UNIT - 4

CHAPTER 4

Software Verification and Validation

Syllabus Topic : Introduction

4.1 Introduction

Q. 4.1.1 Explain software verification and its features. (Ref. Secs. 4.1, 4.1.1 and 4.1.2) **(5 Marks)**

- Software verification and validation is a vital model that allows the team of software developers and testers to ensure the accurate development of product throughout the development life cycle.
- Both software verification and validation help create a software product that efficiently meets the specified requirements of the customer and offers them optimum services.

Syllabus Topic : Verification

4.1.1 What is Software Verification?

Q. 4.1.2 Explain software verification and its features. (Ref. Sec. 4.1, 4.1.1 and 4.1.2) **(5 Marks)**

- Verification is the process of checking or verifying the credentials, data or information to confirm their credibility and accuracy. In the field of software engineering, software verification is defined as the process of evaluating software product, to ensure that the development phase is being carried out accurately, to build the desired software product.
- It is performed during the ongoing phase of software development, to ensure the detection of defects and faults in the early stage of the development life cycle and to determine whether it satisfies the requirements of the customer.
- Software verification offers answers to our query of **"Are we building the software product in a right manner?"** However, if the software passes during the verification process, it does not guarantee its validity. It is highly possible that a software product goes well through the verification process, but might fail to achieve the desired requirements.

4.1.2 Features of Software Verification

Q. 4.1.3 Explain software verification and its features. (Ref. Sec. 4.1, 4.1.1 and 4.1.2) **(5 Marks)**

- Software verification plays an important role in building the product as per the requirements and needs of the customer. Other features of software verification are :
  - Performed during the early stages of the software development process to determine whether the software meets the specified requirements.
  - Verification denotes precision of the end or final product.
  - It conducts software review, walk through, inspection, and evaluates documents, plans, requirements, and specifications.
  - It demonstrates the consistency, completeness, and correctness of the software during each stage of the software development life cycle.
  - Software verification can be termed as the first stage of the software testing life cycle (STLC).

Syllabus Topic : Verification Workbench

4.1.3 Verification Workbench

Q. 4.1.4 Explain verification workbench. (Ref. Sec. 4.1.3) **(5 Marks)**

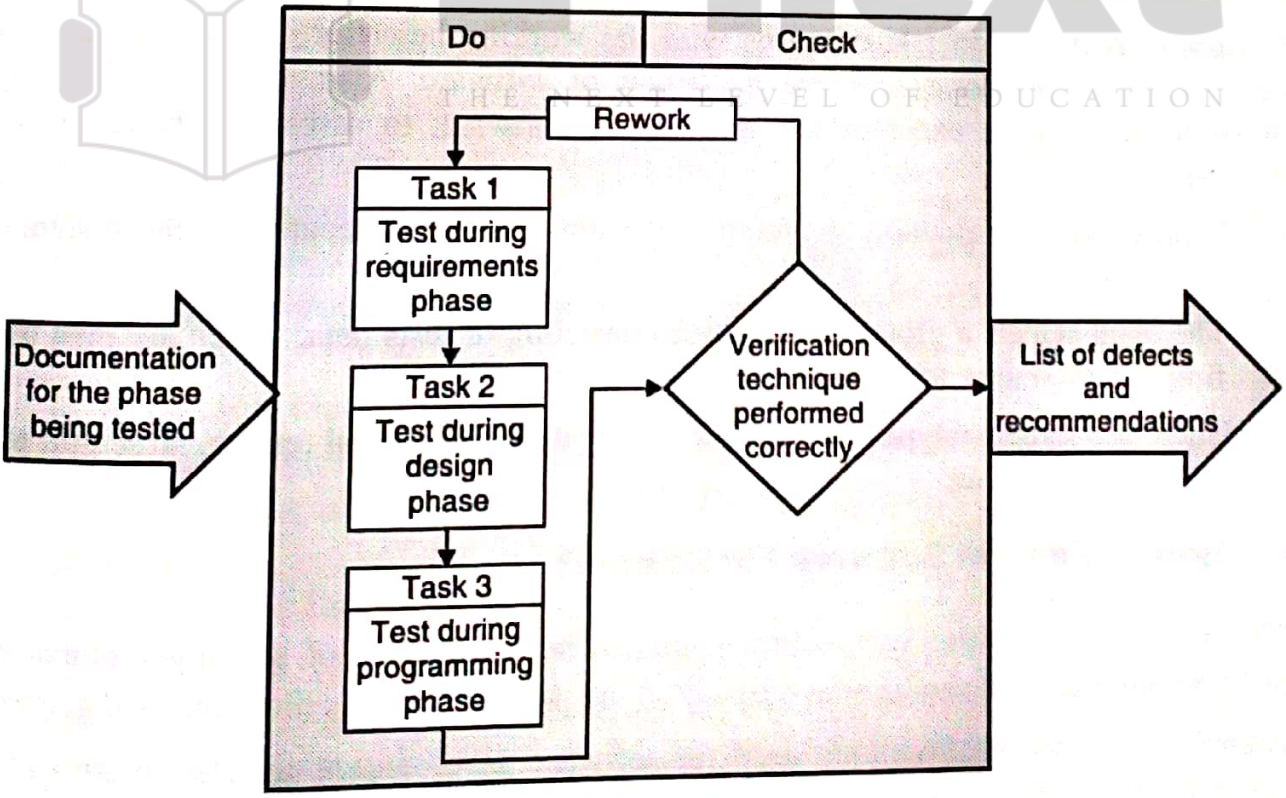

Fig. 4.1.1

- For the phase being tested, the documentation made by the development team is the input to the workbench. An appropriate verification technique will be performed towards the end of the requirements, design and programming phases.

- To check whether the verification techniques have been performed correctly, few quality control procedures are designed.
- Testers should list the discovered bugs at the end of each development phase test. The workbench for performing verification testing is shown in the Fig. 4.1.1.

Syllabus Topic : Methods of Verification

4.1.4 Methods of Verification

Q. 4.1.5 What are the different methods of verification? (Ref. Sec. 4.1.4) (5 Marks)

Software verifications are of two types, Static and Dynamic :

1. **Static Verification :** Static verification involves inspection of the code before its execution, which ensures that the software meets its specified requirements and specifications.
   - It is an analysis based approach, usually carried out by just making an analysis of static aspects of the software system, such as the code conventions, software metrics calculation, anti-pattern detection, etc.
   - It does not involve the operation of the system or component.
   - Static verification includes both manual as well as automated testing techniques, such as consistency techniques and measurement techniques.
2. **Dynamic Verification :** It concerns with the working behavior of the software and is being carried out along with the execution of software. Also, know by 'Test phase' dynamic verification executes test data on the software, to assess the behavior of the software.
   - Unlike static verification, dynamic verification involves execution of the system or its components.
   - The team selects a group of test cases consisting of tests data, which are then used to find out the output test results.
   - There are three subtypes of dynamic verification: functional testing, structural testing and random testing.

