K.S.R. COLLEGE OF ENGINEERING (Autonomous)

Vision of the Institution

- We envision to achieve status as an excellent educational institution in the global knowledge hub, making self-learners, experts, ethical and responsible engineers, technologists, scientists, managers, administrators and entrepreneurs who will significantly contribute to research and environment friendly sustainable growth of the nation and the world.

Mission of the Institution

- To inculcate in the students self-learning abilities that enable them to become competitive and considerate engineers, technologists, scientists, managers, administrators and entrepreneurs by diligently imparting the best of education, nurturing environmental and social needs.
- To foster and maintain a mutually beneficial partnership with global industries and Institutions through knowledge sharing, collaborative research and innovation.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision of the Department

- To create ever green professionals for software industry, academicians for knowledge cultivation and researchers for contemporary society modernization.

Mission of the Department

- To produce proficient design, code and system engineers for software development.
- To keep updated contemporary technology and fore coming challenges for welfare of the society.

Programme Educational Objectives (PEOs)

PEO1 : Figure out, formulate, analyze typical problems and develop effective solutions by imparting the idea and principles of science, mathematics, engineering fundamentals and computing.

PEO2 : Competent professionally and successful in their chosen career through life-long learning.

PEO3 : Excel individually or as member of a team in carrying out projects and exhibit social needs and follow professional ethics.

K.S.R. COLLEGE OF ENGINEERING (Autonomous)

Department of Computer Science and Engineering

Subject Name: SOFTWARE TESTING
Subject Code: 16CS713 **Year/Semester:IV / VII**

Course Outcomes: On completion of this course, the student will be able to

CO1	Know the complete software testing life cycle.
CO2	Determine the various terms and technologies used in testing domain.
CO3	Understand the usage of testing framework, process and test management.
CO4	Generate test plan and designing test cases.
CO5	Understand the concepts of test management process.

Program Outcomes (POs) and Program Specific Outcomes (PSOs)

A. Program Outcomes (POs)

Engineering Graduates will be able to :

PO1 **Engineering knowledge:** Ability to exhibit the knowledge of mathematics, science, engineering fundamentals and programming skills to solve problems in computer science.

PO2 **Problem analysis:** Talent to identify, formulate, analyze and solve complex engineering problems with the knowledge of computer science. .

PO3 **Design/development of solutions:** Capability to design, implement, and evaluate a computer based system, process, component or program to meet desired needs.

PO4 **Conduct investigations of complex problems:** Potential to conduct investigation of complex problems by methods that include appropriate experiments, analysis and synthesis of information in order to reach valid conclusions.

PO5 **Modern tool Usage:** Ability to create, select, and apply appropriate techniques, resources and modern engineering tools to solve complex engineering problems.

PO6 **The engineer and society:** Skill to acquire the broad education necessary to understand the impact of engineering solutions on a global economic, environmental, social, political, ethical, health and safety.

PO7 **Environmental and sustainability:** Ability to understand the impact of the professional engineering solutions in societal and Environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibility and norms of the engineering practices.

PO9 **Individual and team work:** Ability to function individually as well as on multi-disciplinary teams.

PO10 **Communication:** Ability to communicate effectively in both verbal and written mode to excel in the career.

PO11 **Project management and finance:** Ability to integrate the knowledge of engineering and management principles to work as a member and leader in a team on diverse projects.

PO12 **Life-long learning:** Ability to recognize the need of technological change by independent and life-long learning.

B. Program Specific Outcomes (PSOs)

PSO1 Develop and Implement computer solutions that accomplish goals to the industry, government or research by exploring new technologies.

PSO2 Grow intellectually and professionally in the chosen field.

K. S. R. COLLEGE OF ENGINEERING, TIRUCHENGODE – 637 215
(Autonomous)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SEMESTER: VII
16CS713 – SOFTWARE TESTING

Unit I
2 marks

1. **Define software Testing.(Remembering)**

Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.

2. Define Validation. (Remembering)

Validation is the process of evaluating a software system or component during, or at the end of, the development cycle in order to determine whether it satisfies specified requirements.

3. Define Verification. (Remembering)

Verification is the process of evaluating a software system or component to determine whether the product of a given development phase satisfy the conditions imposed at the start of that phase.

4. Explain Objective of Software Testing.(understanding)

- Ensures the quality of product
- Defect prevention and detection
- Ready to integration and revise the component
- Provide information to take a decision for the next phase
- Focus on the accurate and reliable result
- Preventing defects
- Gaining confidence in the level of quality

5. Choose the techniques to improve quality of Software Product(evaluating)

- Create a long-term plan for quality improvement, break it in to small steps, and then make changes to achieve goals of each step.
- Give supreme priority of quality in every plans and procedures. ...
- Talk often with your clients.

6. Differentiate between verification and validation? (analyzing)

Verification	**Validation**
1. Verification is the process of evaluating software system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.	1.Verification is the process of evaluating software system or component during or at the end of the , the development phase satisfy the conditions imposed at the start of that phase.
2. Verification is usually associated with activities such as inspections and reviews of the s/w deliverables.	2. Verification is usually associated with Traditional execution _based testing, i.e., Exercising the code with test cases.

7**. What are software testing objectives and purpose? (Remembering)**

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

8. What is software testing strategy? (Remembering)

A software testing strategy is an outline which describes the software development cycle testing approach.

