MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

SUMMER – 2024 EXAMINATION Model
Answer – Only for the Use of RAC Assessors

Subject Name: Software Testing **Subject Code:** 22518

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**Enlist objectives of software testing.**	**2 M**
	Ans	1. Finding Errors:Testing is process of executing a program with an intention of finding an error. 2. Creating good test cases: A good test case is one that has a high probability of finding yet undiscovered error. 3. Quality Improvement : Defects are fixed by the developer, so quality is improved. 4. Satisfying customer requirements: Testing demonstrates to the customer that software works properly as per specification.	½ M for each objective
	b)	**Compare Alpha testing and Beta testing. (Any two differences).**	**2 M**

8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

	Ans	<table><tr><th>Alpha Testing</th><th>Beta Testing</th></tr><tr><td>Testing is done by end user at developer site.</td><td>Testing is done by end user at end user site.</td></tr><tr><td>It is done in controlled environment.</td><td>Developer is not present so not done in controlled environment.</td></tr><tr><td>Alpha testing is not effective as it is done in controlled environment.</td><td>Beta test is more effective as end user do not have any restriction of testing software.</td></tr><tr><td>End user may have to test software under influence of developer.</td><td>Beta testing is live application testing, not done in influence of developer.</td></tr></table>	1 M for each valid point
	c)	**Define a test plan.**	**2 M**
	Ans	A **Test Plan** is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product. Test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager.	2 M for correct definition
	d)	**List any four skills of software tester.**	**2 M**
	Ans	1) Analytics skills 2) Communication skills 3) Knowledge of test management tools 4) Negotiation skills	½ M for each skill
	e)	**State the classification of defects.**	**2 M**

	Ans	**1. Requirement/Specification Defects:** Requirement-related defects arise in a product when one fails to understand what the customer requires. These defects may be due to the customer gap, where the customer is unable to define his requirements. Producer gap, where the developing team is not able to make a product as per requirements. **2. Design Defects:** Design defects occur when system components, interactions between system components, interactions between the outside software/hardware, or users are incorrectly designed. Design defects generally refer to the way of design creation or its usage while creating a product.	½ M for each classification

		3. Coding Defects: This defect arises when variables are not initialized properly, or variables are not declared correctly, or database is not created properly. Coding also needs adequate commenting to make it readable and maintainable in future. **4. Testing Defects:** These would encompass incorrect, incomplete, missing inappropriate test cases and test procedures.	
	f)	**State the need of automated testing tool. (Any two).**	**2 M**

	Ans	Need for automated testing tools are: • An automated testing tool can playback pre-recorded and predefined actions, compare the results to the expected behavior and report the success or failure of these to a test engineer. • Once automated tests are created, they can easily be repeated, and they can be extended to perform tasks impossible with manual testing. • Automated Software Testing Saves Time and Money. • Software tests must be repeated often during development cycles to ensure quality. • Every time source code is modified software tests should be repeated. For each release of the software, it may be tested on all supported operating systems and hardware configurations. Manually repeating these tests is costly and time consuming. • Once created, automated tests can be run repeatedly at no additional cost, and they are much faster than manual tests. • Testing Improves Accuracy, Even the most conscientious tester will make mistakes during monotonous manual testing. Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results. • They can even be run on multiple computers with different configurations. • Automated software testing can look inside an application and see memory contents, data tables, file contents, and internal program states to determine if the product is behaving as expected.	Any four needs: ½ M for any 4 needs.
	g)	**Define the terms error and defect in relation with software testing.**	**2 M**
	Ans	**Error**: An error is a human action that produces the incorrect result. **Defect:** A defect is an error or a bug in the application which is created. A programmer while designing and building the software can make mistakes or errors. These mistakes or errors mean that there are flaws in the software. These are called defects.	1 M for each definition
2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Write any four differences between quality assurance and quality control.**	**4 M**