4.1.5 How to Perform Software Verification?

- The process of software verification involves the assessment of the development phase and intermediary software product, based on the pre-defined specifications and guidelines.
- It ensures that the methodology used for software development adheres to the specified specifications, set-up before the development phase, to build the correct software product.
- Thus, this process requires cross-checking of these pre-defined specifications with the partially developed software product, to satisfy the process of carrying out the development of the product, in a right way.

Syllabus Topic : Types of Reviews on the basis of Stage Phase

4.2 Types of Reviews on the Basis of Stage Phase

Q. 4.2.1 Explain the different types of reviews on the basis of Stage phase. (Ref. Sec. 4.2) **(5 Marks)**

Specific names are assigned to the review techniques, as follows :

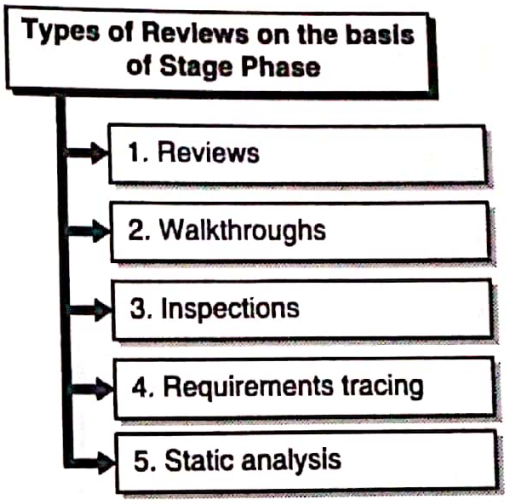

Fig. 4.2.1

1. Reviews

- Peers and/or stakeholders challenge the correctness of the work being reviewed by a formal process called Review. The correctness of requirements is challenged in a requirement review, for example.
- Based on the experience of the organization, review is a formal process, and uses a predetermined set of questions to accomplish the objectives of the review.

2. Walkthroughs

- In this informal process, the peers and other stakeholders talk with the project personnel and ensure that the best possible project is implemented.
- If the project team is unsure that they have resolved issues in the most effective and efficient manner, then they request for a walkthrough.

3. Inspections

- **It is a formal process of review. The main objective is to check that the** entrance criteria for a specific workbench were correctly implemented into the exit criteria.
- This process thoroughly traces the entrance criteria to the exit criteria and ensures that nothing is missing or wrong.

4. Requirements tracing

- This process ensures that requirements are not lost during implementation. The requirements are uniquely identified after they are defined.
- They are then traced step by step to ensure that all the requirements have been processed correctly.

5. Static analysis

- Software is used to perform static analysis. For example, there is a static analyzer in most of the source code compilers that provides information about the correctness of the source code preparation.
- These are the techniques that are incorporated into either the verification process of requirements, design, or programming the software. Some of the techniques can be used in combination with the other. As an example, a review can be coupled with requirements tracing and so on.

Syllabus Topic : Entities Involved in Verification

4.2.1 Entities Involved in Verification

Q. 4.2.2 Which are the various entities involved in verification? (Ref. Sec. 4.2.1) **(5 Marks)**

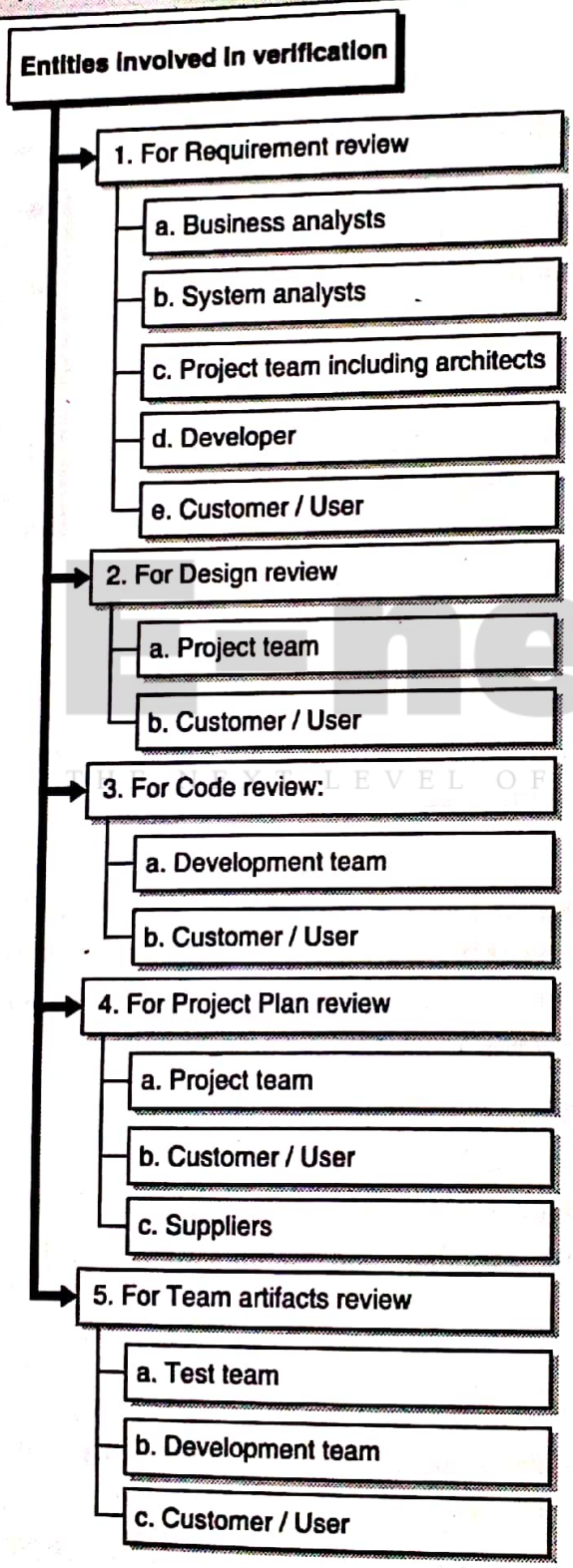

Fig. 4.2.2

Syllabus Topic : Reviews In Testing Lifecycle

4.2.2 Reviews In Testing Lifecycle

Q. 4.2.3 Which are the different reviews in software testing life cycle (STLC) ? (Ref. Sec. 4.2.2) **(5 Marks)**

- Software testing has its own life cycle named as the Software Testing Life Cycle (STLC)
- Reviews associated in the STLC are as follows :

