



SNS COLLEGE OF TECHNOLOGY
(An Autonomous Institution)
COIMBATORE-35
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



19CST201 - AGILE SOFTWARE ENGINEERING

TWO MARKS QUESTIONS AND ANSWERS

UNIT - I

INTRODUCTION TO SOFTWARE ENGINEERING

1. What is software engineering?

Software engineering is a discipline in which theories, methods and tools are applied to develop professional software.

2. What is Software?

Software is nothing but a collection of computer programs that are related documents that are indented to provide desired features, functionalities and better performance.

3. What are the characteristics of the software?

Ø Software is engineered, not manufactured.

Ø Software does not wear out.

Ø Most software is custom built rather than being assembled from components.

4. What are the various categories of software?

Ø System software

Ø Application software

Ø Engineering/Scientific software

Ø Embedded software

Ø Web Applications

Ø Artificial Intelligence software

5. What are the challenges in software?

Ø Copying with legacy systems.

Ø Heterogeneity challenge

Ø Delivery times challenge

6. Define software process.

Software process is defined as the structured set of activities that are required to develop the software system.

7. What are the fundamental activities of a software process?

Ø Specification

Ø Design and implementation

Ø Validation

Ø Evolution

8. What are the umbrella activities of a software process?

Ø Software project tracking and control.

Ø Risk management.

Ø Software Quality Assurance.

Ø Formal Technical Reviews.

Ø Software Configuration Management.

Ø Work product preparation and production.

Ø Reusability management.

Ø Measurement.

9. What are the merits of incremental model?

i. The incremental model can be adopted when there are less number of people involved in the project.

ii. Technical risks can be managed with each increment.

iii. For a very small time span, at least core product can be delivered to the customer.

10. List the task regions in the Spiral model.

Ø **Customer communication** – In this region it is suggested to establish customer communication.

Ø **Planning** – All planning activities are carried out in order to define resources timeline and other project related activities.

Ø **Risk analysis** – The tasks required to calculate technical and management risks.

Ø **Engineering** – In this the task region, tasks required to build one or more representations of applications are carried out.

Ø **Construct and release** – All the necessary tasks required to construct, test, install the applications are conducted. $\frac{3}{4}$ **Customer evaluation** – Customer's feedback is

obtained and based on the customer evaluation required tasks are performed and implemented at installation stage.

11. What are the drawbacks of spiral model?

- i. It is based on customer communication. If the communication is not proper then the software product that gets developed will not be the up to the mark.
- ii. It demands considerable risk assessment. If the risk assessment is done properly then only the successful product can be obtained.

12. What is System Engineering?

System Engineering means designing, implementing, deploying and operating systems which include hardware, software and people

13. List the process maturity levels in SEIs CMM.

Level 1:Initial – Few processes are defined and individual efforts are taken.

Level 2:Repeatable – To track cost schedule and functionality basic project management processes are established.

Level 3:Defined – The process is standardized, documented and followed.

Level 4:Managed – Both the software process and product are quantitatively understood and controlled using detailed measures.

Level 5:Optimizing – Establish mechanisms to plan and implement change.

14. What is an effector process?

The effector process is a process that verifies itself. The effector process exists in certain criteria.

15. Define the computer based system.

The computer based system can be defined as “a set or an arrangement of elements that are organized to accomplish some predefined goal by processing information”.

16. What does Verification represent?

Verification represents the set of activities that are carried out to confirm that the software correctly implements the specific functionality.

17. What does Validation represent?

Validation represents the set of activities that ensure that the software that has been built is satisfying the customer requirements.

18. What are the steps followed in testing?

i. **Unit testing** – The individual components are tested in this type of testing.

ii. **Module testing** – Related collection of independent components are tested.

iii. **Sub-system testing** – This is a kind of integration testing. Various modules are integrated into a subsystem and the whole subsystem is tested.

iv. **System testing** – The whole system is tested in this system.

v. **Acceptance testing** – This type of testing involves testing of the system with customer data. If the system behaves as per customer need then it is accepted.

19. What is the use of CMM?

Capability Maturity Model is used in assessing how well an organisation's processes allow to complete and manage new software projects.