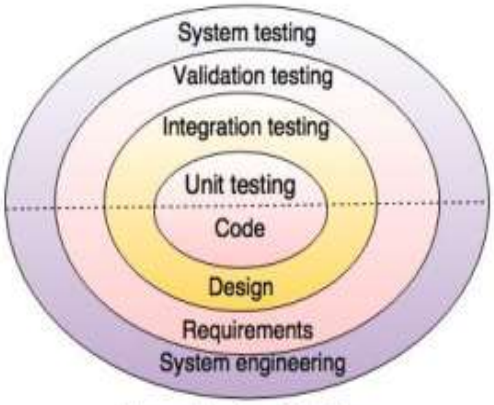
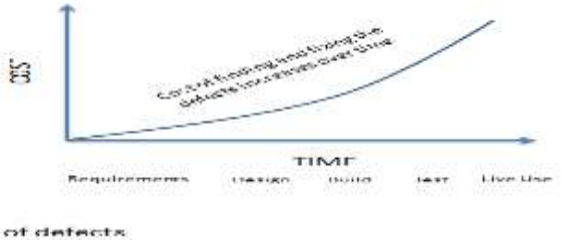


Fig. - Testing Strategy

9. Roles and responsibilities of software tester.(understanding)

- To read all the documents and understand what needs to be tested.
- Based on the information procured in the above step decide how it is to be tested.
- Inform the test lead about what all resources will be required for software testing.
- Develop test cases and prioritize testing activities.
- Execute all the test case and report defects, define severity and priority for each defect.
- Carry out regression testing every time when changes are made to the code to fix defects.

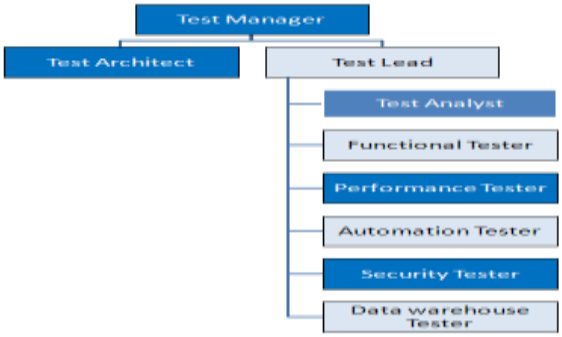
10. what is Cost of Defects? (Remembering)



Cost of defects

Functional and non functional testing
FT- Business process UI screens Integration and data manipulation
NFT-Performance,stability,availability,security,Maintainability,relaiability

11. level structure of the testing team in an organization.(understanding)



12. Roles and responsibilities of Test manager.(understanding)

- Test manager has high level of authority and responsibility in a testing team
- The testing team would work under the guidance and direction of the test manager
    - Planning the test activities and deliverables
    - Estimating and acquiring resources for the budget
    - Assigning task, directing activities and ensuring the testing is complete on time and within budget
    - Controlling the test execution and reporting the progress

- Identifying risks and providing mitigation plan. Escalating issues on timely manner to the higher management

13. Roles and responsibilities of Test architect.(understanding)

- Test architect is responsible to provide the overall testing solution for functional and non-functional requirements to the customer requirements.
  - Understand the customer needs
  - Provide high level approach of framework, solution on functional automation testing and non functional performance and security testing
  - Based on approval develop the framework and solution
  - Work with testing team and customers to explain them the developed solutions and how it needs to be followed in the project

14. Roles and responsibilities of Test lead(evaluating)

- Test lead is responsible to understand the testing requirements, provide test estimates, create and review test cases, review and analyze test results and submit test reports
  - Understand the business requirements in terms of functional and technical requirements
  - Provide resource requirement in terms of hardware , software and staff
  - Work with the team to estimate, plan and execute tests ensuring the team follows the standards, guidelines and processes for the project
  - Review test team deliverables such as test cases, test results, defects
  - Create status reports on the testing activities for the stakeholders

15. Roles and responsibilities of Test analyst.(understanding)

- Understand the requirements
- Create test scenarios
- Test scripts
- Prepare test data
- Execute tests
- Report the defects
- Developing automated test scripts and executing script is also responsibility of the tester if any automation is planned for the project but is based on the technical knowledge and understanding of the tool by the tester.

16.What is Software test metrics? (Remembering)

- Metric is a standard unit of measurement that quantifies results and used for evaluating the software processes, products and services is termed as software metrics
- Software metrics is a measurement based technique which is applied to
  - Processes
  - products
  - Services

17. *What is* Software Testing Life Cycle ? *(Remembering)*

STLC is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product. Let's quickly see what all stages are involved in typical Software Testing Life Cycle (STLC).

18.Software Testing Life Cycle STLC

16 marks

1. Recall the Roles and Responsibilities of a Software Tester. [Remembering]
2. Analyze the purpose of Software Testing.[Analyzing]
3. What is Verification? Explain process of Validating Software Program. [Remembering]
4. List out limitations of Software Testing. [Analyzing]
5. Explain in detail about Verification and Validation[Understanding]
6. Recall the Roles and Responsibilities of Test Engineer[Remembering]

Unit 2

2 marks

1. What are Static Testing Techniques? (Remembering)

1. Document reviews
2. Walkthroughs
3. Inspection
4. Feasibility analysis or any other form of analysis to determine if the software is what it should be or not
5. Code review

2.What are Software Testing Requirements? (Remembering)

- Requirement analysis is the initial phase to discover and identify the detailed business requirements and need for the software system.
- A Software Requirement Specification (SRS) document is a complete description of the behavior of the system to be developed.