	Ans		1 M for each valid point

Quality Assurance	Quality Control
Process oriented activities.	Product oriented activities.
QA is the process of managing for quality.	QC is used to verify the quality of the output.
They measure the process, identify the deficiencies/weakness and suggest improvements.	They measure the product, identify the deficiencies/weakness and suggest improvements.
SQA is a set of activities for ensuring quality in software engineering processes (that ultim ately result in quality in software products). The activities establish and evaluate the processes that produce products.	SQC is a set of activities for ensuring quality in software products. The activities focus on identifying defects in the actual produc ts produced.
Activities of QA are Process Definition and Implementation, Audits and Training	Activities of QC are Reviews and Testing
It includes Preventi on oriented activities .	It includes detection -oriented activities .
Verification is an example of QA	Validation/Software Testing is an example of QC
QA is a managerial tool	QC is a corrective tool

	b)	**Explain GUI testing with example.**	**4 M**

	Ans	**GUI Testing** There are two types of interfaces for a computer application. 1.Command Line Interface is where you type text and the computer responds to that command. 2.GUI stands for Graphical User Interface where you interact with the computer using images rather than text. • GUI testing is the process of testing the system's Graphical User Interface of the Application Under Test. • GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc. • GUI is what the user sees. A user does not see the source code. The interface is visible to the user. • Especially the focus is on the design structure, images that they are working properly or not. **GUI Testing Guidelines or traits:** 1. Check Screen Validations	2 M for explanation and 2 M for example.

		2. Verify All Navigations 3. Check usability Conditions 4. Verify Data Integrity 5. Verify the object states 6. Verify the date Field and Numeric Field Format **Example: Prerequisite**: User should be on www.flipkart.com	

Test Case No	**Test Case Name**	**Actual Input**	**Expected Output**	**Actual Output**	**Status**

			TC-1	Already have an account link	Click on already have an account link	Link should be clickable. It should go to login page	Link is clickable. It is going to login page	Pass	
			TC-2	Login page	Click on already have an account link	It should show mobile number /email Id field in form	It is showing mobile number /email Id field in form	Pass	
			TC-3	Password	Enter password =”123456”	It should show password in encrypted form	It is showing password in encrypted form	Pass	
			TC-4	Language option	1.click on login option 2.click on hindi link	It should show page content in hindi	It is showing page content in hindi	Pass	
			TC-5	Placement of text box and button on page	1.click on login option	It should show mobile number /email Id, password field and submit button in form on proper position	It is showing mobile number /email Id, password field and submit button in form on proper position	Pass	

TC-6	Navigation	1.click on login option 2.Go to mobile number text box	It should convert arrow into cursor	It is converting arrow into cursor	Pass		

	c)	**Describe the contents of "Test Summary Report' used in test reporting.**	**4 M**
	Ans	**Test summary Report**: The final step in a test cycle is to recommend the suitability of a product for release. A report that summarizes the result of a test cycle is the test summary report. **There are two types of test summary report:** 1. Phase wise test summary, which is produced at the end of every phase. 2. Final test summary report, which has all the details of testing done by all phases and teams. A summary report should be presented. 1.Test Summary Report Identifier 2 Description: Identify the test items being reported in this report with test id. 3 Variances: Mention any deviation from test Solution, test procedures, if any. 4 Summary of results should include: Tests that failed with any root cause description and severity of impact of defect found by tests. 5 Comprehensive assessment and recommendation for release should include: "Fit for release" assessment and Recommendation of release.	Correct explanation 4 M
	d)	**Differentiate between static and dynamic testing tools. (any four points)**	**4 M**

	Ans	**Static Techniques**	**Dynamic Techniques**	1 M each for 4 points.
		Testing that is done without physically executing a program or system.	Testing in which system components are physically executed to identify defects.	
		No execution of code, product, documentation.	Execution of test cases is an example of a dynamic testing technique.	

