SOFTWARE ENGINEERING

ASSIGNMENT-2

Abhishek Tyagi

EN18CS301009

CS-A

Q1. Write a note on black box and white box testing.

Ans: *Black box Testing:*

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications involved.

It is also known as Specifications based testing. An independent *Testing Team* usually performs this type of testing during the software testing life cycle. Black box testing can be done in following ways:

1. Syntax Driven Testing : This type of testing is applied to systems that can be syntactically represented by some language. For example, compilers, language that can be represented by context free grammar.

2. Equivalence partitioning: The idea is to partition the input domain of the system into several equivalence classes such that each member of class works in a similar way.

3. Boundary value analysis: Boundaries are particularly good places for errors to occur. Hence, if test cases are designed for

boundary values of input domain then the efficiency of testing improves and probability of finding errors also increase.

4. Cause effect Graphing: This technique establishes relationship between logical input called causes with corresponding actions called effect. The causes and effects are represented using Boolean graphs.

5. Requirement based testing: It includes validating the requirements given in SRS of software system.

6. Compatibility testing: The test case result not only depend on product but also infrastructure for delivering functionality. When the infrastructure parameters are changed it is still expected to work properly.

- *White box Testing:*

  White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

  Working process of white box testing:

  (a) Input: Requirements, Functional specifications, design documents, source code.

  (b) Processing: Performing risk analysis for guiding through the entire process.

  (c) Proper test planning: Designing test cases to cover entire code. Execute rinse-repeat until error-free software is reached.

  (d) Output: Preparing final report of the entire testing process.

Some of the Testing techniques include:

(a) Statement coverage: In this technique, the aim is to traverse all statement at least once. Hence, each line of code is tested.

(b) Branch Coverage: In this technique, test cases are designed so that each branch from all decision points are traversed at least once.

(c) Conditional Coverage: In this technique, all individual conditions must be covered.

(d) Multiple Condition Coverage: In this technique, all the possible combinations of the possible outcomes of conditions are tested at least once.

(e) Basis Path Testing: In this technique, control flow graphs are made from code or flowchart and then Cyclomatic complexity is calculated which defines the number of independent paths so that the minimal number of test cases can be designed for each independent path.

(f) Loop Testing: Loops are widely used, these are fundamental to many algorithms hence, their testing is important. Errors often occur at the beginnings and ends of loops.

Q2. Define the terms cohesion and coupling.

Ans:

- **Cohesion:** It is a measure of the degree to which the elements of the module are functionally related. It is the degree to which all elements directed towards performing a single task are

contained in the component. A good software design will have high cohesion.

- **Coupling:** It is the measure of the degree of interdependence between the modules. A good software will have low coupling.

Q3. What is meant by software quality management? Explain the difference between ISO and CMMI approach for software quality.

Ans:

- **Software Quality Management:**

It is a process that ensures the required level of software quality is achieved when it reaches the users, so that they are satisfied by its performance. The process involves quality assurance, quality planning, and quality control.

- **Difference between ISO and CMMI approach for software quality:**

(a) The fundamental difference between CMMI vs ISO is conceptual. CMMI is a process model and ISO is an audit standard.

(b) CMMI is rigid and extends only to businesses developing software intensive systems. ISO is flexible and applicable to all manufacturing industries.

(c) CMMI approaches risk management as an organized and technical discipline by identifying risk factors, quantifying such risk factors, and tracking them throughout the project life cycle. ISO was until recently neutral on risk management.

(d) Neither CMMI nor ISO requires the establishment of new processes. CMMI compares the existing processes to industry best practices whereas ISO requires adjustment of existing processes to confirm to the specific ISO requirements.

Q4. Differentiate between verification and validation.

Ans:

Verification	Validation
Verification is the process of evaluating products of a development phase to find out whether they meet the specified requirements.	Validation is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements.
Execution of code does not come under verification.	Execution of code comes under validation.
Verification is carried out before the validation.	Validation is carried out just after the verification.
Plans, Requirement Specifications, Design Specifications, Code, Test Cases etc., are evaluated during verification.	Actual product or Software is evaluated during validation.
Cost of errors caught in verification is less than errors found in validation.	Cost of errors caught in validation is more than errors found in verification.

Q5. Define the following terminologies related to design concept:

I. Abstraction

ii. Modularity

iii. User interface design

Ans:

(a) Abstraction: It simply means to hide the details to reduce complexity and increases efficiency or quality. Different levels of Abstraction are necessary and must be applied at each stage of the design process so that any error that is present can be removed to increase the efficiency of the software solution and to refine the software solution.

(b) Modularity: It simply means to divide the system or project into smaller parts to reduce the complexity of the system or project. Modularity in design means to subdivide a system into smaller parts so that these parts can be created independently and then use these parts in different systems to perform different functions.

(c) User interface design: User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction. The UI can be graphical, text-based, audio-video based, depending upon the underlying

hardware and software combination. UI can be hardware or software or a combination of both.