3. Discuss Requirement Types(evaluating)

- Functional Requirements
- Non Functional Requirements
- Customer Requirements
- Design Requirements
- Derived Requirements
- Allocated Requirements

4. What are theFunctional Requirements? (Remembering)

- The **high level functional requirements** are documented in the **SRS document** which the team should verify and understand the requirement.
- During the **design phase**, high level functionalities will be elaborated into **architectural and detailed design**
- The **functional design specification** defines all user-perceivable aspects of a system.
- **Functional design documentation** allow the business team to approve the description of each unit process
- It contain sufficient details to allow the project team to proceed with the **low level design**
- Each functional requirement should contain the **data input** for that process/functionality and the **output** that is received from this action.

5. What are Non Functional Requirements? (Remembering)

- Non-Functional requirements are the information about the product performance, Security, reliability, maintainability, Configuration information and so on.

Software Testing Review Process

- The main aim of the software review processes is to monitor the results or output of every phase to determine whether it is complying with the given requirements and standards as defined.
- **There multiple review process within the organization. Few of those are explained in this section.**
    - Software Peer Reviews
    - Technical Reviews
    - Walkthrough
    - Inspection
    - Software Management Reviews

- Software Audits

6.Explain about Software Peer Reviews .(understanding)

- This review will be conducted by **one or more colleagues within the team** who is well versed in the requirement and the domain
- During Requirement Phase, the requirement document can be reviewed by another **analyst in the same team or in external project team** to make sure all the characteristics of the good requirement document are covered.
- During Coding phase the source code developed by the developer should be reviewed by another developer in the team to make sure the developer used the optimized logic in the code and followed all the standards and guidelines for the project.
- Findings from this review are logged in the **review log and shared to the project team**.

7.Discuss about Technical Reviews(evaluating)

- This review is the form of a discussion meeting that focuses on **achieving agreement about the technical content** of a document or code or manual & automation test cases.
- Main goal of this review is to **assess the importance and value of the technical concepts** and alternative ways of doing things in the product and project environments.
- This review emphasis on the project teams to have the **consistencies** in the use of the representation of the technical concepts deployed in the project.
- This is the documented defect detection process where this involves Technical subject matter experts.
- **Technical Subject matter** expert can be a trained moderator who then drives this review till its completion.
- It's up to the reviewer whether to use the **checklists for the review and the review logs** to document the findings from the review.

8.What is Walkthrough? (Remembering)

- This review is mainly to **share the thought processes** applied to the functionality to achieve a common understanding with the reviewers and to gather their feedback.
- This walkthrough is useful for the review of the **high level documents** like architectural design document, SRS document, Algorithm, Logic applied to the code and framework or the automation test scripts.
- The owner of the artifact explains the content to the reviewers and the reasons behind it.
- Reviewers would analyze and identify any alternatives or accept the content as it is if it is the best suit for the solution.

9.Discuss about Inspection(evaluating)

- This review is the **most formal review type**.
- The documents, code or test cases that are to be reviewed were prepared by the authors and reviewed well before the inspection.
- Inspection will typically carry out by **comparing the product with its sources and other references** provided by the project team or the customer and using the rules and checklists.
- During Inspection the reviewer will **verify the content and document the findings.**
- Findings won't be revealed till the inspection is complete, during the discussion phase of this meeting the reviewers will list out the findings.
- Main goal of the Inspection is to make the author **to improve on the quality of the product** under inspection
- These goals can be achieved by creating the common understanding by **explaining the information between the author and the inspection reviewers**

10. What are Review Logs? (Remembering)

The review log would be a standardized document or a spreadsheet maintained at the central repository or it could be a tool used by the organization. This is the log to maintain all review comments and its current status. It is the same checklist used during the review process but the status would change based on the activities performed based on the review comment.

16 marks

1. Explain how Software program is Tested by a Software Tester. [Evaluating]
2. Illustrate in detail about Functional and Non Functional Requirements[Applying]
3. Explain in detail about Inspection Review Process.[Applying]
4. List out the difference between Functional and Non Functional Requirement [Creating]
5. Discuss about different types of Software Testing Review Process. [Creating]
6. Identify the need of Software Testing Review Process. .[Applying]
7. Discuss about Review Log [Creating]

Unit 3

2 marks

1. What are Two Software Testing Techniques? (Remembering)

Black-box testing

- Knowing the specified function that a product has been designed to perform, test to see if that function is fully operational and error free
- Includes tests that are conducted at the software interface
- Not concerned with internal logical structure of the software

White-box testing

- Knowing the internal workings of a product, test that all internal operations are performed according to specifications and all internal components have been exercised
- Involves tests that concentrate on close examination of procedural detail
- Logical paths through the software are tested
- Test cases exercise specific sets of conditions and loops

2. Difference between White Box Testing and Black Box Testing (analyzing)

WHITE BOX TESTING	BLACK BOX TESTING
Related to structural knowledge	Related to functional knowledge
Knowledge of the code is required	No knowledge of code is required
It has to be performed only after the completion of the Designing phase	It can be performed early in the software development i.e. in requirement gathering phase itself.
The performance of the application is not tested.	The performance of the application is tested.
Large number of test cases are required	Only selective number of test cases are required
Performed by the developers.	Performed by the team of professionals
Synonyms of White-Box : structural and glass-box.	Synonyms of Black-Box : behavioral, functional, opaque-box and closed-box

3. What areStatic Testing Techniques? (Remembering)

To summarize, static testing is the verification part of software testing

6. Document reviews
7. Walkthroughs
8. Inspection
9. Feasibility analysis or any other form of analysis to determine if the software is what it should be or not
10. Code review

4. What is Dynamic testing? (Remembering)

- Once the code quality is confirmed by checking the compliance with standards and guidelines, the next step is to execute the code and test the behavior of the code for a given input.

The output is compared with the expected result to see the functional coverage of the code.