			Helps in establishing conformance to requirements view	Dynamic testing is a validation technique which includes dummy or actual execution of work products to evaluate it with expected behavior.		
			Checking the software product and related artifacts without executing them.	It includes black box testing ing and methodology such as system test structural white box testing.		
			The work product is reviewed by the reviewer with the help of a checklist, standards, any other artifact, knowledge, and experience, to locate the defect with respect to the established criteria.	The testing methods evaluate ents the product with respect to k it as requirem defined; designs created and ma 'pass' or 'fail'.		
			It may include reviews, walkthroughs, inspection, and audits.	This technique establishes 'fitness for use' view.		
3.		**Attempt any THREE of the following:**				**12 M**
	a)	**Distinguish between white box testing and black box testing. (any four points)**				**4 M**

Ans

White box testing	Black Box Testing
The tester needs to have the knowledge of internal code or program.	This technique is used to test the software without the knowledge of internal code or program
It aims at testing the structure of the item being tested.	It aims at testing the functionality of the software
It is also called structural testing, clear box testing, code-based testing, or glass box testing.	It also knowns as data driven, closed box testing, data-, and functional testing.
Testing is best suited for a lower level of testing like Unit Testing or Integration testing.	This type of testing is ideal for higher levels of testing like System Testing, Acceptance testing.
Statement Coverage, Branch coverage, and Path coverage are White Box testing techniques.	Equivalence partitioning, Boundary value analysis are Black Box testing technique

1 M for each point

			Can be based on detailed design documents.	Can be based on Requirement specification document	
	b)	**Explain the need of test deliverables & test plan for test planning.**			**4 M**
	Ans	**The need of test deliverables for test planning include:**			Need of Test

The deliverables include the following,	
The test plan	Helpful for tester
Test case Specification	Details needed for testing
Test design specification documents	Helpful in designing test
Testing Strategy	Approach to follow testing
Testing Scripts/ procedures	Need to be followed
Test data	Data useful during testing
Test Incident report	Details of situation where testing performed
Test Traceability matrix	Metrix to follow testing
Test results /Reports	Entire report of testing
Install/Configuration guides	Provides guidelines before testing
Test logs produced	Useful for future testing
Defect Report/ Release report	After completion of test this report is generated/prepared

deliverables 2 M

Need of Test plan 2 M

Test Plan A test plan is a systematic approach to testing a system, i.e. software. The plan typically contains a detailed understanding of what the eventual testing workflow will be.

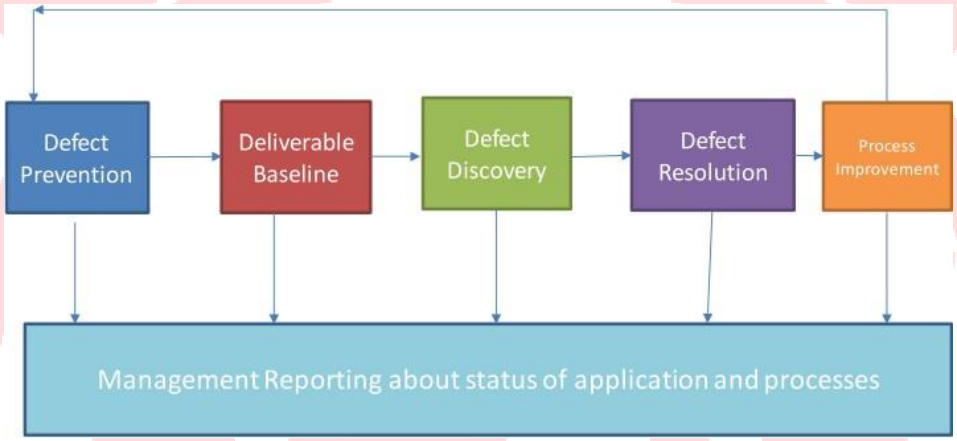
Need of test plan:

- Test Plan Ensures all Functional and Design Requirements are implemented as specified in the documentation.
- To provide a procedure for Unit and System Testing.
- To identify the documentation process for Unit and System Testing.
- To identify the test methods for Unit and System Testing.
- Planning Activities like:

