UNIT - III

Software design.

Q.10 What are the design quality attributes 'FURPS' meant?

- The design quality attributes popularly known as 'FURPS' (Functionality, Usability, Reliability, Performance and Supportability) is a set of criteria developed by Hewlett and Packard.

Following point represents meaning of each attributes.

i) Functionality
- It can be checked by assessing the set of features and capabilities of the functions.
- The function should be general and should not only work for particular set of inputs.
- Similarly the security aspect should be considered while designing the function

ii) Usability
- It can be assessed by knowing the usefulness of the system.

iii) Reliability.
- It is measure of frequency and severity of failure.

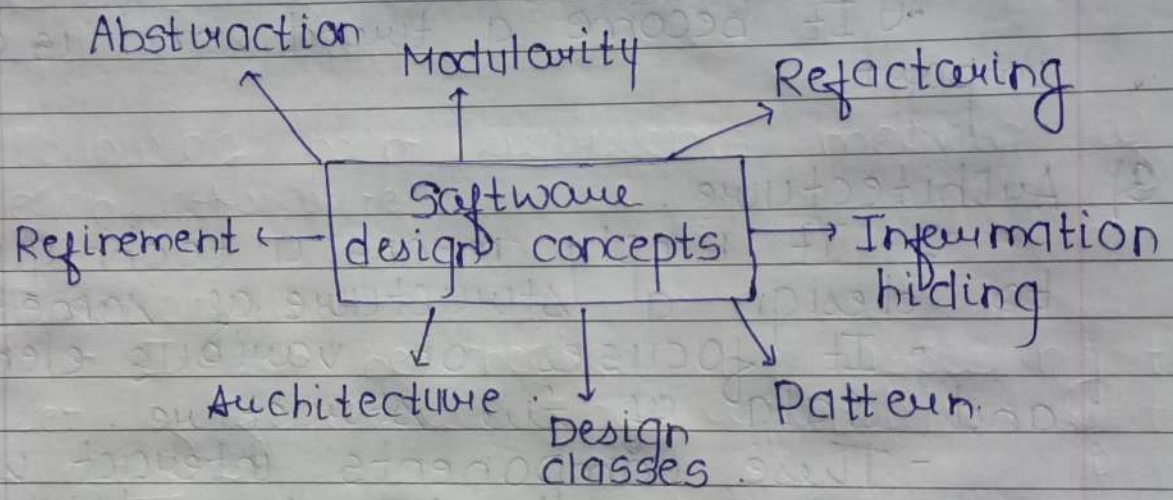
- MTTF is a metric that is widely used to measure the product's performance & reliability

4) Performance
- It is a measure represent the response of the system.
- Measuring it means measuring processing speed, memory usage, etc.

5) Supportability
- It is also called as maintainability.
- It is the ability to adopt the enhancement or changes made in the software.

Q.1
Q.3 Expain about various design concepts
Q.5 considered during design
Q.2.

→ Software design concepts simply means means the idea or principle behind the design.



1) Abstraction
- It means the ability to cope with complexity. Software design occurs at different level of abstraction
- At higher level of abstraction, the soln should be stated in broad terms and in the lower level more detail description of the soln is given.
- The procedural abstraction gives the named sequence of instructions in the specific function.

4343

Example: Search the record is a procedural abstraction in which implementation details are hidden.

2) Modularity
- It simply means dividing the system or project into smaller parts to reduce the complexity of project or system.
- It become a trend and is also impart.

3) Architecture.
- It simply means a technique to design a structure of somethich.
- It focuses on various elements and data of the structure.
- These components interact withe each other and usethe the data of the structure in architecture.

4) Refinement
- It means to refine something to remove impurities if present any increase the quality.
- The refinement is very neccessary to find out bany error if present and then to reduced it.

5) Refactoring
- It means reconstructing something

in such a way that it doesn't affect the behaviour of any other feature.
- It means reconstructing the design to reduce the complexity.
- The redundance can be achieved.

6) Information hiding.
- It is one of the important property of effective modular design.
- It simply means to hide the information so that it can't be accessed by an unwanted party.
- In software design it can be ~~achieved~~ achieved by designing the modules in a manner that the info. gathered or contained in one module is hidden and can't be accessed by any other modules.

Q.4 Concept of cohesion & coupling and diff. b/w them.

Answer. Cohesion

- With the help of this the information hiding can be done.
- A cohesive module performs only 'one task' in software procedure with little interaction with other module. In other words, cohesive module performs only one ~~task~~ thing.