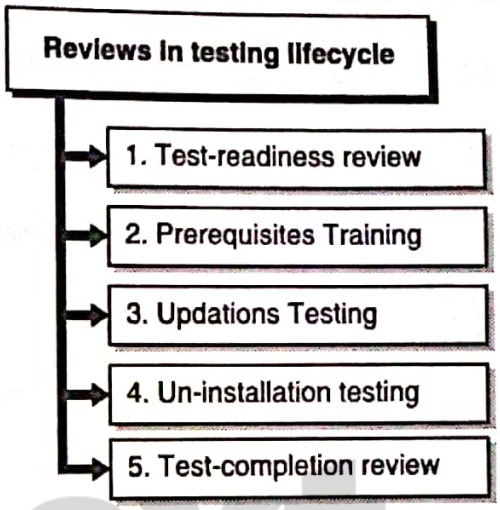

Fig. 4.2.3

➜ 1. Test-readiness review

- It is conducted by test managers or test leads with the project manager to ensure that the product under development is ready for testing as per the phase of testing available.
- It is carried out at every stage of development (Unit testing, integration testing, system testing, interface testing, system testing, acceptance testing, etc.)

➜ 2. Prerequisites Training

- Prerequisites such as the database, operating system, etc. may be needed during the software installation.
- If it is specified in the requirement statement that installation must check for the availability of such prerequisites, then it must be tested for these requirements.

➜ 3. Updations Testing

- During software installation, it is expected that the system should check whether the same software already exists there or not.
- If the existing version of the software is older than the one being installed, then it also prompts for a repair or new installation.

➜ 4. Un-installation testing

- This is done by the organizations to verify if the un-installation is clean or not.
- Whenever any application is un-installed, all the associated files should be removed from the disk.
- Care should be taken that applications running before installation must be running after installation and also once un-installation is done.

➜ **5. Test-completion review**

- It is conducted when a testing cycle is completed.
- It recommends about the status of an application.
- It indicates the development team whether a product can go to the next phase or if it needs any repairing before it can be declared successful.

Syllabus Topic : Coverage in Verification

4.2.3 Coverage in Verification

Q. 4.2.4 Explain coverage in verification. (Ref. Sec. 4.2.3) **(5 Marks)**

☞ **Coverage offered during verification**

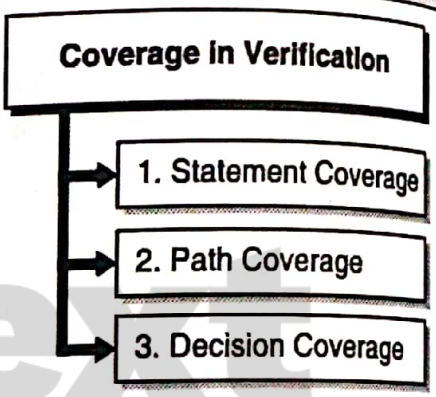

Fig. 4.2.4

➜ **1. Statement Coverage**

- Code includes many Statements.
- Some people consider ";" as a sign of statement, while others consider a code of line with or without ";" as a statement. These definitions must be used consistently for every separate project.
- Statement coverage can be represented as, the number of lines verified against the total number of lines available.
- If all lines of code are verified, then coverage is said to be 100%.

➜ **2. Path Coverage**

- The sequence of control flow from entry to exit is called the Path in a given code.
- An application can have a single path or multiple paths.
- If each path is considered for verification, then it may give a 100% path coverage.

➜ **3. Decision Coverage**

- At several places, when an application is getting executed, it may have to make decisions based upon the situation it faces.
- There might be several paths for each decision.
- Decision coverage is less than 100%, if one outcome is verified with an assumption that all other outcomes will be correct and if the one selected is correct.

Syllabus Topic : Concerns of Verification

4.2.4 Concerns of Verification

Q. 4.2.5 List various concerns of verification. (Ref. Sec. 4.2.4) (5 Marks)

- Testers should have the following concerns when selecting and executing verification testing :
  - Assurance that the best verification techniques will be used.
  - Assurance that the verification technique will be integrated into a developmental process.
  - Assurance that the right staff and appropriate resources will be available when the technique is scheduled for execution.
  - Assurance that those responsible for the verification technique are adequately trained.
  - Assurance that the technique will be executed properly.

Syllabus Topic : Validation

4.3 Validation

Q. 4.3.1 Explain validation. (Ref. Sec. 4.3) (5 Marks)

Q. 4.3.2 Differentiate between verification and validation. (Ref. Sec. 4.3) (5 Marks)

- Validation is an actual testing performed on the software product. It gives answer to our query of ***"Are we developing the right software product?"*** Further, it also ensures the identification of defects that were got missed during the verification process.
- As stated earlier, validation is the actual testing performed on the software product. Thus, it requires proper testing strategy. According to ISO/IEC 12207:2008, validation process includes :
  - Gathering and analysis of the specifications and requirements.
  - Based on specifications and requirements, preparation of test strategies, plans and cases, that seems fit for use.
  - Go for testing the boundary values along with stress and functionalities test.
  - Test the error message.
  - Conducts software evaluation, as it ensures that the software meets the all pre-decided requirements and is acceptable for use.

4.3.1 Difference between Software Verification and Validation

Software verification and validation are two of the most important processes used for ensuring the quality and accuracy of the product. Here is a comparison between these two software testing techniques.

Sr. No.	Software Verification	Software Validation
1.	Software verification is the process of evaluating the development phase to ensure the accuracy of the software.	Software validation evaluates the software during the development phase to ensure its compliance with business requirements.
2.	It is performed before the execution of software validation.	It is performed once the process of software verification is completed.
3.	Verification ensures the product is being developed as per the requirements and design specification.	Validation ensures that the software meets the user's needs and is fit to be used by them.
4.	Evaluates plans, requirement specification, documents, test cases, etc.	Evaluates the software product to verify its readiness to be released.

Syllabus Topic : Validation Workbench

4.3.2 Validation Workbench

Q. 4.3.3 Explain validation workbench. (Ref. Sec. 4.3.2) (5 Marks)

All validation activities on a work product are conducted in the workbench.