5. Difference between Static Testing and Dynamic Testing (analyzing)

NO	Static Testing	Dynamic Testing
1	It's testing of without executing the Software.	It's testing that involves the execution of the Software.
2	In Static Testing software are examined manually and some Static analysis tool used.	In Dynamic Testing software executed by giving set of inputs,examined it's output and compared what is expected.
3	Static Testing can start early in the life cycle.Eg: By Verifying User Requirements.	Dynamic testing can start after development of software components.
4	Types of defect find in Static testing are : Missing requirements, Desgin defect ,Syntax Error etc.	Types of defect find in dynamic testing are : Variables not constant ,checking if output from the expected values.
5	Types of Static Testing : Review ,Inspection , Walk-through.	Types of Dynamic Testing: Unit testing,Integartion testing, System Testing, Acceptance Testing.
6	Static Testing find bug before you compile.	Dynamic testing find bug after compilation, linking.
7	Static Testing is about prevention.	Dynamic Testing is about cure.
8	Static Testing is most cost effective than Dynamic Testing.	Dynamic Testing not Cost effective as compare to Static Testing
9	Static Testing done in the verification stage.	Dynamic Testing done in validation stage.
10	Static Testing gives 100% statement coverage.	Dynamic Testing does not give 100% statement coverage.

6. What is meant by Statement Coverage? (Remembering)

- Statement coverage is the process to identify if all lines of code within the unit under test are in use or any unwanted code
- unwanted code will be removed if found
- This would eliminate number of defects that is leaked to the next phase of development.
- There are multiple tools available to automate the unit testing and to calculate the coverage.
- It is not possible always to get 100 percent coverage
- **Statement Coverage = (Number of statements exercised /Total number of statements)*100**

7. What is meant by Decision Coverage? (Remembering)

- This technique also known as branch coverage and this technique will be used if there are 2 or more possible exists from the statements like IF statement, DO-WHILE, REPEAT-UNTIL and CASE statements
- This technique will make the developer to go through all the decisional paths in the code to make sure the result is properly achieved based on the requirements
- Decision Coverage percentage is calculated using the below formula:
  - **Decision Coverage = Number of decision outcomes exercised * 100%**
- **Total number of decision outcomes**

8. Discuss Black Box Test Techniques (evaluating)

- A Process or Procedure to derive and or select test cases based on the analysis of the specification either functional or non-functional of a component or system without reference to its internal structure.
- two main categories of Black Box testing
  - **Functional testing**
  - **Non-functional testing**

9. What is equivalence partitioning?(Remembering)

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.

10.What is boundary value analysis?(Remembering)

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.

11. What is positive testing? (Remembering)

Positive testing tries to prove that a given products does what is is supposed to do. When a test case verifies the requirements of the product with a set of expected output, it is called positive test case. The purpose of positive testing is to prove that the product works as per specifications and expectations. A product delivering an error when it is expected to given an error, is also a part of positive testing.

12. What is Negative testing ? (Remembering)

Negative testing is done to show that the product does not fail when an unexpected input is given. The purpose of negative testing is to try and break the system. negative testing focuses on test conditions that lie outside the specification.

13.What are DECISION TABLES? (Remembering)

- List of decision variables The variables related to the decision are given as the columns of a decision table. However, when more number of decision variables or less number of distinct combinations of variable, these may be given as rows.
- Conditions set for every decision variable All the conditions assumed for every decision variable is continuously given one after the other.
- Actions to be done in each combination of the conditions. The actions are listed as last column of the decision tables. If the value of decision variable does not affect the outcome of decision then entries are marked as "-". These are said to be "don't cares". It minimizes the total number of tests to be done.

14.What is STATE BASED TESTING ? (Remembering)

State or graph based testing is very useful in situations where,

1. The product under test is a language processor (eg. Compiler)
2. Workflow modeling
3. Dataflow modeling

15. What is COMPATABILITY TESTING? (Remembering)

Testing done to ensure that the product features work consistently with different infrastructure components is called compatibility testing.

The parameters that generally affect the compatibility of the product are,

- Processor and the number of processor (p3,p4, xenon etc)
- Architecture and characteristics of the machine (32 bit, 64 bit…)
- Resource availability on the machine (RAM, disk space, network card..)
- Equipment that the product is expected to work (printers, modem, router….)
- Operating system and its services
- Backend components such as database servers (oracle, Sybase…)
- Any software used to generate product binaries (compiler, linker,…)

16.Discuss about DOCUMENTATION TESTING? (evaluating)

User documentation covers all the manuals, user guides, installation guides, setup guides, read me file, software release notes and online help that are provided along with the software to help the end user to understand the software system. Objectives of user documentation testing • To check if what is stated in the document is available in the product • To check if what is there in the product is explained correctly in the document

17. What is STRUCTURAL TESTING? (Remembering)

Structural testing takes into account the code, code structure, internal design and how they are coded. The main difference between structural testing and static testing is that in structural testing tests are actually run by the computer on the built product.

18.What is CODE(UNIT) FUNCTIONAL TESTING? (Remembering)

- Initially the developer can perform certain static testing. This can be quick test that checks out any obvious mistakes. By repeating these tests for multiple values of input variables, the confidence level of the developer to go to the next level increases.
- For modules with complex logic or conditions, the developer can build a "debug version" of the product by putting intermediate print statement. This is to make sure the program is passing through the right loops and iterations the right number of times.
- Another approach is to run the product under a debugger or an Integrated Development Environment. These tools allow single stepping of instructions, setting break points at any function or instruction. Then to view the various system parameters or program variable values.

