

Module - 3

Assignment (Software Requirements)

Q1: Explain the Requirement Eng in details.

Soln: Requirements Engineering (RE) refers to the process of defining, documenting and maintaining requirements in the engineering design process. It provides the appropriate mechanism to understand what customer desires, analysing the need, and assessing feasibility, negotiating a reasonable solution, specifying the solution clearly, validating the specifications and managing the requirements as they are transferred into a working system.

Q2: List out the various types of requirements.

Ans: Largely software requirements must be categorized into two categories.

a) Functional requirements: Functional requirements define a function that a system or system elements must be qualified to perform and must be documented into different forms. The functional requirements are describing the behaviour of a function system as it co-relates to the system functionality.

(b) non functional requirements: This can be necessities that specify the criteria that can be used to decide the operation instead of the system as it co-relates behaviour.
non functional requirements are divided into two main categories.

(i) Execution qualities like security and usability, which are observable at runtime.

(ii) Evolution qualities: like testability, maintainability, extensibility and scalability that are embodied in the static structure of the software system.

Q3: State the importance of feasibility study.

Ans: The objective behind the feasibility study is to create the reasons for developing the software that is acceptable to users, flexible to change and conformable to establish standards.

Types of feasibility.

(i) Technical Feasibility: Technical feasibility evaluates the current technologies which are needed to accomplish customer req within the time and budget.

(ii) Operational Feasibility: Operational feasibility assesses the range in which the required software performs the scale series of levels to solve business problems & customer requirements.

(iii) Economic Feasibility: Economic feasibility decides whether the necessary software can generate financial profits for an organization.

Q4: Explain requirement elicitation how is it different from software req Engineering.

Soln- Requirement elicitation is the process of gathering and defining the requirements of a software system. The goal requirement elicitation is to ensure that the software development process is based on clear & comprehensive understanding of customer's needs. The difference b/w the two is that requirement analysis is the process of reasoning about the requirements that have been elicited it involves activities such as examining requirements for conflicts or inconsistencies combining related requirements, identifying missing requirements.

Q5 Explain SRS Document in detail (Open
= Solution of SRS)

Soln: The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves goals depending on who is writing it. First the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various system situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectations of the users. The second case SRS, is written for various purposes & serves as a contract document between customer & developer. These are the following features of SRS

- (i) Correctness
- (ii) Completeness
- (iii) Consistency
- (iv) Unambiguity
- (v) Ranking for importance & stability
- (vi) Modifiability
- (vii) Verifiability

Q6. List out the characteristics of SRS Document

- Soln:
- (i) Concise: The SRS report should be small and at the same time, unambiguous, consistent and complete.
 - (ii) Structured: A well structured document is simple to understand and modify.
 - (iii) Black-box view: It should only define what the system should do and refrain from stating how to do these. This means that the SRS document should define the external document behaviour of the system and not discuss the implementation issues.
 - (iv) Conceptual Integrity: This features enables the system to make user merely understand it.
 - (v) Response to undesired events: It should characterize acceptable responses to unwanted events. These are also called system response to exceptional conditions.
 - (vi) Verifiable: All requirements of the system as documented in SRS document should be correct. This means that it should be possible to decide whether or not req have been met in an imp.

module - 3
(Assignment - 2)

DATE: ___/___/___ **B+**
PAGE NO. _____

Q
Ans:

Explain the term Soft design.
Software Design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the client's requirements, as described in SRS document into a form i.e. easily implementable using programming language.

The software design phase is the first step in SDLC (Software Design Life Cycle), which moves the concentration from the problem domain to the solution domain.

Q
Ans:

List out the principles of software design.

Software design principles are concerned with providing means to handle the complexity of the design process effectively.

(i) problem partitioning

(ii) Modularity

(iii) Top down & bottom up strategy.

(iv) Abstraction.

(i) problem partitioning.

For small problem, we can handle the entire program at once but for the significant problem, divide the problems and conquer the problem.

Be Positive...

(ii) Abstraction

An abstraction is a tool that enables a designer to consider a component at an abstract level without bothering about the internal details of the implementation.

(iii) Modularity

Modularity specifies to the division of software into separate modules which are differently named and addressed and are integrated later on to obtain the completely functionality software.

Q3: Explain modularity in reference to SD also explain its properties.

Ans: Modular design reduces the design complexity and faster implementation by allowing parallel development of various parts of a system. we discuss a different section of modular design in detail in this section.

(i) Functional Independence:

Functional independence is achieved by developing functions that perform only one kind of task and do not extensively interact with other modules.

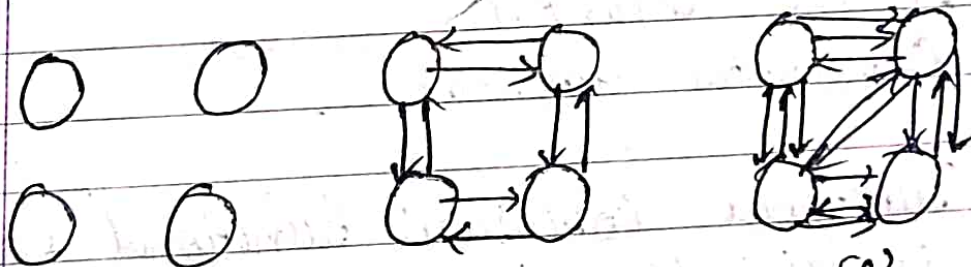
(ii) Information hiding: The fundamental of information hiding suggest that modules

Be Positive...

can be characterized by the design decision that protect from others.

Q4: Describe module coupling with it's levels.

Ans:- In software engineering, the coupling is the degree of interdependence between software modules. Two modules are tightly coupled all strongly dependent on each other. However, two modules that are loosely coupled all not dependent on each other. uncoupled modules have no interdependence at all within them.



(a) uncoupled no-dependence
 (b) Loosely Coupled Some-dependencies
 (c) Highly Coupled

A good design is the one which has low coupling. If coupling is measured by the number of relations between the modules.