1. Preparing test plan
2. Scope management
3. Deciding Test approach/ strategy
4. Setting up criteria for testing
5. Identifying responsibilities, staffing & Training needs:
6. Identifying Resource Requirement

7. Identifying Test Deliverables
8. Testing task

	c)	**Explain defect management process with suitable diagram.**	**4 M**

	Ans	**Defect Management Process** • Defects found during Verification and validation process in SDLC must be recorded so that it helps in further analysis and root causes of the defect. • The defects found during these phases are used to find weak areas of project /process so that action can be initiated to strengthen it.  **Defect Management Process: Prevention** • It is a process of improving quality and productivity by preventing the defects into a software product. • It is virtually impossible to eliminate the defects altogether. • Implementation of techniques, methodology and standard processes are used to reduce the risk of defects. • Defect prevention is intended to remove the possibility of any defects before it occurs. **Defect Management Process: Deliverable Baselining** • Once the defect is fixed, retested and found to be closed, the product is created again. • If the newly created product satisfies the acceptance criteria, it is base lined. • Only base lined work products can go to the next stage. **Defect Management Process: Defect Discovery** • A defect is said to be discovered when it is brought to the attention of the developers and acknowledged (i.e., "Accepted") to be valid one.	Diagram- 1 M, Explanation- 3 M

- Team should find defects before they become major problems. As soon as a team finds the defects, they should report them so that those can be resolved.

- Team should also make sure that defects should be acknowledged by developers and should bevalid one.

Defect Management Process: Defect Resolution

- Work by the development team to prioritize, schedule and fix a defect, and document the resolution.
- This also includes notification back to the tester to ensure that the resolution is verified

Defect Management Process: Process Improvement

- All problems are due to failures in the process involved in creating software.
- Defects give an opportunity to identify the problem with process used and update them
- Better processes mean better products with less defect.

Defect Management Process: Management Reporting

- Analysis and reporting of defect information to assist management with risk management, process improvement and project management.

d) Give any four differences between manual and automated testing. (Any 4 points) — **4 M**

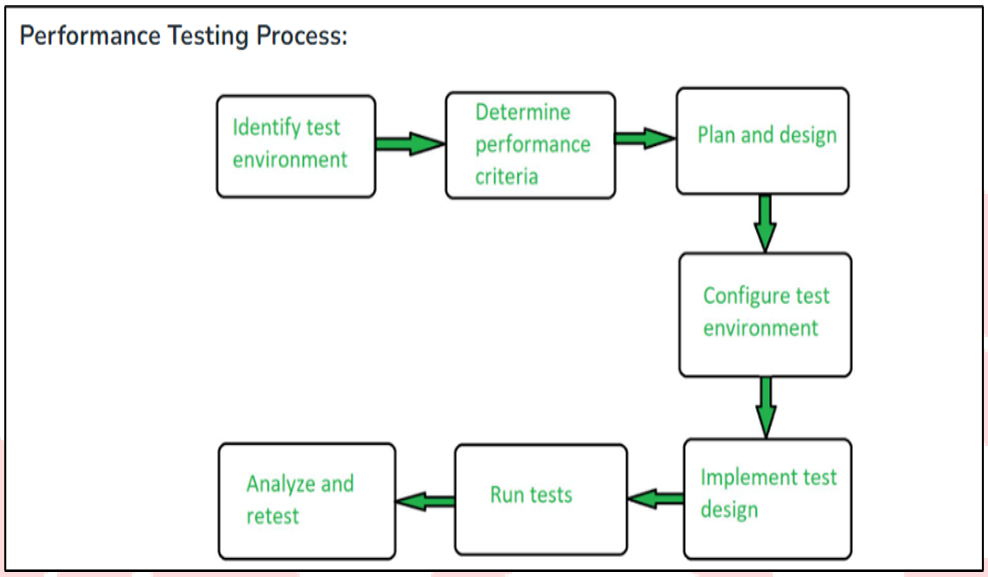
Ans

Manual testing	Automation Testing
Test cases are executed manually	Test cases are executed with the help of tools
Time required to execute test cases is high	Time required to execute test cases is low
Initial investment required for manual testing is lower	Initial investment for automation testing is higher
Manual testing won't be as accurate as it includes human errors	Automation testing can be accurate as it is performed by tools and scripts