16 marks

1. Explain about black box testing[Evaluating]
2. What are the requirement based testing available? .[Understanding]
3. Explain about white box testing.[Understanding]
4. Discuss about Positive and Negative Testing - Boundary Value Analysis- Decision Table[Evaluating][
5. What are Equivalence Partitioning ,State Based Testing? .[Understanding]
6. Discuss about Compatibility testing –and User Documentation Testing.[Analyzing]
7. Explain White Box Testing: .[Understanding]
    - Structural Testing
    - Code Functional Testing
    - Code Coverage Testing
    - Code Complexity Testing.

Unit 4 - Test Data Management and Defects Management

1) Write the different types of goals?(Remembering)

A. Business goal
B. Technical Goal
C. Business/technical Goal
D. Political Goal

2) Differentiate between Goal and Policy.(Remembering)

- A goal can be described as a statement of intent and a statement of an accomplishment that an individual or an org wants to achieve.
- A Policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an org.

3) Define Plan & Milestone .(understanding)

- A plan is a document that provides a framework or approach for achieving a set of goals.
- Milestones are events that are expected to occur at a certain time in the Project's Lifetime span. Managers use them to determine project status.

4) Define a Work Breakdown Structure.(WBS) .(understanding)

A Work Breakdown Structure (WBS) is a hierarchical like representation of all the tasks that are required to complete a project.

5) List out the Test plan components. (Remembering)

1. Test plan identifier
2. Introduction
3. Items to be tested
4. Features to be tested
5. Approach
6. Pass/fail criteria
7. Suspension and resumption criteria
8. Test deliverables

9. Testing Tasks
10. Test environment
11. Responsibilities
12. Staffing and training needs
13. Scheduling
14. Risks and contingencies
15. Testing costs
16. Approvals.

6) Write the approaches to test cost Estimation? (Remembering)

1. The COCOMO model and heuristics
2. Use of test cost drivers
3. Test tasks
4. Tester/developer ratios
5. Expert judgment

7) Write the Work breakdown Structure elements for testing. (Remembering)

1. Project startup
2. Management coordination
3. Tool selection
4. Test planning
5. Test design
6. Test development
7. Test execution
8. Test measurement, and monitoring
9. Test analysis and reporting

8) Define Test incident Report.(understanding)

The tester should record in attest problem report any event that occurs during the execution of the tests that is unexpected , unexplainable, and that requires a follow- up investigation.

9) Define Test Log. .(understanding)

The Test log should be prepared by the person executing the tests. It is a diary of the events that take place during the test. It supports the concept of a test as a repeatable experiment.

10) What are the Three critical groups in testing planning and test plan policy?(analyzing)

1. Managers
2. Testers/Developers
3) Users/Clients

11) Define Procedure. (understanding)

A procedure in general is a sequence of steps required to carry out a specific task.

12)What are the skills needed by a test specialist? (Remembering)

- Personal and managerial Skills , Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively.
- Technical Skills : General software engineering principles and practices, understanding of testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.

13) Write the test term hierarchy? (Remembering)

1) Test Manager
2) Test leader
3) Test Engineer
4) Junior Test Engineer.

14) Define Breaking the System. .(understanding)

The goal of stress test is to try to break the system , Find the circumstances under which it will crash. This is sometimes called “breaking the system”.

15) Explain what Test Deliverables is? .(understanding)

- Test Deliverables are a set of documents, tools and other components that have to be developed and maintained in support of testing.
- There are different test deliverables at every phase of the software development lifecycle

- Before Testing
- During Testing
- After the Testing

16) How will you conduct Risk Analysis? (Remembering)

For the risk analysis following steps need to be implemented

- Finding the score of the risk
- Making a profile for the risk
- Changing the risk properties
- Deploy the resources of that test risk
- Making a database of risk

17. What are the categories of debugging? (Remembering)

1. Brute force debugging
2. Backtracking
3. Cause elimination
4. Program Slicing
5. Fault tree analysis

18.What is the common risk that leads to project failure? (Remembering)

The common risk that leads to a project failure are

1. Not having enough human resource
2. Testing Environment may not be set up properly
3. Limited Budget
4. Time Limitations

19.What is test management review and why it is important? (Remembering)

Management review is also referred to as Software Quality Assurance or SQA. SQA focusses more on the software process rather than the software work products. It is a set of activities designed to make sure that the project manager follows the standard process. SQA helps test manager to benchmark the project against the set standards.

20. Mention what are the categories of defects? (Remembering)

Mainly there are three defect categories

- **Wrong**: When a requirement is implemented incorrectly
- **Missing**: It is a variance from the specification, an indication that a specification was not implemented or a requirement of the customer is not met
- **Extra**: A requirement incorporated into the product that was not given by the end customer. It is considered as a defect because it is a variance from the existing requirements

Mention what the difference between a "defect" and a "failure" in software testing is?

- In simple terms when a defect reaches the end customer, it is called a failure while the defect is identified internally and resolved; then it is referred to as a defect.

21. Difference between latent and masked defect. .(understanding)

- **Latent defect:** A latent defect is an existing defect that has not caused a failure because the sets of conditions were never met
- **Masked defect:** It is an existing defect that has not caused a failure because another defect has prevented that part of the code from being executed

22.Mention what the basic components of defect report format are? (Remembering)

The essential components of defect report format include

- Project Name , Module Name
- Defect detected on , Defect detected by
- Defect ID and Name , Snapshot of the defect
- Priority and Severity status
- Defect resolved by , Defect resolved on

23. Define : Test Data Management(Remembering)

A process of planning, designing, storing, managing the software or the source codes of an application for testing is called Software test data management

24. Mention any four Test Data Management Tools.(understanding)

- Informatica , DATPROF , CA Test Data Manager (Datamaker) , Compuware's , HP

Mention the Challenges of Test Data Management .(understanding)

- Additional time for data set up/management instead of actual testing
- Additional administrative efforts in test data management
- Additional expense including personnel and hardware
- Inaccurate/difficult to access data negatively impacts testing
- Sensitivity of private information (credit cards, medical records, etc.)