20. Name the Evolutionary process Models.

i. Incremental model ii. Spiral model

iii. WIN-WIN spiral model iv. Concurrent Development

21. What is requirement engineering?

Requirement engineering is the process of establishing the services that the customer requires from the system and the constraints under which it operates and is developed.

22. What are the various types of traceability in software engineering?

i. **Source traceability** – These are basically the links from requirement to stakeholders who propose these requirements.

ii. **Requirements traceability** – These are links between dependant requirements.

iii. **Design traceability** – These are links from requirements to design.

UNIT – II AGILE DEVELOPMENT

Q1. Name the most important components of agile?

Ans.

- Test driven development, pair programming, continuous deployment, etc.
- Class responsibilities and collaborators cards
- Daily stand-up meetings

Q2. What is the difference between agile and traditional project development?

Ans. Agile enables iterative approaches to project implementation, whereas traditional project development involves distinct project life cycle phases.

Q3. What are the advantages of an agile model?

Ans. The advantages are as follows:

- Working software to be delivered in weeks rather than months
- Fixation of the bugs in the middle of the project
- An iterative approach to software design
- Daily cooperation between individuals and developers
- A continuous delivery of useful software

[Get Agile & Scrum Certified Now>>](#)

Q4. Do you know some agile frameworks?

Ans. Some of the most popular agile frameworks are Scrum, Test driven development, feature driven development and Kanban.

Q5. What qualities should an agile tester have?

Ans.

- Must know the requirements quickly
- Must possess good communication skills to have in-depth discussions with business associates
- Must have the right set of tools to be used to help the testing
- Must know to execute test on a constant basis

Q6. Explain agile manifesto?

Ans. It defines four main points:

- People and interaction over tools
- Customer collaboration over contract negotiation

- Responding to change over a plan
- Working software over comprehensive documentation

Q7. What are the main agile quality strategies?

- Small feedback cycles
- Re-factoring
- Dynamic code analysis
- Iteration

Q8. State the difference between Agile and Scrum.

Agile is a broader term and a practice that enables continuous iteration of development and testing in the Software Development Life Cycle (SDLC). Scrum falls under the Agile umbrella and is a specific set of rules to be followed during Agile software development.

Q9. How long the Scrum cycle last?

The life of Scrum cycle varies as per project and team size. It takes usually 3 – 4 weeks to complete a Scrum sprint.

Q10. What is Scrumban?

A Scrumban is a management framework and Kanban-based model for software development. It is used in software projects that require continuous maintenance, problem-solving, error fixation, etc. This model is chosen to complete a project within minimum time.

Q11. Explain a test stub.

A test stub is a code that simulates behavior of software components or modules. It also mimics a component in the system and replaces it. Its output is same as the component replaced by it.

Q12. What is Test Driven Development (TDD)?

TDD is a programming practice that instructs developers to write error-free and simpler codes in case an automated test has failed.

Q13. How will you define storyboard in Agile?

Storyboard is a graphic organizer or visual representation of the progress of a software project. This has four columns –

- To do
- In Progress
- Test
- Done

This project insight helps the developer understand the work flow and complete the work or rework in case of any misses or issues.

Q14. What are Epic, User Stories, and Tasks?

Epic – A story that is so big that it is broken down into a group of related user stories.

User Stories – Actual business requirements created by the business owner.

Task – Created by the development team. Here, a story is broken down into tasks to ensure efficient estimation, development and testing.

Q15. What is “Build Breaker”?

A Build Breaker is usually a bug in the software created because of any execution fail. It may stop the build process, cause unacceptable warnings, and even lead to failures in the automated test environment.

Q16. What are Prototypes and Wireframes used for?

Prototypes and Wireframes are essential components of website or mobile app design process. They are used to present the design concept and the basic functions of the website for the clients.

Q17. What is API?

API or Application Programming Interface is a communication protocol between clients and servers. This contract is created to ensure smooth functioning and building of client-side software. The client describes its requirements and the server is responsible to deliver the product as agreed.

Q18. Are you aware of any drawbacks of the Agile model?

Yes, some of the major disadvantages of the Agile system are –

- Pretty unpredictable at many instances
- Improper focus on design and documentation
- Project may fall apart if consumer guidelines aren't clearly understood
- Decision making limited only for top management, leaving less scope for new developers to learn

Q19. What is a product roadmap?

