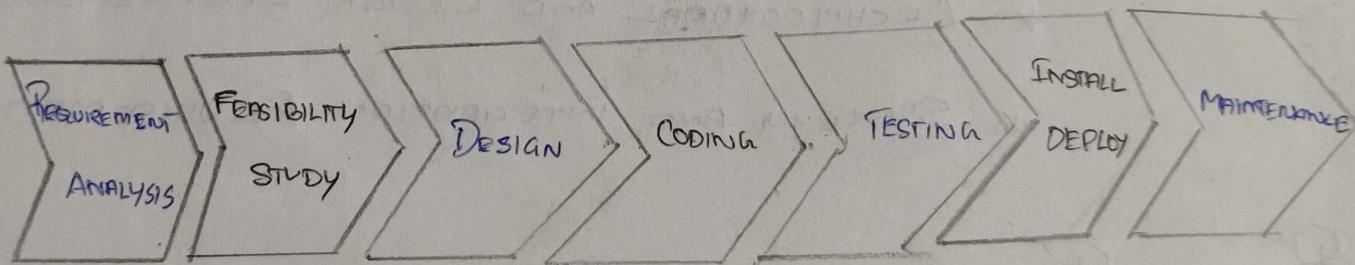


1. EXPLAIN SOFTWARE REQUIREMENT ANALYSIS AND SPECIFICATION?

SOFTWARE REQUIREMENT:

- * Requirements are description of features and functionalities of the target system.
- * Requirements convey the expectations of users from the software product.
- * The goal of requirements engineering is to develop and maintain descriptive 'System Requirements Specification' document.

SDLC PHASES



REQUIREMENT ANALYSIS:

- * The Requirement is the first stage in the SDLC process.

* It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry.

* Software requirement is a functional or non-functional needs to be implemented in the system.

* Functional means providing particular service to the user.

For example: In context to banking application the functional requirement will be when customer selects 'VIEW BALANCE' they must be able to look at their latest account balance.

* Software Requirement can also be a non-functional.

TYPES OF REQUIREMENT

* BUSINESS REQUIREMENT

* ARCHITECTURAL AND DESIGN REQUIREMENTS

* SYSTEM AND INTEGRATION REQUIREMENTS

BUSINESS REQUIREMENT:

Business requirements are high-level requirements, that are taken from the business case.
For example .

COUNTRY	COMPANY PROVIDING BANKING FUNCTIONAL OR SERVICES
INDIA	Account Summary + FUND Transfer
China	Account Summary + Bill Payment

11 (2)

ARCHITECTURAL AND DESIGN REQUIREMENTS: are more detailed than business requirements. It determines the overall design required to implement the business requirement.

SYSTEM AND INTEGRATION REQUIREMENT: are detailed description of each and every requirement.

REQUIREMENT ANALYSIS AND SPECIFICATION:

* The Requirement Analysis and Specification phases starts after the feasibility study stage is complete and the project is financially viable and technically feasible.

* The Requirement Analysis and Specification phases ends when the requirements specification document has been developed and reviewed.

* The Requirements Specification document is usually called the Software Requirement Specification Document.

* The engineers who gather and analyse customer requirements and then write the requirement specification document are known as system analysts in the software industry.

* We can conceptually divide the requirements gathering and analysis activity into two separate tasks.

* Requirement gathering.

* Requirement analysis.

Q. COMPARE COHESION AND COUPLING? EXPLAIN VARIOUS TYPES OF COUPLING TECHNIQUES:

COHESION

* Cohesion is also called Intra-Module Binding.

* Coupling Shows the relationship between modules.

* Cohesion Shows the module's relative functional strength.

* While creating you should aim for the low or high cohesion i.e. cohesive compound module focuses on a single function with little interaction.

* In cohesion, the module focuses on a single thing.

COUPLING

* Coupling is also called Inter-module Binding.

* Coupling shows the relationship between the modules.

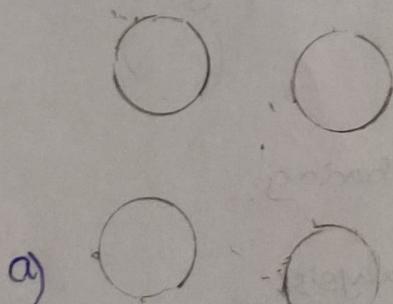
* Coupling shows the relative independence between the modules.

* While creating you should aim for low coupling i.e. dependency among modules should be less.

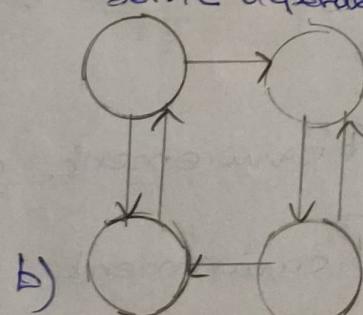
* In coupling, modules are linked to the other modules.

