identified.

→ The relationship & collaboration between classes are identified & a variety of supplementary diagrams are produced.