1 M for each point

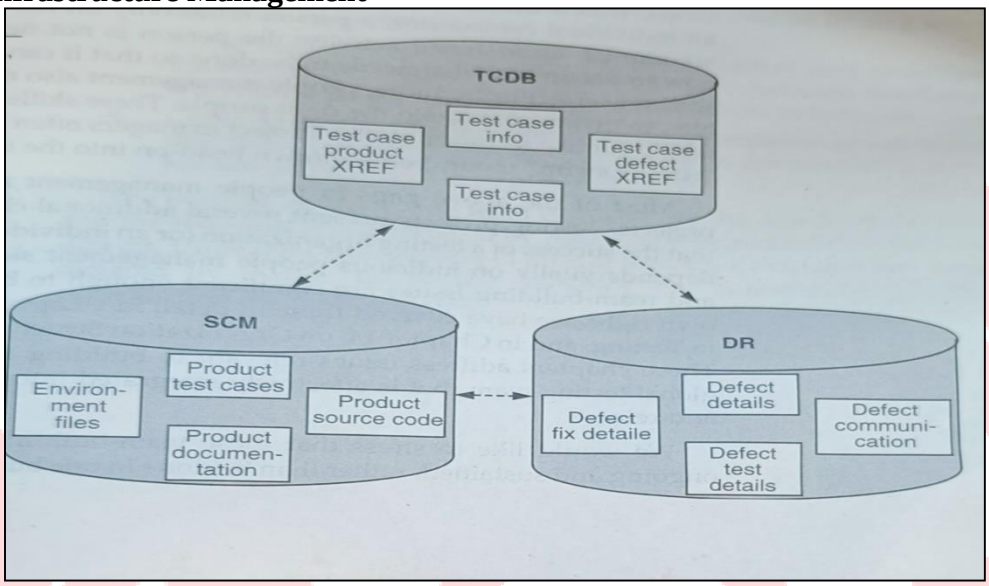
		It provides human observation, can be used to assess user friendliness and customer experience	Cannot guarantee user friendliness or good customer experience	
		It is suitable for almost any software product	It is suitable only for stable systems and used mainly for regression.	

4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**State the process of performance testing.**	**4 M**

	Ans	 1. Identify the Testing Environment: Identifying the hardware, software, network configurations and tools available allows the testing team to design the test and identify performance testing challenges early on. 2. Determine Performance criteria: In addition to identifying metrics such as response time, throughput and constraints, identify what are the success criteria for performance testing. 3. Plan and Design Performance Tests: Identify performance test scenarios that take into account user variability, test data, and target metrics. This will create one or two models. 4. Configure the Test Environment: Prepare the elements of the test environment and instruments needed to monitor resources. 5. Implement Your Test Design: Develop the tests. 6.Execute Tests: In addition to running the performance tests, monitor and capture the data generated.	1 M for diagram 3 M for explanation

