

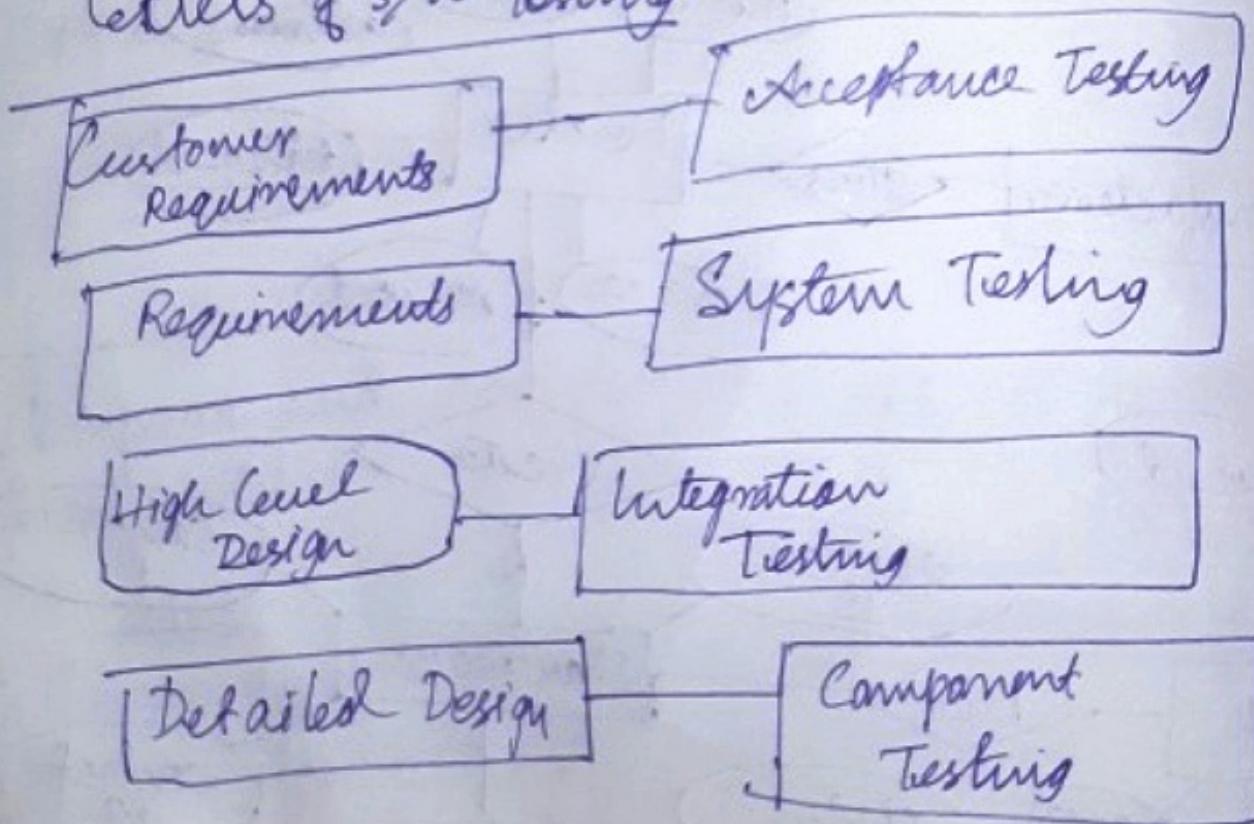
26/4/23

## Testing

What is testing?

Software testing determines when a software system can be released and delivered for future performance. Testing is a major source of feedback and provides a basis for interaction with project stakeholders with the growing complexity of S/W Systems.

Levels of S/W Testing



27/4/23

White Box Testing, Black Box Testing  
In component testing test cases verify the implementation of design of a software element and it traces detailed design.

In integration testing, hardware and software elements are combined and tested until the entire system has been integrated. In system testing it checks for system requirements.

In acceptance testing it determines if test results satisfies acceptance criteria of project stakeholders.

Different types of testing :

i) Black Box Testing ii) White Box Testing or Glass Box

Black Box Testing : It is also known as behavioral testing. This focuses on the functional requirements of the SW. Black Box Testing enables the SW engineer to derive set of input conditions that can fully exercise all functional requirements for a program.

Errors found by Black Box Testing :

- (i) Incorrect or missing functions
- (ii) Interface errors
- (iii) Errors in data structures or external database access
- (iv) Behaviour or performance errors.
- (v) Initialization and termination errors.

(b) White Box Testing : It is also known as glass box. It is a test case design method that uses control flow. It is a test case design method that uses control structure of procedural design to derive test cases. The programmer uses his own understanding and access to the source code to develop test cases.