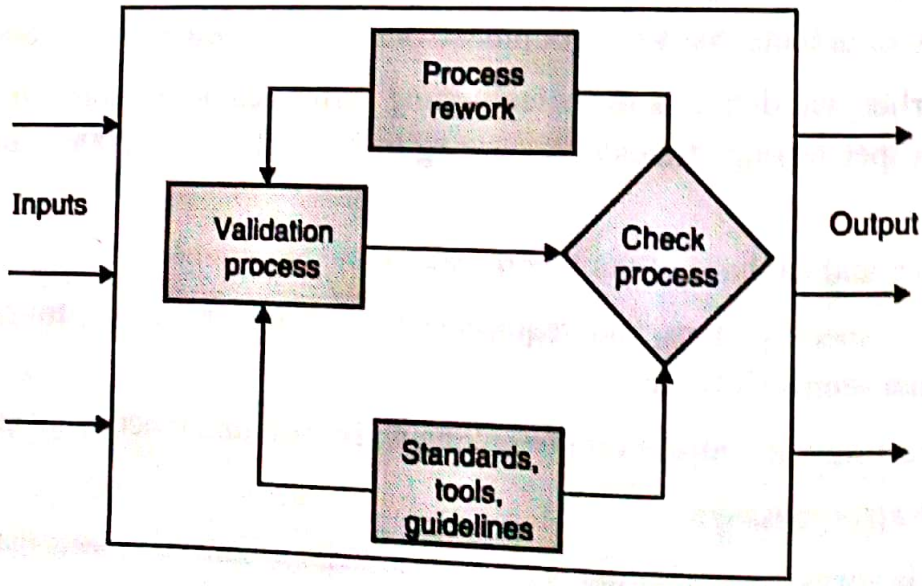

Fig. 4.3.1

- **Basic requirements**

Inputs

While the inputs are entering workbench, there has to be some entry-criteria definition. As an example, suppose we are validating an application or some part of it, we should have a code written and compiled as an input to the workbench along with a test plan as per definition of input.

Output

The input criteria for the next workbench should match with some exit criteria from the workbench. Validated work products, defects, etc. can be included in the output.

- **Validation process**

Sequential activities to be conducted in a workbench are described by the validation process. The activities done while validating the work product under testing are described in the validation process.

- **Check process**

It validates the process and not the work product. It describes how the validation process is checked.

- **Standards, tools, guidelines**

These may be referred to as the tools available for validation. There may be testing standards available. At times, checklists are also used for doing validation.

Syllabus Topic : Levels of Validation

4.3.3 Levels of Validation

Q. 4.3.4 What are the different levels of validation? (Ref. Sec. 4.3.3) (5 Marks)

Validation can be applied at different stages of software development. Following are the stages :

- **Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design - the software component or module.

- **Unit Test Considerations**

The module interface is tested to ensure that information properly flows into and out of the program unit under test.

- The local data structure is checked to ensure that the data stored temporarily retains its Integrity during all steps in an algorithm's execution.
- Boundary conditions are tested to make sure that the module operates properly at boundaries established to restrict processing.
- All independent paths (basis paths) through the control structure are exercised to ensure that all statements in a module have been executed at least once.
- And finally, all error handling paths are tested.

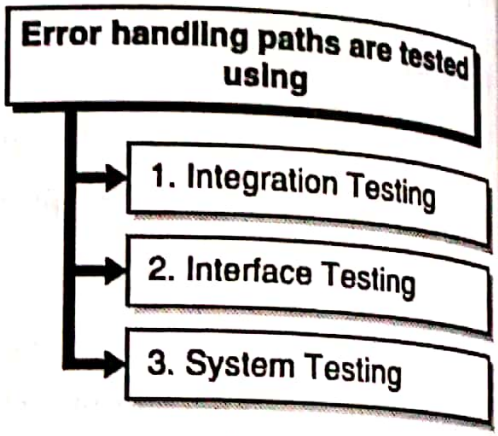

Fig. 4.3.2

1. Integration Testing

- Integration testing is a technique used for constructing the program structure while at the same time carrying out tests to uncover errors related with interfacing.
- The main objective is to take unit tested components and build a program structure that has been dictated by design.

2. Interface Testing

- When an application or a software or a website is developed, then there are several components of it. Those components can be server, database etc.
- The connection which integrates and facilitates the communication between these components is termed as an Interface.
- In simple terms, an interface is software comprising of a set of commands, messages etc.
- **The testing that is done to verify the interface functionality is called Interface testing.**

3. System Testing

- System testing of software or hardware is testing conducted on a whole, integrated system to estimate the systems compliance with its specified set of requirements.
- The only things Tester should be testing at the System Test stage are things that he and she couldn't test before.
- It is done to check how the system as a whole is functioning. Here the functionality and performance of the system is validated.
- User is not involved in System Testing.

Syllabus Topic : Coverage in Validation

4.3.4 Coverage in Validation

Q. 4.3.5 Explain the coverage offered during validation. (Ref. Sec. 4.3.4) **(5 Marks)**

Following are some of the coverage offered during validation:

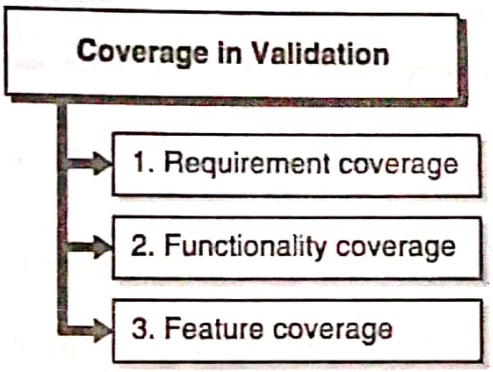

Fig.4.3.3

➜ 1. Requirement Coverage

Typical requirement coverage definition is shown below:

Priority of requirement	Coverage offered
P1 (Must)	100%
P2 (Should be)	50%
P3 (Could be)	25%

All the requirements need not be necessarily covered. They may be segregated into separate classes such as 'Must', 'Should be" and "Could be'. These are prioritized accordingly. Almost all high-priority requirements expressed as "Must' and some lower priority requirement expressed as "Should be' and 'Could be' should be covered.

➜ 2. Functional Coverage

At times, requirements are expressed in the functionality required for the application to work correctly. As per the importance to users, functionalities are also defined at different priorities. Thus, the coverage can be expressed as follows :

Priority of functionality	Coverage offered
P1 (Must)	100%
P2 (Should be)	50%
P3 (Could be)	25%

➜ 3. Feature Coverage

It describes about covering a feature required by the user. Feature may be a group of functionalities doing similar things. A typical definition of feature coverage is mentioned below :

Priority of feature	Coverage offered
P1 (Must)	100%
P2 (Should be)	50%
P3 (Could be)	25%

Syllabus Topic : Acceptance Testing

4.4 Acceptance Testing

Q. 4.4.1 Explain acceptance testing in detail. (Ref. Sec. 4.4) **(5 Marks)**

- Acceptance testing is one of the last types of software testing performed over a software or application.
- It is conducted by a pool of targeted users to ensure the readiness and quality of the system from user's perspective, which allows the team to meet their needs and expectations.
- The main goal behind acceptance testing is to check whether the developed software product passes the acceptance norms defined on the basis of user and business requirements, so as to declare it acceptable or non-acceptable for its use by the users.