26.What is the major difference between Error, Defect, and Failure? (Remembering)

- Error: If the developers find that there is a mismatch in the actual and expected behavior of an application in the development phase then they call it as an Error.
- Defect: If testers find a mismatch in the actual and expected behavior of an application in the testing phase then they call it as a Defect.
- Failure: If customers or end users find a mismatch in the actual and expected behavior of an application in the production phase then they call it as a Failure.

PART – B

1. Why is testing planning important for developing a repeatable and managed testing process? Discuss.(analzing)

2. Why is it so important to integrate testing activities into the software life cycle? Explain. (understanding)

3. What role do managers play in support of a test group? Explain the steps involved in preparing a test plan in detail .(understanding)

4. Discuss in detail about resource needed for testing and how do you estimate the resource utilization. (analzing)

5 Elaborate on Test infrastructure and people management.(understanding)

6. What is Test Plan? What is the information that should be covered in Test Plan? Explain. (Remembering)

- A test plan can be defined as a document describing the scope, approach, resources, and schedule of testing activities and a test plan should cover the following details.
  - Test Strategy
  - Test Objective
  - Exit/Suspension Criteria
  - Resource Planning
  - Test Deliverables

7.What does the Test Plan consist of? Discuss . (or) Elaborate the components of a test plan. (Remembering)

Test design, scope, test strategies, approach are various details that Test plan document consists of.

- Test case identifier & Scope
- Features to be tested
- Features not to be tested
- Test strategy & Test approach
- Test deliverables
- Responsibilities
- Staffing and training
- Risk and Contingencies

8.List and discuss the things you have to consider while monitoring your project? (Remembering)

The things that have to be taken in considerations are

- Is your project on schedule
- Are you over budget
- Are you working towards the same career goal
- Have you got enough resources
- Are there any warning signs of impending problems
- Is there any pressure from management to complete the project sooner

9. Elaborate the processing steps for Test Data Management (TDM) tool that follows ? (Remembering)

- In any system, data is stored in different formats, types, and locations. Different rules are applied to this data. Hence, the test tool finds out the appropriate test data from these data for the testing process.
- Now the tool extracts the subset of data from the selected test data collected from multiple data sources.
- After selecting the subset test data, test tool uses masking for sensitive test data, such as a client's personal information.
- Now the tool performs the comparison between the actual data and baseline test data to check the accuracy of the application.
- To increase the efficiency of the application, the tool refreshes the test data.

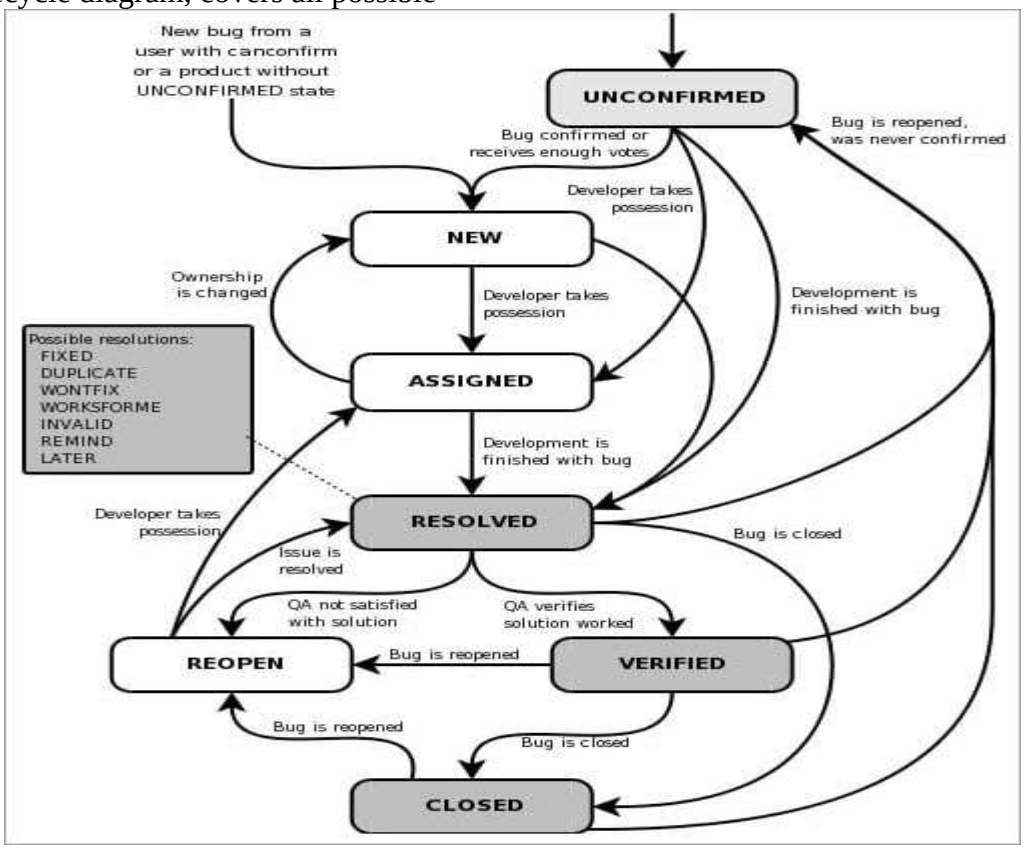
10. Tabulate the *steps involved / key phases* involved in a TDM process ? (Remembering)