It is a product vision that offers a holistic view of product features. A product roadmap helps the product teams to communicate the strategy with cross-functional teams as well as key stakeholders.

Q20. Name some Agile project management tools.

Some popular Agile project management tools are –

- Active Collab
- Agilo for Scrum

- Atlassian Jira
- Gravity
- Icescrum
- Kanbanize
- com
- Nifty
- Pivotal Tracker
- SpiraTeam
- SpiraTeam
- SprintGround
- VSTS
- Wrike

Q21. What are the popular Agile Scrum certifications one can take?

Some of the popular certifications to strengthen the candidate of any professional working in the field of Agile Scrum are –

- Agile Certified Practitioner (ACP)
- Agile Scrum Master (ASM)
- Professional Scrum Master (PSM)
- Certified Scrum Master (CSM)
- Safe Agilist
- PMI Agile Certified Practitioner
- SAFe 4.0 Scrum Master
- Advanced Certified Scrum Master
- Certified Scrum Product Owner/ Certified Scrum Developer

Q22. Name different types of impediments?

Ans. Impediments can be of the following types:

- Technical issues
- Natural calamities
- Organizational issues
- Infrastructure issues
- Stakeholder issues
- Unskilled team members

Q23. Name the components in the user story?

Ans. User story has three components (3Cs)-

- **Card** – It describes the whole user story in the basic form and written in a physical form on a card. The format used in the user story: As a [type of user], I want [goal] so that [some reason].
- **Conversation** – It depicts the communication between product owner, consumer, other members to discuss details in the card.
- **Confirmation** – It helps in finding acceptance criteria in the user story so that the story can be used.

Q24. Name the various estimation methods in agile?

Ans. Following are the various estimation methods used in Agile:

- Dot Voting
- T-Shirt Sizing
- Affinity Mapping
- Planning Poker
- Ordering method
- The Bucket System

Q25. Who can re-prioritize the timeboxed plan if it is required?

Ans. A whole team with the product owner, developer are involved to re-prioritize the timeboxed plan.

Q26. What is a tracer bullet in Agile?

Ans: Tracer bullet is implemented to explore the working of codes and check the feasibility in the end goal.

Q27. Explain Kanban in Agile?

Ans. It is a tool used for estimating the work under the process. Here, the kanban board examines the status of a developing story and also traces it.

Q28. What is the difference between Incremental and iterative development?

Ans. In incremental development, the software is created and delivered in increments where every increment consists of completed features of sub functionality of the system whereas in iterative development the software is created and delivered to the customers. After the delivery, they got feedback from customers which is received in the software. Again they create software in a new sprint and then deliver it to customers.

Q29. Explain what is Zero sprint in Agile?

Ans. A zero sprint is described as one previous step to the first sprint. It includes activities like setting up the environment to develop, prepare backlog, and more. Zero sprint needs to start before the start of the initial sprint.

Q30. Mention the most critical agile Matrices?

Ans. Below are the most critical agile Matrices.

- **Defect removal awareness** – It works for the maintaining of quality of products by the active members.
- **The cumulative flow diagram** –In this uniform workflow the x-axis represents time and the y-axis defines the number of efforts.

- **Velocity** – It includes a clear understanding of capacity, progress and much more. Keeping track of the velocity is required and it can be measured by combining all the approved estimates of the story.
- **Work category allocation** – It depicts a clear picture of the priority of work and where you are investing your time.
- **The sprint burndown matric** – It works for keeping the track of the finishing of work in the sprint.
- **Time coverage** – Here, the time period is calculated with the ratio of the number of lines of code that is considered by the number of similar lines of code.
- **Defect resolution time** – In this alignment, the team members find the errors and fix them. The fixing of bugs involves several processes.

Q31. What are the conditions where we don't have to use Agile?

Ans: Below are the conditions we have to follow:

- Time constraint
- Customer availability
- Team skill
- Functionality division
- Flexible requirement

Q32. How do QA provide benefits to an agile team?

Ans. QA adds value to the Agile team by working in a different scenario of story thinking and provides their valuable feedback to the developers whether the new functionality is worthful or not.

Q33. What is the main difference between Agile and Waterfall?