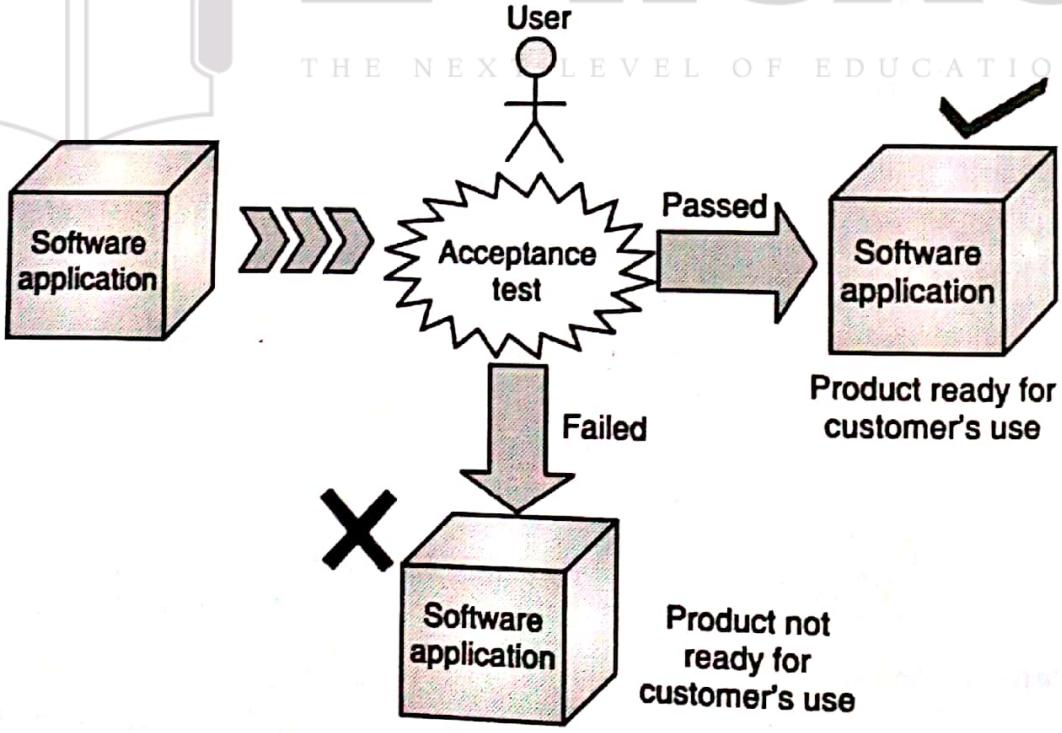

Fig. 4.4.1

- Acceptance testing is also referred to as **red box testing**. Additionally, the acceptance tests are derived from the user story and are based on the acceptance criteria, which are defined on the basis of the following :

- Correctness and completeness of the functionality.
- Data integrity and data conversion.
- Software usability, performance and scalability.
- Documentation.
- Timeliness.
- Product's confidentiality and availability.
- Install-ability as well as upgrade-ability.

4.4.1 Types of Acceptance Testing

Broadly, there are two main types of acceptance testing as shown in Fig. 4.4.2

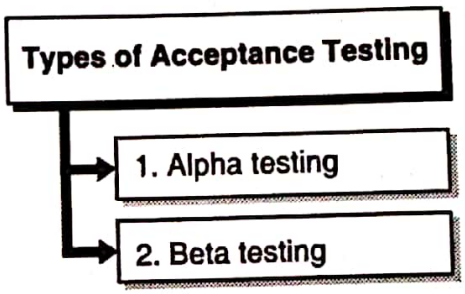

Fig. 4.4.2

➜ **1. Alpha testing**

- It is an on-site acceptance testing executed at developer's site usually by the group of skilled testers.
- This is carried out at the end of the software development process and before the handing over of the software product to clients/users.
- The feedback or the collected result from the alpha testing is significantly used to fix bugs or defects, and it is feasible to improve the usability of the product.
- **Explore more of this testing in our article Alpha Testing.**

➜ **2. Beta testing**

- Also known by the name **field testing**, beta testing is an off-site acceptance testing, carried out at client's site by the stakeholder or the end-users.
- This testing involves the evaluation of the software in the real environment by its potential users before its actual release in the market.

Syllabus Topic : Management of Verification and Validation

4.4.2 Management of Verification and Validation

Q. 4.4.2 Explain management of verification and validation. (Ref. Sec. 4.4.2) (5 Marks)

- Management tasks for V and V span the entire life cycle. These tasks are to :
  - plan the V and V process;
  - coordinate and interpret performance and quality of the V and V effort;

- report discrepancies promptly to the user or development group;
- identify early problem trends and focus V and V activities on them;
- provide a technical evaluation of the software performance and quality at each major software program review (so a determination can be made of whether the software product has satisfied its requirements well enough to proceed to the next phase); and
- assess the full impact of proposed software changes.

– The output of the V and V activities consists of the Software Verification and Validation Plan (SVVP), task reports, phase summary reports, final report and discrepancy report. Major steps in developing the V and V plan are to :

- Define the quality and performance objectives (e.g., verify conformance to specifications, verify compliance with safety and security objectives, assess efficiency and quality of software, and assess performance across the full operating environment).
- Characterize the types of problems anticipated in the system and define how they would show up in the software.
- Select the V and V analysis and testing techniques to effectively detect the system and software problems.

– The plan may include a tool acquisition and development plan and a personnel training plan. The SVVP is a living document, constantly being revised as knowledge accumulates about the characteristics of the system, the software, and the problem areas in the software.

Syllabus Topic : Software Development Verification and Validation (V and V) Activities

4.4.3 Software Development Verification and Validation (V and V) Activities

Q. 4.4.3 Explain software development verification and validation activities. (Ref. Sec. 4.4.3) **(5 Marks)**

– Software V and V comprehensively analyzes and tests software during all stages of its development and maintenance to:

- determine that it performs its intended functions correctly,
- ensure that it performs no unintended functions, and
- measure its quality and reliability.

– Software V and V is a systems engineering discipline which evaluates the software in a systems context, relative to all system elements of hardware, users and other software.

- Like systems engineering, it uses a structured approach to analyze and test the software against all system functions and all hardware, user, and other software interfaces.
- Software quality depends on many attributes, (e.g., correctness, completeness, accuracy, consistency, testability, safety, maintainability, security, reusability). Each organization involved in the software development process contributes to the building of quality of the software.
- Few important activities are :
  - Uncovering high risk errors early, giving the design team time to evolve a comprehensive solution rather than forcing them into a makeshift fix to accommodate software deadlines.
  - Evaluating the products against system requirements.
  - Providing management with visibility into the quality and progress of the development effort that is continuous and comprehensive, not just at major review milestones (which may occur infrequently).
  - Giving the user an incremental preview of system performance, with the chance to make early adjustments.
  - Providing decision criteria for whether or not to proceed to the next development phase.