Phase	Steps Involved
Planning	1. Assign Test Data Manager (TDM) 2. Define data requirements and templates for data management 3. Prepare documentation including list of tests and data landscape reference 4. Establish a service level agreement 5. Set up the test data management team 6. Appropriate plans and papers signed off
Analysis	1. Initial set up and synch exercises involve data profiling for each individual data store assignment/recording of version numbers for existing data in all environments 2. Collection/consolidation of data requirements 3. Update project lists 4. Analyze data requirements and latest distribution log 5. Asses for gaps and impact of data modification 6. Define data security, back up, storage, and access policy 7. Prepare reports
Design	1. Decide strategy for data preparation 2. Identify regions needing data to be loaded/refreshed 3. Identify appropriate methods 4. Identify data sources and providers 5. Identify tools 6. Data Distribution plans 7. Coordination/communication plan 8. Test activities plan 9. Document for data plan
Build	1.Execute plans 2. Execute masking/de-identification where applicable 3. Back up data 4. Update logs
Maintenance	1.Support change requests, unplanned data needs, problems/incidents 2. Prioritize requests where applicable 3. Analyze requirements and consider if they can be met from existing/modified current data including data assigned to other projects 4. Required data modification **5**. Back up new data **6.** Assign version markers and log with appropriate description **7.** Review status of ongoing projects **8.** Data profile exercises **9.** Assess/address gaps **10.** Refresh data where needed **11.** Schedule and communicate maintenance **12.** If necessary, redirect requests **13.** Documentation and reports

11. Explain the various types of defects in Testing and Elaborate on Defect Life Cycle . (Remembering)

12.What is Defect Life Cycle? Sketch and discuss the Defect/Bug Life Cycle in Software Testing. **(Remembering)**

- Defect Life Cycle or Bug Life Cycle is the specific set of states that a Bug goes through from discovery to defect fixation.
- Bug Life Cycle Status : The number of states that a defect goes through varies from project to project.
- Below lifecycle diagram, covers all possible



13. Describe about the Test Data Management (TDM) in Functional and Performance testing.(understanding)

Unit V – AUTOMATION TESTING

Part A

1.What is Automation Testing? (Remembering)

Automation testing or Test Automation is a process of automating the manual process to test the application/system under test. Automation testing involves the use of a separate testing tool which lets you create test scripts which can be executed repeatedly and doesn't require any manual intervention.

2.What are the benefits of Automation Testing? (Remembering)

Benefits of Automation testing are:

1. Supports execution of repeated test cases
2. Aids in testing a large test matrix
3. Enables parallel execution
4. Encourages unattended execution
5. Improves accuracy thereby reducing human-generated errors
6. Saves time and money

3.Why should Selenium be selected as a test tool?

1. free and open source
2. large user base and helping communities
3. cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)
4. great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. fresh and regular repository developments
7. supports distributed testing

4.What all things you should consider before selecting automation tools for the AUT? (Remembering)

- Technical Feasibility
- Complexity level
- Application stability
- Test data
- Application size
- Re-usability of automated scripts
- Execution across environment

5.In manual testing , what are stubs and drivers? .(analzing)

- Both stubs and drivers are part of incremental testing. In incremental testing, there are two approaches namely bottom-up and top-down approach. Drivers are used in bottom-up testing and stub is used for a top-down approach. In order to test the main module, the stub is used, which is a dummy code or program.

6.What are the testing types that can be supported by Selenium? (Remembering)

Selenium supports the following types of testing:

1. Functional Testing
2. Regression Testing

7.What are the limitations of Selenium? (Remembering)

Following are the limitations of Selenium:

- Selenium supports testing of only web-based applications
- Mobile applications cannot be tested using Selenium
- Captcha and Barcode readers cannot be tested using Selenium
- Reports can only be generated using third-party tools like TestNG or JUnit.
- As Selenium is a free tool, thus there is no ready vendor support through the user can find numerous helping communities.
- The user is expected to possess prior programming language knowledge.

8.What is Selenese? When should I use Selenium IDE? (Remembering)

- Selenese is the language which is used to write test scripts in Selenium IDE.
- Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

9.What are the different types of locators in Selenium? (Remembering)

The locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium: ID ,ClassName , Name , TagName , LinkText ,PartialLinkText , Xpath , CSS Selector , DOM

10.What is an XPath? (Remembering)

XPath is used to locate a web element based on its XML path. XML stands for Extensible Markup Language and is used to store, organize and transport arbitrary data. It stores data in a key-value pair which is very much similar to HTML tags.

11.How do I launch the browser using WebDriver? .(analzing)

*WebDriver driver = **new** FirefoxDriver();* *WebDriver driver = **new** ChromeDriver();*
*WebDriver driver = **new** InternetExplorerDriver();*

12. What are the different types of waits available in WebDriver? .(analzing)

There are two types of waits available in WebDriver:

1. Implicit Wait
2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, the subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

13.What is a framework? (Remembering)

The framework is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity, reporting mechanism, test data injections etc. to pillar automation testing.