		7. Analyze, Report, Retest:	

		Analyze the data and share the findings. Run the performance tests again using the same parameters and different parameters.	
	b)	**Explain people management in test planning.**	**4 M**
	Ans	People management in test planning involves coordinating and leading a team of testers to ensure that the testing process is efficient, thorough, and effective. Below listed are some key aspects of people management in test planning: 1. **Team Composition**: Assemble a diverse team with complementary skills and experiences. Consider factors like technical expertise, domain knowledge, and testing methodologies. 2. **Clear Objectives**: Communicate the objectives of the testing phase clearly to your team. Ensure everyone understands the goals, scope, and expected outcomes of the testing effort. 3. **Assigning Roles and Responsibilities**: Clearly define roles and responsibilities within the testing team. Assign tasks based on individual strengths and expertise, while also providing opportunities for skill development. 4. **Setting Expectations**: Establish clear expectations regarding timelines, quality standards, and reporting mechanisms. Ensure everyone understands their individual and collective responsibilities. 5. **Effective Communication**: Foster open and transparent communication within the team. Encourage regular updates, discussions, and feedback sessions to address any issues or challenges promptly. 6. **Risk Management**: Identify potential risks and challenges early in the planning phase. Work with your team to develop mitigation strategies and contingency plans to address any unforeseen issues during testing.	1 M for each aspect or point
	c)	**Explain test infrastructure management with its component.**	**4 M**

	Ans	**Test Infrastructure Management**  	1 M for diagram 3 M for explanation

		Test Infrastructure is made up of three essential elements: • Test Case Database (TCDB) • Defect Repository • Configuration Management Repository and Tools • **Test Case Database (TCDB)** • **Test Case** • Purpose: Record all the static information about the test. • Attributes: Test case ID, Test case name, Test case Owner • **Test case product cross reference.** • Purpose: Provides mapping between test and corresponding feature • Attributes: Test Case ID, Module ID • **Test case run history.** • Purpose: When was test run? What was the result? • Attribute: Test Case ID, Run date, Time taken, Status • Test Case –Defect Cross Reference • Purpose: Provides mapping between test case and defect • Attribute: Test Case ID , Defect ID • **Defect Repository** • It captures relevant details of defect. • It is tool of communication. • Defect Metrics are derived from defect repository. • **Configuration Management Repository and Tools** • Keeps track of change control of all the files/entities that makeup a software product. • Keeps track of version control of all the files/entities that makeup a software product.	
	d)	**Describe any four limitations of manual testing.**	**4 M**

	Ans	**Limitation of manual testing** 1. **Time-consuming:** Manual testing requires time since test cases must be executed manually. Complex software program testing could take some time. Testing teams might not have enough time to cover all test cases because of the delay in software development. 2. **Human Error:** Human errors can happen when testing is done manually. By failing to test particular scenarios or by making mistakes when executing test cases, testers may come up with erroneous results. These mistakes can make it impossible to find flaws, which would affect the caliber of the software.	1 M for each limitation

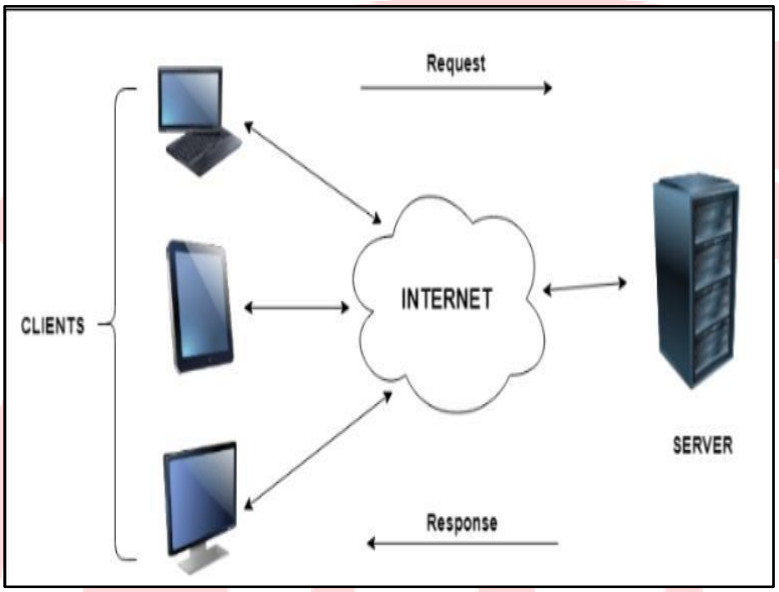
		3. **Not reusable**- The test cases created for manual testing apply to a specific version of software, but when the updates are made these test cases will become unusable and need to be rewritten. Which does not seem to be an effective use of resources and can lead increase the time. thus, slowing the development process 4. **Difficult to Measure:** It is challenging to quantify the manual testing process since there are no objective metrics provided by manual testing to assess the software' quality. It is difficult to assess the efficiency of the testing process since it is difficut to keep track of the quantity of test cases executed, errors discovered, and tet coverage attained.	
	e)	**Describe acceptance testing with its advantages.**	