Benefits

(i) Focused Testing :

The programmer can test the program in pieces. It is much easier to give an individual suspect module and a thorough workout in glass box testing than black box testing. The programmer can also find out which part of the program are exercised by any test.

It is possible to find out which lines of code, which branches or which part have not yet been tested.

(ii) Control flow : The programmer knows what the program is supposed to do next as a function of its current state.

(iii) Data integrity : The programmer knows which part of the program modifies any part of the item of data by tracking a data item through the system, the programmer can spot data manipulation by inappropriate modules.

(iv) Internal Boundaries : The programmer can see IB in the code that are completely invisible to the outside tester.

(v) Algorithm Specific : The programmer can apply standard numerical analysis techniques to predict the results.

Q Difference between Black Box and white Box Testing.

Q testing Method & Types → Difference  
→ Adv & Disadv

## Methods of Testing

4/5/23

- Methods of testing

  - ① Unit Testing (Belongs to white box testing) cost less
  - ② Integration " (" " black box. " ) " high
  - ③ System "
  - ④ Acceptance "  
done by customer to check whether the delivered product perform derived task or not.

(1) Unit Testing What is unit testing?

It is testing a module that

(1) Unit Testing What is Unit testing?

## (ii) Integration Testing

Integration testing

The primary objective of integration testing is to test the module interfaces in order to ensure that there are no errors in the parameter passing, when one module involves another module.

module.  
Q Unit & Integration Testing Difference

Integration Testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. There is often a tendency to attempt non-int incremental integration i.e. to construct a program using a Big Bang approach.

In the incremental or plan integration the program is constructed and tested in small increments, where errors are easier to isolate and correct, interfaces are more likely to be tested completely and a systematic test approach may be applied.

## (ii) System Testing

Software is incorporated with other system elements and a series of system integration and validation tests are conducted. System Testing is actually a series of different tests whose primary purpose is to fully exercise the computerbased system.

There are many tests conducted to ensure assure that it ~~needs~~ meets all its requirements.

Q System Testing & Acceptance Testing difference

Cohesion & Coupling : are two key concepts in Software Engineering that are used to measure the quality of a system design.

Coupling refers to the degree of interdependence between software modules. High coupling means modules are closely connected and changes in one module ~~make~~ may affect other modules.

Low coupling means modules are independent and changes in one module have little impact on other modules.

Cohesion refers to degree to which elements within a module work together to fulfill a single well defined purpose.

While low cohesion means that elements are loosely related and serve multiple purposes.

High cohesion means that elements are closely related on a single purpose.

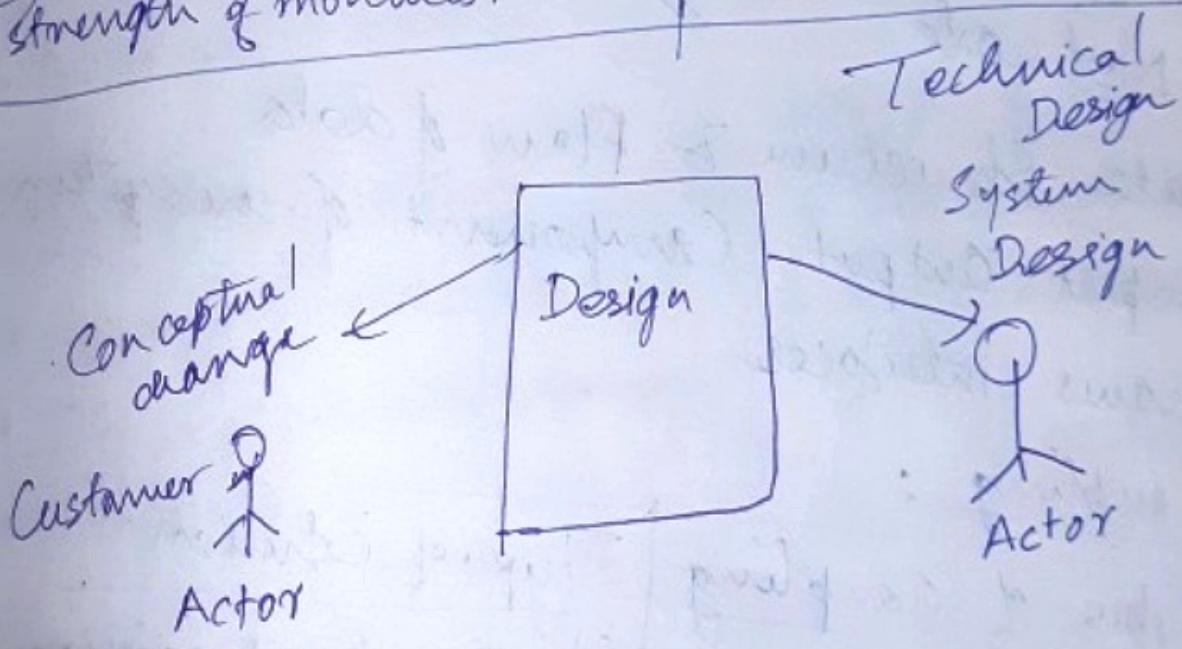
Q) Difference b/w Cohesion & Coupling.