14.What are the advantages of the Automation framework? (Remembering)

- Reusability of code
- Maximum coverage
- Recovery scenario
- Low-cost maintenance
- Minimal manual intervention
- Easy Reporting

15.How to Choose an Automation Tool? .(analzing)

Selecting the right tool can be a tricky task. Following criterion will help you select the best tool for your requirement-

- Environment Support
- Ease of use
- Testing of Database
- Object identification
- Image Testing
- Error Recovery Testing
- Object Mapping

16.Which Test Cases are to be Automated? (Remembering)

Test cases to be automated can be selected using the following criterion to increase the automation ROI

- High Risk - Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming

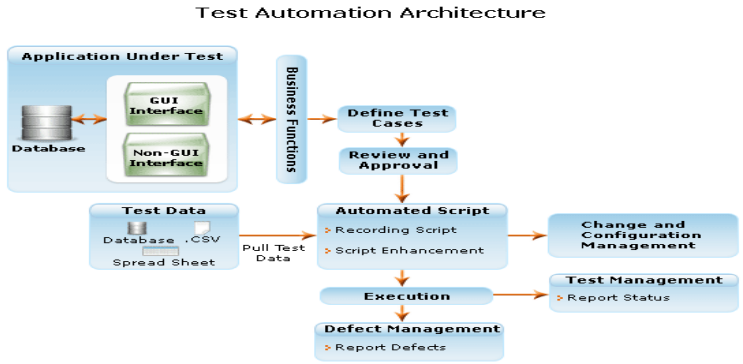
Part B

1. Explain in a testing project what testing activities would you automate? (Remembering)

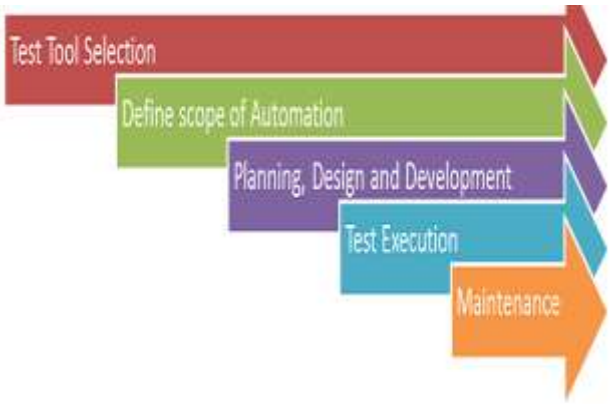
In testing project testing activities, you would automate are

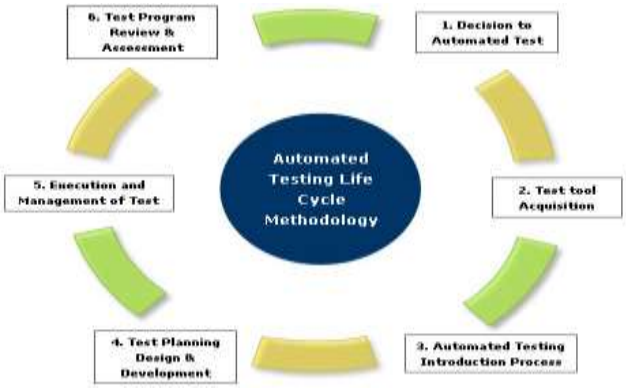
- Tests that need to be run for every build of the application
- Tests that use multiple data for the same set of actions
- Identical tests that need to be executed using different browsers
- Mission critical pages
- A transaction with pages that do not change in a short time

2.With an neat diagram , explain the design and architecture for Test automation. (Remembering)



3.Sketch and discuss the process model of Automation. **(Remembering)**





4.Tabulate the difference between Selenium IDE, Selenium RC, and WebDriver? .(analzing)

Selenium IDE	Selenium RC	Selenium WebDriver
It only works in Mozilla browser.	It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc.	It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc.
It supports Record and playback	It doesn't supports Record and playback	It doesn't supports Record and playback
Doesn't required to start server before executing the test script.	Required to start server before executing the test script.	Doesn't required to start server before executing the test script.
It is a GUI Plug-in	It is standalone java program which allow you to run Html test suites.	It actual core API which has binding in a range of languages.
Core engine is Javascript based	Core engine is Javascript based	Interacts natively with browser application
Very simple to use as it is record & playback.	It is easy and small API	As compared to RC, it is bit complex and large API.
It is not object oriented	API's are less Object oriented	API's are entirely Object oriented
It doesn't supports of moving mouse cursors.	It doesn't supports of moving mouse cursors.	It supports of moving mouse cursors.
Need to append full xpath with 'xpath=\\' syntax	Need to append full xpath with 'xpath=\\' syntax	No need to append full xpath with 'xpath=\\' syntax
It does not supports listeners	It does not supports listeners	It supports the implementation of listeners
It does not support to test iphone/Android applications	It does not support to test iphone/Android applications	It support to test iphone/Android applications

5.What is framework ?What are the different types of frameworks for automation. (Remembering)

1. **Module Based Testing Framework:** The framework divides the entire "Application Under Test" into the number of logical and isolated modules. For each module, we create a separate and independent test script. Thus, when these test scripts have taken together builds a larger test script representing more than one module.
2. **Data Driven Testing Framework**: Data Driven Testing Framework helps the user segregate the test script logic and the test data from each other. It lets the user store the test data into an external database. The data is conventionally stored in "Key-Value" pairs. Thus, the key can be used to access and populate the data within the test scripts.
3. **Keyword Driven Testing Framework:** The Keyword Driven testing framework is an extension to Data-driven Testing Framework in a sense that it not only segregates the test data from the scripts, it also keeps the certain set of code belonging to the test script into an external data file.
4. **Hybrid Testing Framework:** Hybrid Testing Framework is a combination of more than one above mentioned frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.
5. **Behavior Driven Development Framework:** Behavior Driven Development framework allows automation of functional validations in an easily readable and understandable format to Business Analysts, Developers, Testers, etc.