Different types of cohesion are:

1) Coincidentally cohesive.
2) Logically cohesive.
3) Temporal cohesion
4) Procedural cohesion
5) Communicational cohesion

Coupling.

- It effectively represent how the modules can be connected to each other module or with outside the world.
- It is a measure of interconnection among modules in the program structure.
- It depends on the interface complexity between modules.

Types of module coupling.

i) Data coupling.
ii) Control coupling.
iii) Common coupling.
iv) Content coupling.

Coupling	Cohesion
1) It represent how the models are connected with each other or with outside world	In this the cohesive module perform one thing
2) With this, interface complexity is decided.	With this, data hiding can be done.
3) The goal is to achieve lowest coupling.	The goal is to achieve high cohesion
4) Various type Data coupling, control coupling etc.	Various types Coincidentally cohesion, Procedural cohesion, etc

Q.7 Architectural styles.

Answer It is a pattern for creating the system architecture for given problem.

The commonly used architectural styles are:

1) Data centered architectures.
2) Data flow architectures
3) Call and return architectures
4) Object oriented architectures
5) Layered architectures.

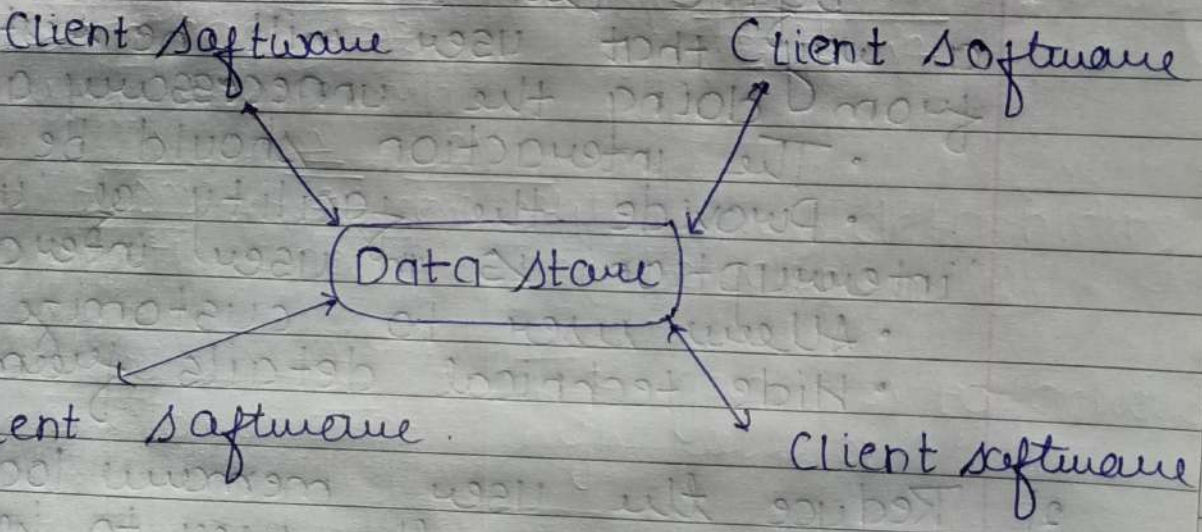
1) Data centered architectures.
   - In this, the data store lies at the center of the architecture and other components frequently accessing it by performing add, modify and delete operation.
   - The client software requests for the data to central repository.
   - It passes a property of interchangibility
   - In this, the data can be passed among the components.

In data centered architecture:

i) Components are, database elements such as tables, queries.

ii) Communication are by relationship.

iii) Constraint are: Client software has to request central data store for info.



Q.8 Golden rules of user interface design

Answer Thao Mandel has proposed three golden rules for user interface design.

1) Place the user in control.
2) Reduce the user's memory load.
3) Make the interface consitent.

1) Place the user in control.
- Define the interaction modes in such a way that user will be restricted from doing the unnecessary action.
- The interaction should be flexible.
- Provide the facility of 'undo' on 'interruption' is in user interaction.
- Allow user to customize interaction
- Hide technical details from users.

2) Reduce the user memory load.
- Don't force the user to have short term memory.
- Establish meaningful defaults.
- Use intuitive shortcuts
- The visual layout of the user interface should be realistic.
- Disclose the info. gradually.

3) Make the interface consitent.
- The visual info. should be consistent throught & it should be as per the design standards.

que. not done 11, 12, 9, 6

Ayushi

PAGE NO. 19
DATE: / /

- There should be a limited set of input holding non conflicting info.
- The info flow transiting from one task to another should be consitent.
- Maintain consitency across family of product.