## H.W Cohesion

- (i) refers to degree to which elements within a module work together to fulfil a single, well defined purpose.
- (ii) concept of intra module
- (iii) represents relationship within a module
- (iv) increasing cohesion is good for software.
- (v) represents functional strength of modules.

## Coupling

- (i) refers to degree of interdependence between software modules.
- (ii) concept of inter module
- (iii) represents relationships bet'w modules.
- (iv) increasing coupling is avoided for software.
- (v) represents interdependence among modules.



## Conceptual Design

Software Design has two Design processes.  
Written in simple language—that is understandable  
by the customer.

Detailed explanation about system characteristics. Describe the functionality of the system. It is complemented in ~~independ~~ of implementation. Linked with requirement documents. Hardware component and design. Functionally known of software components.

## Software Architecture Network "

Data Structure & Flow of data  
Input Output Component of the system  
Shows interfaces

Coupling :

### Types of Coupling

- (1) Data Coupling
- (2) Stamp "
- (3) Control "
- (4) External "

### Types of cohesion

- (1) Functional cohesion
- (2) Sequential "
- (3) Communicational "
- (4) Procedural "

5) Common "

6) Content "

(3) Temporal cohesion

(4) Logical "

## Adv & Disadv of Cohesion & Coupling

### Cohesion

#### Adv of high cohesion

- Improved readability & understandability
- Better error isolation

#### Disadv of low cohesion

- Increased code duplication
- Difficulty in understanding the module

### Coupling

#### Adv of low coupling

- Improved maintainability
- Enhanced modularity

#### Disadv of high coupling

- Increased complexity
- Reduced flexibility

15/5/2023

Acceptance Testing : Method done to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

Validation Testing : Ensures that the product actually meets the client's needs. To demonstrate that the product fulfills its intended use when deployed on appropriate environment.

Regression Testing : performed to find out whether the updates or changes had caused new defects in the existing functions.

Smoke Testing : To determine if a new software build is ready for the next testing phase.

Validation Testing are of two types :

(I) Alpha Testing : performed to identify bugs before releasing the product to real users or to the public.

(II) Beta Testing : performed by real users of the software application in a real environment.

Two characteristics of Risk Management are:  
uncertainty & loss

Risk Management : Project Risk & Product Risk

or risk as a portable problem.

Uncertainty Risk management is a sequence of steps that helps a S/a team to understand, analyse and manage uncertainty.

Risk Management has parts :

- (I) Risk identification
- (II) " analysis
- (III) " Planning
- (IV) " Monitoring

Coincidental cohesion : It occurs when the elements within a module have no apparent relationship to one another. It occurs when one existing program is modularized into pieces. If a module is created to save duplicate code by combining some part of code that occurs at many different places that module is likely to have coincidental cohesion.

Logical cohesion : A module have logical cohesion if there is some logical relationship b/w the elements of the module and the elements performed function that fall in the same logical class.

Temporal Cohesion : It is the same as logical cohesion except the elements are also related in the time and are executed together. The modules that perform activities like initialization, pinup and combination are usually temporally bound.

### Procedural cohesion

It contains elements that belong to a common procedural unit. A module with only procedural cohesion may contain only part of a complete function or parts of several functions.

### Communicational cohesion

It has elements that are related by a reference to the same input or output.

### Sequential cohesion

Sequential cohesion of elements occurs when the o/p of one element is the i/p for the next element.

### Functional cohesion

Functional cohesion is the strongest cohesion. If functionally bound module, all the elements of the module are related to performing a single function. Modules of functional cohesion can always be described by a single structure but

sometimes it can also be described  
using compound sentences.

Content Coupling : It occurs when one module modifies logical data blocks or instructions in another module.

Content coupling can occur in assembly language programs. It occurs when branches are made into the middle of the module.

Common Coupling : In this coupling modules are bound together by global data structure. While common coupling, it can be difficult to determine which component is responsible for having set of variables for a particular value.

Central Coupling : When one component passes parameters to control the activity of another component, we can say that there is control coupling between the two.

Stamp Coupling : It is similar to common coupling except that the

Global data items are shared selectively among routines that require the data.

Stamp coupling is more desirable than common coupling because fewer modules will have to be modified if a shared data structure is modified.

### Data Coupling

involves the use of parameters test to the use of parameters to pass data items between ~~super~~ routines.

Data coupling is simpler and leaves less room for errors. If coupling must exist between components data coupling is the most desirable, it is the easiest through which to trace data coupling and to make changes.