Ans. Agile is a continuous process. In comparison, the Waterfall methodology is a linear process. Both have their advantages and disadvantages. Waterfall is best when the requirements are already determined. In case of agile, there is room for change in requirements as the project progresses.

Q34. Name the 12 Agile principles.

Ans. The 12 principles are based on the Agile manifesto.

1. Satisfy the customer
2. Welcome change in requirements through the last stages
3. Deliver work on shorter time scales
4. Work together with business people and developer
5. Provide an environment for motivation
6. Face-to-face conversation
7. Deliver working software as a sign of progress
8. Promote sustainable development
9. Continuous importance to design
10. Simplicity in work
11. Self organising teams

UNIT - III

AGILE SCRUM FRAMEWORK

Q #1) What is Agile Testing?

Answer: Agile Testing is a practice that a QA follows in a dynamic environment where testing requirements keep changing according to customer needs. It is done parallel to the development activity where the testing team receives frequent small codes from the development team for testing.

Q #2) What is the difference between burn-up and burn-down charts?

Answer: Burn-up and burn-down charts are used to keep track of the progress of the project. Burn-up charts represent how much work has been completed in any project whereas Burn-down chart represents the remaining work in a project.

Q #3) Define the roles in Scrum?

Answer:

There are mainly three roles that a Scrum team have:

1. **Project Owner** has the responsibility of managing the product backlog. Works with end-users and customers and provides proper requirements to the team to build the proper product.
2. **Scrum Master** works with the scrum team to make sure each sprint gets completed on time. Scrum master ensures proper workflow for the team.
3. **Scrum Team:** Each member of the team should be self-organized, dedicated and responsible for the high quality of the work.

Q #4) What is Product Backlog & Sprint Backlog?

Answer: The **Product backlog** is maintained by the project owner which contains every feature and requirement of the product.

Sprint backlog can be treated as the subset of product backlog which contains features and requirements related to that particular sprint only.

Q #5) Explain Velocity in Agile.

Answer: Velocity is a metric that is calculated by the addition of all efforts estimates associated with user stories completed in an iteration. It predicts how much work Agile can complete in a sprint and how much time will it require to complete a project.

Q #6) Explain the difference between a traditional Waterfall model and Agile testing?

Answer: Agile testing is done parallel to the development activity whereas a traditional waterfall model testing is done at the end of the development.

As done in parallel, agile testing is done on small features whereas, in a waterfall model, testing is performed on the whole application.

Q #7) Explain Pair Programming and its benefits?

Answer: Pair programming is a technique in which two programmer works as a team in which one programmer writes code and other one reviews that code. They both can switch their roles.

Benefits:

- **Improved code quality:** As the second partner reviews the code simultaneously, it reduces the chances of mistake.

- **Knowledge transfer is easy:** One experienced partner can teach another partner about the techniques and codes.

Q #8) What is Re-factoring?

Answer: Modification of the code without changing its functionality to improve the performance is called Re-factoring.

Q #9) Explain the Iterative and Incremental Development in Agile?

Answer:

Iterative Development: Software is developed and delivered to the customer and based on the feedback again developed in cycles or releases and sprints. **Example:** Release 1 software is developed in 5 sprints and delivered to the customer. Now, the customer wants some changes, then the development team plan for 2nd release which can be completed in some sprints and so on.

Incremental Development: Software is developed in parts or increments. In each increment, a portion of the complete requirement is delivered.

Q #10) How do you deal when requirements change frequently?

Answer: This question is to test the analytical capability of the candidate.

The answer can be: Work with PO to understand the exact requirement to update test cases. Also, understand the risk of changing the requirement. Apart from this, one should be able to write a generic test plan and test cases. Don't go for the automation until requirements are finalized.

Q #11) What is a test stub?

Answer: Test stub is a small code that mimics a specific component in the system and can replace it. Its output is the same as the component it replaces.

Q #12) What qualities should a good Agile tester have?

Answer:

- He should be able to understand the requirements quickly.
- He should know Agile concepts and principals.
- As requirements keep changing, he should understand the risk involved in it.
- The agile tester should be able to prioritize the work based on the requirements.
- Communication is a must for an Agile tester as it requires a lot of communication with developers and business associates.

Q #13) What is the difference between Epic, User stories & Tasks?

Answer:

User Stories: It defines the actual business requirement. Generally created by the business owner.