THE VARIOUS TYPES OF COUPLING TECHNIQUES

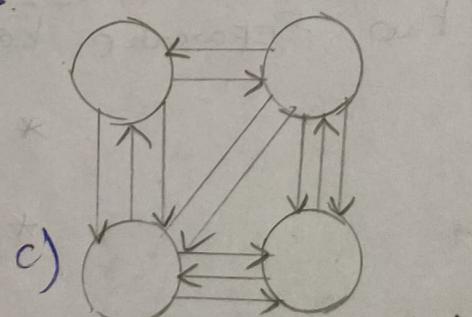
Loosely coupled:
Some dependencies.



Uncoupled: no dependencies



MODULE COUPLING

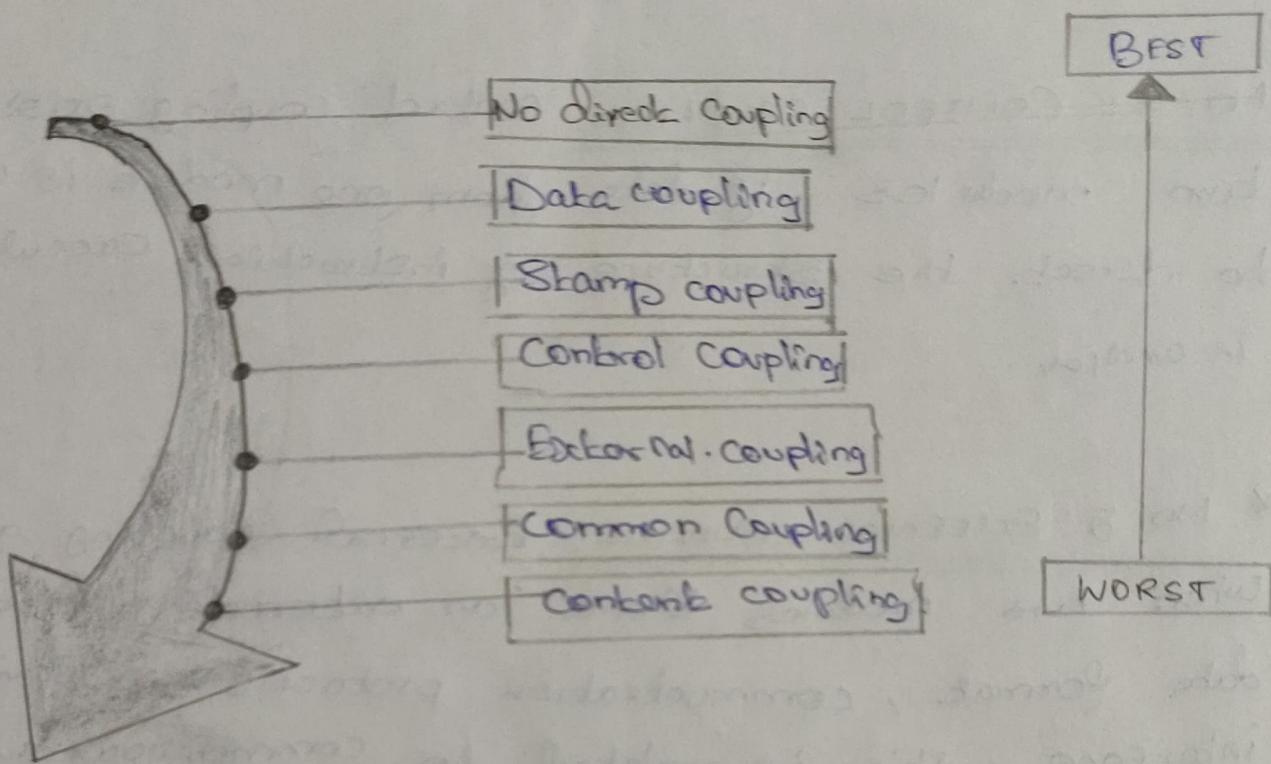


Highly coupled
Many dependencies

TYPES OF MODULES COUPLING

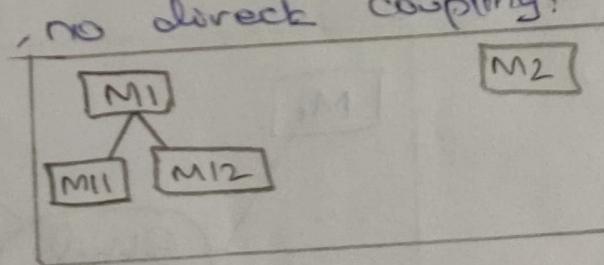
There are various types of module coupling areas

as follows:



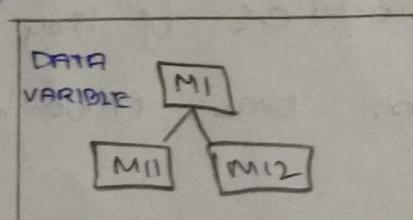
* No: I: No direct coupling: There is a direct coupling between M₁ and M₂.

In case, modules are subordinates to different modules. Therefore, no direct coupling.



* No: 2: DATA COUPLING:

When data of one module is passed to another module, this is called data coupling.