Syllabus Topic : V Test Model-Introduction

4.5 V Test Model

Q. 4.5.1 Explain the V Test model in detail. (Ref. Sec. 4.5) (5 Marks)

4.5.1 Introduction

- Testing is a lifecycle activity. It begins when the proposal of software development is made, and ends only when application is accepted by the customer/end user.
- For every development activity, there is a testing activity attached with it, so that the phase achieves its milestone deliverable with minimal issue.
- Every phase of software development activities must consider corresponding testing activity associated with it. Project plan and estimations for a project must consider all the activities of testing (verification and validation) along with development activities corresponding to different phases.
- The following diagram below shows the details of software lifecycle development activities, and software life cycle testing activities attached with them.

Syllabus Topic : V-Model for Software

4.5.2 V-Model for Software

Q. 4.5.2 Explain V-model for Software. (Ref. Sec. 4.5.2) **(5 Marks)**

- Validation model explains the validation activities associated with different phases of software development.
- Initially, during the requirement phase, there is system testing and acceptance testing, where system testers and users confirm that requirements have been met or not.
- Design phase is associated with interface testing which covers design specification testing as well as structural testing.
- Program-level designs are associated with integration testing.
- At the code level to validate individual units, unit testing is done.

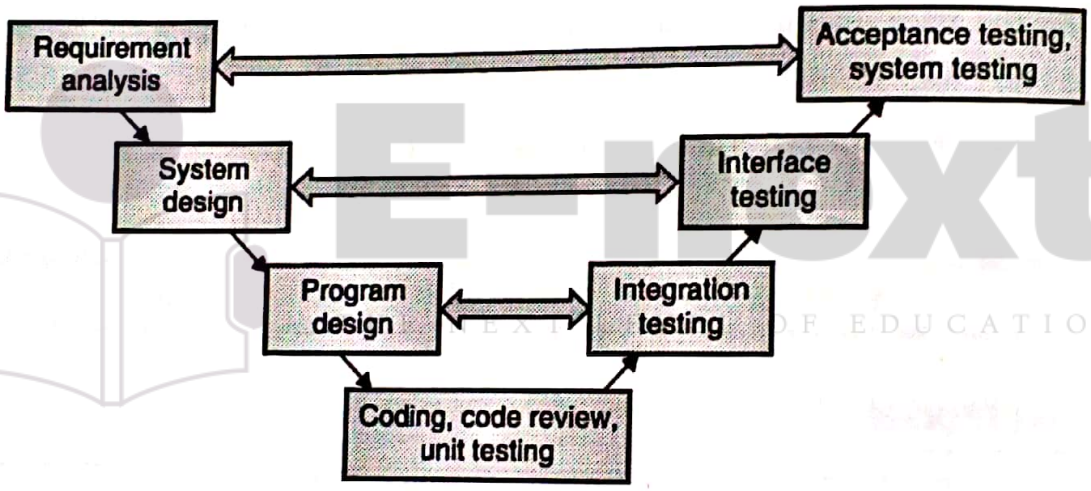

Fig. 4.5.1

Syllabus Topic : Testing During Proposal Stage

4.5.3 Testing During Proposal Stage

Q. 4.5.3 Why is testing required during proposal stage? (Ref. Sec. 4.5.3) **(5 Marks)**

Requirement statement development begins with system proposal. A proposal is created when the customer asks for information, quotation, proposal, etc. At the proposal stage, system description may not be fully clear.

People use various approaches such as formal/informal proof of concept, modeling, prototyping, etc. The success of all these approaches during proposal stage lies in successfully defining the problem and proposed solution.

- Feasibility study may or may not be a part of the proposal. Sometimes, conducting feasibility study also needs approval as it may involve some cost and effort, and customer may not want people from other organizations to understand things completely.
- Feasibility study is done by the customer as well as the supplier in search of a possible solution/approach to solve the problem faced by the customer.

Syllabus Topic : Testing During Requirement Stage

4.5.4 Testing During Requirement Stage

Q. 4.5.4 Explain testing during requirement stage. (Ref. Sec. 4.5.4) (5 Marks)

- This stage must cover all the requirements for the system. Verification of problem definition and requirements definition is the basis on which the system requirement specification is developed further.

Characteristics of good requirements

☞ **Adequate**

- The requirements must describe the entire system covering the end-to-end scenario from the user's side.
- All the assumptions and limitations in the system must be defined and documented with possible answers. Requirements must not talk about any impossible thing happening.

☞ **Clear**

- Customer expectation must be reflected in requirements very clearly. Requirements must be crystal clear to the architect, designer as well as developer.
- There should not be any query about the outcome of system working, both to the customer as well as organization developing the project.

☞ **Verifiable and Testable**

- The requirements should be both verifiable and testable. The requirement statement is used for defining functional and structural test scenarios and test cases.
- It must be used for defining functional as well as structural test data and test events.

Syllabus Topic : Testing During Test-Planning Phase

4.5.5 Testing During Test-Planning Phase

Q. 4.5.5 Explain testing during test-planning phase. (Ref. Sec. 4.5.5) (5 Marks)

- This phase includes defining a test strategy for the given application, writing test plan, test scenario, test cases, and test data.

- The test manager is responsible for defining test strategy, while the test lead develops a test plan to incorporate test strategies, define schedules, methods of testing, evaluation criteria to declare the outcome of testing, and define test-team approach with roles and responsibilities and structure for the system under testing.
- Testing artifacts must be reviewed for their consistency. Verification of testing artifacts may include the following :

Generate Test Plan to Support Development Activities

- Test plan should be consistent with the application development methodology, schedule, and deliverables.
- It must describe respective verification and validation activities to be conducted during each phase of software development life cycle.

Generate Test Cases Based on System Structure

- Functional test cases must be based on functional requirements, while the structural test cases must be defined on the basis of design and non-functional requirements of the system.
- It should define test data using different techniques available such as boundary value analysis , equivalence partitioning, error guessing. and state transition. Test cases and test data must be derived from test scenario.

Syllabus Topic : Testing During Design Phase

4.5.6 Testing During Design Phase

Q. 4.5.6 Explain testing during design phase. (Ref. Sec. 4.5.6) **(5 Marks)**

- Design is the backbone of any software application. Only a successful design can convert the requirements into a good application.
- Design may be made by system architects or designers, and reviewed and approved by the project manager and/or customer.
- Design must reflect the requirements correctly. Verification and validation of design may include the following.