Task: To accomplish the business requirements development team create tasks.

Epic: A group of related user stories is called an Epic.

Q #14) What is a Taskboard in Agile?

Answer: Taskboard is a dashboard that shows the progress of the project.

It contains:

- **User Story:** It has the actual business requirement.
- **To Do:** Tasks that can be worked on.
- **In Progress:** Tasks in progress.

- **To Verify:** Tasks pending for verification or testing
- **Done:** Completed tasks.

Q #15) What is Test Driven Development (TDD)?

Answer: It is a Test-first development technique in which we add a test first before we write the complete production code. Next, we run the test and based on the result refactor the code to fulfill the test requirement.

Q #16) How QA can add value to an agile team?

Answer: QA can provide value addition by think outside the box about the various scenarios to test a story. They can provide quick feedback to the developers about whether new functionality is working fine or not.

Q #17) What is Scrum ban?

Answer: It is a software development model that is a combination of Scrum and Kanban. Scrumban is considered for maintaining projects in which there are frequent changes or unexpected user stories. It can reduce the minimum completion time for user stories.

Q #18) What is the Application Binary Interface?

Answer: Application Binary Interface or ABI is defined as an interface for compiled application programs or we can say it describes the low-level interface between an application and the operating system.

Q #19) What is the Zero sprint in Agile?

Answer: It can be defined as a pre-preparation step to the first sprint. Activities like setting development environment, preparing backlog, etc need to be done before starting the first sprint and can be treated as Sprint zero.

Q #20) What is Spike?

Answer: There may be some technical issues or design problem in the project which needs to be resolved first. To provide the solution to this problem “Spikes” are created.

Spikes are of two types- Functional and Technical.

Q #21) Name some Agile quality strategies.

Answer: Some Agile quality strategies are-

1. Re-factoring
2. Small feedback cycles
3. Dynamic code analysis
4. Iteration

Q #22) What is the importance of daily stand up meetings?

Answer: Daily stand up meeting is essential for any team in which team discuss,

1. How much work has been completed?
2. What are the plans to resolve technical issues?
3. What steps need to done to complete the projects etc?

Q #23) What is a tracer bullet?

Answer: It can be defined as a spike with the current architecture or the current set of best practices. The purpose of a tracer bullet is to examine how an end-to-end process will work and examine feasibility.

Q #24) How the velocity of the sprint is measured?

Answer: If capacity is measured as a percentage of a 40 hours weeks then, completed story points * team capacity

If capacity is measured in man-hours then Completed story points/team capacity

Q #25) What is Agile manifesto?

Answer: Agile manifesto defines an iterative and people-centric approach to software development. It has 4 key values and 12 principals.

UNIT IV

USER INTERFACE DESIGN

01. What are the elements of design model?

- i. Data design
- ii. Architectural design
- iii. Interface design
- iv. Component-level design

02. Define design process.

Design process is a sequence of steps carried through which the requirements are translated into a system or software model.

03. List the principles of a software design.

- i. The design process should not suffer from “tunnel vision”.
- ii. The design should be traceable to the analysis model.
- iii. The design should exhibit uniformity and integration.
- iv. Design is not coding.
- v. The design should not reinvent the wheel.

04. What is the benefit of modular design?

Changes made during testing and maintenance becomes manageable and they do not affect other modules.

05. What is a cohesive module?

A cohesive module performs only “one task” in software procedure with little interaction with other modules. In other words cohesive module performs only one thing.

06. What are the different types of Cohesion?

i. Coincidentally cohesive –The modules in which the set of tasks are related with each other loosely then such modules are called coincidentally cohesive.

ii. Logically cohesive – A module that performs the tasks that are logically related with each other is called logically cohesive.

iii. Temporal cohesion – The module in which the tasks need to be executed in some specific time span is called temporal cohesive.

iv. Procedural cohesion – When processing elements of a module are related with one another and must be executed in some specific order then such module is called procedural cohesive.

v. Communicational cohesion – When the processing elements of a module share the data then such module is called communicational cohesive.

07. What is Coupling?

Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules.