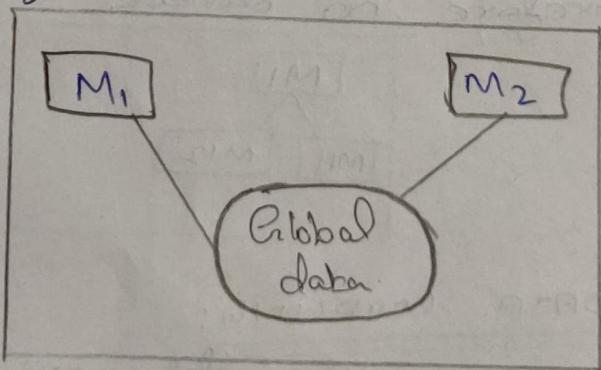


* No: 3: STAMP COUPLING: Two module are stamp coupled if they communicate using composite data items such as Structure, Objects etc.

* No: 4: CONTROL COUPLING: Control coupling exists among two modules if data from one module is used to direct the structure of instruction execution in another.

* No: 5: EXTERNAL COUPLING: External coupling arises when two modules share an externally imposed data format, communication protocols, or device interface. This is related to communication!

* No: 6: COMMON COUPLING: Two modules are common coupled if they share information through some global data items.



* No: 7: CONTENT COUPLING: Content coupling exists among two modules if they share code, eg: a branch from one module into another module.

3. WHAT IS MODULARITY? EXPLAIN ITS ADVANTAGE AND DISADVANTAGES?

MODULARITY Specifies to the division of Software into Separate module which are differently named and addressed and are integrated later on to obtain the completely functional Software. It is the only property that allows a program to be intellectually manageable.

* Single large Programs are difficult to understand and read due to large number of reference variables Controls paths , global variables , etc.

* The desirable properties of a modular system are :

* Each module is a well-defined System that can be used with other applications.

* Each module has single specific objectives

* Modules can be Separately compiled and saved in the library .

* Modules should be easier to use than to build .

* Modules are simpler from outside than inside .

* Software is divided into Separately named and addressable components , called modules , that are integrated to satisfy problem requirements.

* High Cohesion and low Coupling within Components.

ADVANTAGES AND DISADVANTAGES OF MODULARITY

- * It allows programs to be written by several or different people.
- * It encourages the creation of commonly used routines to be placed in the library and used by other programs.
- * It simplifies the overlay procedure of loading a large program into main storage.
- * It provides more check points to measure progress.
- * It provides a framework for complete testing more accessible to test.
- * It produces the well designed and more readable programs.
- * DISADVANTAGES OF MODULARITY
 - * Execution time may be, but not certainly, longer.
 - * Storage size perhaps, but is not certain increased.
 - * Compilation and loading time may be taken.
 - * Inter-module communication problems may be increased.
 - * More linkage required, run-time may be longer, more source lines must be written & more documentation has to be done.

4. EXPLAIN THE PROCESS OF VALIDATION THE REQUIREMENTS SPECIFICATION ?

REQUIREMENT VALIDATION is the process of checking that requirements defined for development, define the system that the customer really wants. To check issues related to requirements, we perform requirement validation.

In the Requirement validation process, we have a different type of test to check the requirements mentioned in the Software Requirement Specification (SRS), these checks include:

- * Completeness checks
- * Consistency checks
- * Validity checks
- * Realism checks
- * Ambiguity checks
- * Verifiability,

* Requirement validation is the process of checking that requirements actually define the system that the customer really wants.

* It overlaps with analysis as it is concerned with finding problems with the requirements.

* The cost of fixing a requirements problem system change is usually much greater than design or coding errors.

* During the requirements validation process, different types of checks should be carried out on the requirement in the requirement document.

These check includes:

1. Validity checks: A user may think that a system is needed to perform certain functions.

2. Consistency checks: Requirements in the documents should not conflict.

3. Completeness checks: The requirement documents should include requirements that define all functions and the constraints intended by the system.

4. Realistic checks: Using knowledge, technology, the requirements should be checked to ensure that they can actually be implemented.

5. Verifiability: To reduce the potential for dispute between customer and contractor, system requirements should always be written so that they are verifiable.

- 114 (2)
- * The output of requirements validation is the list of problems and agreed on actions of detected problems.
 - * The list of problems indicate the problem detected during the process of requirement validation. the list of agreed action把握 the corrective action that should be taken to fix the detected problem.
 - * There are several techniques which are used either individually or in conjunction with other techniques to check the entire or part of the system.

* 1. TEST GENERATION:

- * Requirements mentioned in SRS document should be testable, the conducted tests reveal the error present in the requirement.
- * It is generally believed that if the test is difficult or impossible to design then, this usually means that the requirement will be difficult to implement and it should be reconsidered.

* 2: PROTOTYPING:

* In this validation techniques the prototype of the system is presented before the end-user or customer, they experiment with the presented model and check if it meets their need.

* 3: REQUIREMENT REVIEWS:

* In this approach, the SRS is carefully reviewed by a group of people including people from both the contractor organisations and the client side. The reviewer systematically analyse the document to check.

* 4: AUTOMATED CONSISTENCY ANALYSIS:

* This approach is used for automated detection of an error, such as nondeterministic cases, a type error and circular definition in requirement specifications.

* 5: WALK-THROUGH:

* A walk through does not have a formally defined procedure and does not require a differentiated role assignment.