Consistency with respect to Requirements

- Designs must be consistent with requirements defined. Often, the customer expectations as defined in requirements may contradict with each other, and one must perform trade-off.
- If there is any trade-off between the requirements for implementation, it should be mentioned clearly in design, and customer approval is mandatory.

Analyze Design for Errors

- Any design is implemented by coding. Errors in the design will directly reflect the errors in coding and application so developed.
- Hence, consistency between design elements must be maintained to avoid any mismatches.

Syllabus Topic : Testing During Coding

4.5.7 Testing During Coding

Q. 4.5.7 Explain testing during coding phase. (Ref. Sec. 4.5.7) **(5 Marks)**

- Coding is the most vital stage in software development, where product is actually built.
- It is important for the organization to establish verification/validation of a code so that the product complies with the requirement specifications.

Aspects to be Checked

Coding Standards Implementation

- Organizations have coding standards defined which are used in the coding phase of product development. When the code files are written, the developers must use these standards/guidelines.
- While reviewing code, the peer must make sure that standards and guidelines are implemented correctly.

Coding Optimization

- Coding standards must also talk about optimization of code. It talks about how nesting must be done, how declaration of variables and functions must be done, how reusable components must be used and so on.
- Peer review must verify that the written code is properly adhering to optimization guidelines.

Unit Testing

- Unit testing should be done by the developers to ensure that written code is working as expected.
- Defects in unit testing must be logged, and correction, as well as corrective/preventive actions, must be taken to close them.

Syllabus Topic : VV Model

4.6 VV Model

Q. 4.6.1 Explain the VV model in detail. (Ref. Sec. 4.6) **(5 Marks)**

- 'VV model' describes verification and validation activities associated with software development during the entire lifecycle.
- Various activities associated with each phase of the software development lifecycle:

☞ **Requirements**

- As requirements are obtained from the customer using various techniques, there must be some arrangement for conducting requirement review.
- The intention would be to check if there is any gap existing between user requirements and requirement definition.

☞ **Requirement Verification**

- The requirement verification is carried out through inspection of requirement specification document using checklist, standards or guidelines.
- Outcome of the inspection must be recorded and defects must be corrected before going further. Trade-offs, if any, must be documented.

☞ **Requirement Validation**

- Requirement validation occurs at two or more stages during software development.
- The first stage of validation involves writing complete use cases by referring to requirement statement. The second stage of validation is through system testing.

4.7 Design

High level designs are created by architects and low level designs by the designers.

☞ **Design Verification**

- The project team along with architect/designer may walk through the design to find the completeness and give comments, if any.
- Traceability of design with requirement must be established in requirement traceability matrix.

☞ **Design Validation**

- Validation of design occurs at two or more stages during the software development lifecycle.

The first stage happens when data flow diagrams can be created by referring to the design document.

If the flow of data is complete, the design is considered to be complete. Any interruption in the flow of data indicates lacunae in design.

This is also termed 'flow anomaly'. The second stage happens at integration testing and interface testing.

4.7.1 Coding

Coding is done by developers where the low-level designs are implemented.

Code Verification

Peer / Code review helps in identification of errors with respect to coding standards, indenting standards, commenting standards, and variable declaration issues.

Checklist approach is used in code review.

Code Validation

Validation of coding happens through unit testing where individual units are tested separately. The developer may have to write special programs (such as stubs and drivers) so that individual units can be tested.

The executable is created by combining stubs, drivers and the units under testing. This executable is tested with the test cases mainly derived from low-level design.

Test results and defects are then logged.

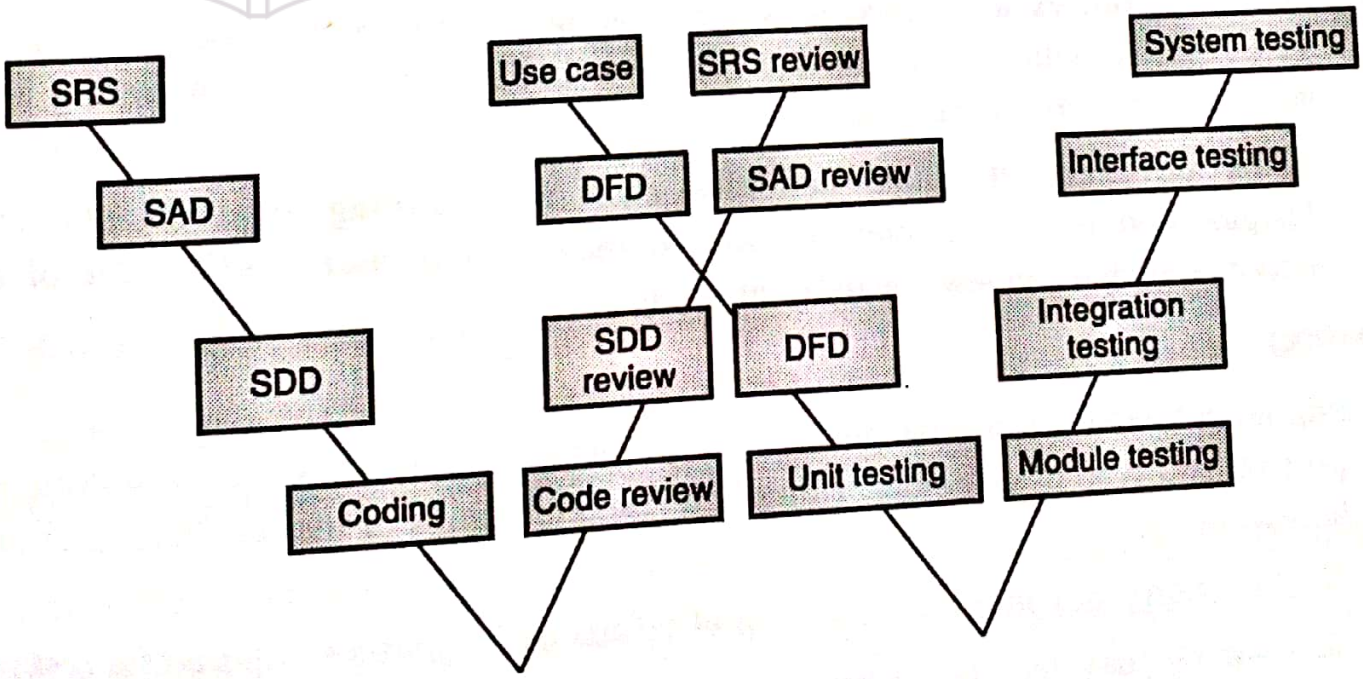

Fig.4.7.1

Syllabus Topic : Critical Roles and Responsibilities

4.8 Critical Roles and Responsibilities

Q. 4.8.1 What are the various critical roles and responsibilities in verification and validation? (Ref. Sec. 4.8) **(5 Marks)**

In software verification and validation, three critical roles can be identified. Let us try to define these roles and their responsibilities.