08. What are the various types of coupling?

i. Data coupling – The data coupling is possible by parameter passing or data interaction.

ii. Control coupling – The modules share related control data in control coupling.

iii. Common coupling – The common data or a global data is shared among modules. **iv.**

Content coupling – Content coupling occurs when one module makes use of data or control information maintained in another module.

09. What are the common activities in design process?

i. System structuring – The system is subdivided into principle subsystems components and communications between these subsystems are identified.

ii. Control modeling – A model of control relationships between different parts of the system is established.

iii. Modular decomposition – The identified subsystems are decomposed into modules.

10. What are the benefits of horizontal partitioning?

i. Software that is easy to test.

ii. Software that is easier to maintain.

iii. Propagation of fewer sideeffects. **iv.** Software that is easier to extend.

11. What is vertical partitioning?

Vertical partitioning often called factoring suggests that the control and work should be distributed top-down in program structure.

12. What are the advantages of vertical partitioning?

- i. These are easy to maintain changes.
- ii. They reduce the change impact and error propagation.

13. What are the various elements of data design?

i. Data object – The data objects are identified and relationship among various data objects can be represented using ERD or data dictionaries.

ii. Databases – Using software design model, the data models are translated into data structures and data bases at the application level.

iii. Data warehouses – At the business level useful information is identified from various databases and the data warehouses are created.

14. List the guidelines for data design.

- i. Apply systematic analysis on data.
- ii. Identify data structures and related operations.
- iii. Establish data dictionary.
- iv. Use information hiding in the design of data structure.
- v. Apply a library of useful data structures and operations.

15. Name the commonly used architectural styles.

- i. Data centered architecture.
- ii. Data flow architecture.
- iii. Call and return architecture.
- iv. Object-oriented architecture.
- v. Layered architecture.

16. What is Transform mapping?

The transform mapping is a set of design steps applied on the DFD in order to map the transformed flow characteristics into specific architectural style.

17. What is a Real time system?

Real time system is a software system in which the correct functionalities of the system are dependent upon results produced by the system and the time at which these results are produced.

18. What is SCM?

Software Configuration Management is a set of activities carried out for identifying, organizing and controlling changes throughout the lifecycle of computer software.

19. What is SCI?

Software Configuration Item is information that is carried as part of the software engineering process.

UNIT V

SOFTWARE TESTING & SCM

1. Define software testing?

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding.

2. What are the objectives of testing?

- i. Testing is a process of executing a program with the intend of finding an error.
- ii. A good test case is one that has high probability of finding an undiscovered error.
- iii. A successful test is one that uncovers as an-yet undiscovered error.

3. What are the testing principles the software engineer must apply while performing the software testing?

- i. All tests should be traceable to customer requirements.
- ii. Tests should be planned long before testing begins.
- iii. The pareto principle can be applied to software testing-80% of all errors uncovered during testing will likely be traceable to 20% of all program modules.
- iv. Testing should begin “in the small” and progress toward testing “in the large”.
- v. Exhaustive testing is not possible.
- vi. To be most effective, an independent third party should conduct testing.

4. What are the two levels of testing?

- i. Component testing** Individual components are tested. Tests are derived from developer’s experience.
- ii. System Testing** The group of components are integrated to create a system or sub- system is done. These tests are based on the system specification.

5. What are the various testing activities?

- i. Test planning
- ii. Test case design
- iii. Test execution
- iv. Data collection
- v. Effective evaluation

6. Write short note on black box testing.

The black box testing is also called as behavioral testing. This method fully focuses on the functional requirements of the software. Tests are derived that fully exercise all functional requirements.

7. What is equivalence partitioning?

Equivalence partitioning is a black box technique that divides the input domain into classes of data. From this data test cases can be derived. Equivalence class represents a set of valid or invalid states for input conditions.

8. What is a boundary value analysis?

A boundary value analysis is a testing technique in which the elements at the edge of the domain are selected and tested. It is a test case design technique that complements equivalence partitioning technique. Here instead of focusing on input conditions only, the test cases are derived from the output domain.

9. What are the reasons behind to perform white box testing?

There are three main reasons behind performing the white box testing.

1. Programmers may have some incorrect assumptions while designing or implementing some functions. Due to this there are chances of having logical errors in the program. To detect and

correct such logical errors procedural details need to be examined.

2. Certain assumptions on flow of control and data may lead programmer to make design errors. To uncover the errors on logical path, white box testing is must.

3. There may be certain typographical errors that remain undetected even after syntax and type checking mechanisms. Such errors can be uncovered during white box testing.

10. What is cyclomatic complexity?

Cyclomatic complexity is a software metric that gives the quantitative measure of logical complexity of the program. The Cyclomatic complexity defines the number of independent paths in the basis set of the program that provides the upper bound for the number of tests

that must be conducted to ensure that all the statements have been executed at least once.

11. How to compute the cyclomatic complexity?

The cyclomatic complexity can be computed by any one of the following ways.

1. The numbers of regions of the flow graph correspond to the cyclomatic complexity.

2. Cyclomatic complexity, $V(G)$, for the flow graph, G , is defined as: $V(G) = E - N + 2$,

E -- number of flow graph edges, N -- number of flow graph nodes

3. $V(G) = P + 1$ Where P is the number of predicate nodes contained in the flow graph.

12. Distinguish between verification and validation. $\frac{3}{4}$ Verification refers to the set of activities that ensure that software correctly implements a specific function.

$\frac{3}{4}$ Validation refers to a different set of activities that ensure that the software that has been built is traceable to the customer requirements.

According to Boehm,

Ø Verification: "Are we building the product right?"

Ø Validation: "Are we building the right product?"

13. What are the various testing strategies for conventional software?

i. Unit testing ii. Integration testing. iii. Validation testing. iv. System testing.

14. Write about drivers and stubs.

Drivers and stub software need to be developed to test incompatible software.

Ø The "driver" is a program that accepts the test data and prints the relevant results.

Ø The "stub" is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.

15. What are the approaches of integration testing?

The integration testing can be carried out using two approaches.

1. The non-incremental testing.

2. Incremental testing.

16. What are the advantages and disadvantages of big-bang? Advantages:

Ø This approach is simple.

Disadvantages:

Ø It is hard to debug.

Ø It is not easy to isolate errors while testing.

Ø In this approach it is not easy to validate test results.

Ø After performing testing, it is impossible to form an integrated system.

17. What are the benefits of smoke testing?

Ø Integration risk is minimized.

Ø The quality of the end-product is improved.

Ø Error diagnosis and correction are simplified.

Ø Assessment of program is easy.

18. What are the conditions exists after performing validation testing?

After performing the validation testing there exists two conditions.

Ø The function or performance characteristics are according to the specifications and are accepted.

Ø The requirement specifications are derived and the deficiency list is created. The deficiencies then can be resolved by establishing the proper communication with the customer.\

19. Distinguish between alpha and beta testing.

Ø Alpha and beta testing are the types of acceptance testing.

Ø **Alpha test:** The alpha testing is attesting in which the version of complete software is tested by the customer under the supervision of developer. This testing is performed at developer's site.

Ø **Beta test:** The beta testing is a testing in which the version of the software is tested by the customer without the developer being present. This testing is performed at customer's site.

20. What are the various types of system testing?

1. **Recovery testing** – is intended to check the system's ability to recover from failures.

2. **Security testing** – verifies that system protection mechanism prevent improper penetration or data alteration.

3. **Stress testing** – Determines breakpoint of a system to establish maximum service level.

4. **Performance testing** – evaluates the run time performance of the software, especially real-time software.

21. Define debugging.

Debugging is defined as the process of removal of defect. It occurs as a consequence of successful testing.

22. What are the common approaches in debugging?

Ø **Brute force method:** The memory dumps and run-time tracks are examined and program with write statements is loaded to obtain clues to error causes.

Ø **Back tracking method:** The source code is examined by looking backwards from symptom to potential causes of errors.

Ø **Cause elimination method:** This method uses binary partitioning to reduce the number of locations where errors can exist.

23. Write about the types of project plan.

Ø **Quality plan** – This plan describes the quality procedures and standards that will be used in a project.

Ø **Validation plan** – This plan describes the approach, resources and schedule required for system validation.

Ø **Configuration management plan** – This plan focuses on the configuration management procedures and structures to be used.

Ø **Maintenance plan** – The purpose of maintenance plan is to predict the maintenance requirements of the system, maintenance cost and efforts required.

Ø **Staff development plan** – This plan describes how to develop the skills and experience of the project team members.