Development

- Development team may be comprised of various roles under them. They may be performing various activities as per their roles and responsibilities at different stages.
- These sub roles may include development manager, project leads, module leads, team leads, developers and unit testers as per organization structure and project requirements.
- Some of the activities related to the development group may be as follows :
  - Project planning activities include requirement elicitation, estimation, project planning, scheduling, and definition of quality attributes required by the customer.
  - Resourcing may include identification and organization of an adequate number of people, machines, hardware, software, and tools as required by the project.
  - Interacting with customer and other stakeholders as per project requirements.
  - Defining policies and procedures for creating development work, verification and validation activities to ensure that quality is built properly, and delivering it to test team/customer as the case may be.
  - Doing development-related activities such as capturing requirements, creating designs, coding, and implementation. It may also include quality control related activities such as reviews, walkthrough. etc.

Testing

- Testing team includes test manager, test leads, and testers as per scope of testing, size of the project, and type of customer. Roles and responsibilities of the test team may include the following :
  - Test planning may include estimation of efforts and resources required for testing.
  - Resourcing may include identification and organization of an adequate number of people, machines, hardware, software, and tools as required by the project.
  - Interacting with customer and other stakeholders as per project requirements. It may include asking queries, responding to customer/development team requests and so on.

- Defining policies and procedures for creating and executing tests as per test strategy and test plan. Testers may have to take part in verification and validation activities related to test artifacts.
- Supporting development team by providing adequate information about the defects. If required, testers may have to reproduce defects in front of customer/development team.

Customer

- Customer may be the final user group, or people who are actually sponsoring the project.
- Roles and responsibilities of a customer may include the following :
  - Specifying requirements and signing off requirement statement and designs as per contract. It may also include solving any queries or issues raised by the development/test team.
  - Participating in acceptance testing as per roles and responsibilities defined in acceptance test plan. A customer may be responsible for alpha, beta and gamma testing, as the case may be.

4.9 Exam Pack (Review Questions)

Syllabus Topic : Introduction

Q. 1 Explain software verification and its features. *(Ans. : Refer sections 4.1, 4.1.1 and 4.1.2)* **(5 Marks)**

Syllabus Topic : Verification

Q. 2 Explain software verification and its features. *(Ans. : Refer sections 4.1, 4.1.1 and 4.1.2)* **(5 Marks)**

Syllabus Topic : Verification Workbench

Q. 3 Explain verification workbench. *(Ans. : Refer section 4.1.3)* **(5 Marks)**

Syllabus Topic : Methods of Verification

Q. 4 What are the different methods of verification ? *(Ans. : Refer section 4.1.4)* **(5 Marks)**

Syllabus Topic : Types of Reviews on the basis of Stage Phase

Q. 5 Explain the different types of reviews on the basis of Stage phase. *(Ans. : Refer section 4.2)* **(5 Marks)**

Syllabus Topic : Entities Involved in Verification

Q. 6 Which are the various entities involved in verification? *(Ans. : Refer section 4.2.1)* **(5 Marks)**

☞ **Syllabus Topic : Reviews in Testing Lifecycle**

Q. 7 Which are the different reviews in software testing life cycle (STLC) ? *(Ans. : Refer section 4.2.2)* **(5 Marks)**

☞ **Syllabus Topic : Coverage in Verification**

Q. 8 Explain coverage in verification. *(Ans. : Refer section 4.2.3)* **(5 Marks)**

☞ **Syllabus Topic : Concerns of Verification**

Q. 9 List various concerns of verification. *(Ans. : Refer section 4.2.4)* **(5 Marks)**

☞ **Syllabus Topic : Validation**

Q. 10 Explain validation. *(Ans. : Refer section 4.3)* **(5 Marks)**

Q. 11 Differentiate between verification and validation. *(Ans. : Refer section 4.3)* **(5 Marks)**

☞ **Syllabus Topic : Validation Workbench**

Q. 12 Explain validation workbench. *(Ans. : Refer section 4.3.2)* **(5 Marks)**

☞ **Syllabus Topic : Levels of Validation**

Q. 13 What are the different levels of validation? *(Ans. : Refer section 4.3.3)* **(5 Marks)**

☞ **Syllabus Topic : Coverage in Validation**

Q. 14 Explain the coverage offered during validation. *(Ans. : Refer section 4.3.4)* **(5 Marks)**

☞ **Syllabus Topic : Acceptance Testing**

Q. 15 Explain acceptance testing in detail. *(Ans. : Refer section 4.4)* **(5 Marks)**

☞ **Syllabus Topic : Management of Verification and Validation**

Q. 16 Explain management of verification and validation. *(Ans. : Refer section 4.4.2)* **(5 Marks)**

☞ **Syllabus Topic : Software Development Verification and Validation (V and V) Activities**

Q. 17 Explain software development verification and validation activities. *(Ans. : Refer section 4.4.3)* **(5 Marks)**

☞ **Syllabus Topic : V Test Model-Introduction**

Q. 18 Explain the V Test model in detail. *(Ans. : Refer section 4.5)* **(5 Marks)**

☞ **Syllabus Topic : V-Model for Software**

Q. 19 Explain V-model for Software. *(Ans. : Refer section 4.5.2)* **(5 Marks)**

Syllabus Topic : Testing During Proposal Stage

Q. 20 Why is testing required during proposal stage? *(Ans. : Refer section 4.5.3)* **(5 Marks)**

Syllabus Topic : Testing During Requirement Stage

Q. 21 Explain testing during requirement stage. *(Ans. : Refer section 4.5.4)* **(5 Marks)**

Syllabus Topic : Testing During Test-Planning Phase

Q. 22 Explain testing during test-planning phase. *(Ans. : Refer section 4.5.5)* **(5 Marks)**

Syllabus Topic : Testing During Design Phase

Q. 23 Explain testing during design phase. *(Ans. : Refer section 4.5.6)* **(5 Marks)**

Syllabus Topic : Testing During Coding

Q. 24 Explain testing during coding phase. *(Ans. : Refer section 4.5.7)* **(5 Marks)**

Syllabus Topic : VV Model

Q. 25 Explain the VV model in detail. *(Ans. : Refer section 4.6)* **(5 Marks)**

Syllabus Topic : Critical Roles and Responsibilities

Q. 26 What are the various critical roles and responsibilities in verification and validation? *(Ans. : Refer section 4.8)* **(5 Marks)**

❑❑❑