	Ans	Acceptance testing is software testing that evaluates whether a system meets its business and user requirements. Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the requirements and assess whether it is acceptable for delivery or not. It is formal testing according to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers, or other authorized entities to determine whether to accept the system or not. Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use. Advantages of Acceptance Testing 1.This testing helps the project team to know the further requirements from the users directly as it involves the users for testing. 2.Automated test execution. 3.It brings confidence and satisfaction to the clients as they are directly involved in the testing process. 4.It is easier for the user to describe their requirement. 5.It covers only the Black-Box testing process and hence the entire functionality of the product will be tested.	Description of Acceptance testing - 2 M Advantages - 2 M
5.		**Attempt any TWO of the following:**	**12 M**
	a)	**Prepare and write six test cases for Library Management System of college.**	**6 M**

	Ans	**Test Case No**	**Test Case Name**	**Actual Input**	**Expected output**	**Actual Output**	**Status**	1 M for each valid test cases
		TC1	User Authentication	1.Enter Username: “22203A0011”	Login should be done successfully	Login is done successfully.	Pass	
				2.Enter Password=”sandip@1234”				

			TC2	Borrowing Book	Enter Book Name: "Data structure using C"	1.Book should be issued to student. 2.Book marked as borrowed	1.Book is issued to student. 2. Book is marked as borrowed in the database.	Pass	
			TC3	Borrowing Book	Enter Book Name: "123abcd"	It should display message "Enter Valid Book Name"	It is displaying message "Enter Valid book name"	Pass	
			TC4	Borrowing Book	Enter Book Name: "Data structure and algorithm"	It should display message "Book not available"	It is displaying message: "Book not available"	Pass	
			TC5	Returning Book	Return a borrowed book	It should mark Book as returned in student and library database.	It is marking Book as returned in student and library database.	Pass	
			TC6	Fine calculation	Return a book after the due date	It should display calculated fine in student login.	Fine calculated and displayed in student login.	Pass	
			TC7	Notification after borrowing book	1.Enter book Name: "Data structure and algorithm" 2.Collect book from librarian.	Student should Receive notification for successfully borrowed books	Student receives notification for successfully borrowed books	Pass	
	b)	**With the help of diagram, describe client-server testing.**							**6 M**

	Ans	The Client-Server application consists of two systems, one is the Client, and the other is a th Server. Here, the client and server interact with each other over the computer network.	2 M for any relevant

		In Client-Server application testing, the client sends requests to the server for specific information and the server sends the response back to the client with the requested information. Hence, this testing is also known as two-tier application testing.  FIG: Client Server Application Testing approaches of client server system is as follows. **Component Testing** ➢For testing Client and server individually approach and test plan need to be defined. ➢ One may have to devise simulators to replace corresponding components while testing the component targeted by the test. ➢ When server is tested, we may need a client simulator, while testing of client may need server simulator. **Integration Testing**	diagram, and 4 M for relevant explanation

		➢After successful testing of servers, clients, and network, they are brought together to form the system and system test cases are executed. ➢ Communication between client and server is tested in integration testing.	

Performance Testing

➢ System performance is tested when number of clients are communicating with server at a time.

➢ we can test the system under maximum load as well as normal load expected.

Concurrency Testing

➢ It may be possible that multiple users may be accessing same record at a time.

➢ Concurrency testing is required to understand the behaviour of a system under such circumstances.

Disaster Recovery Testing

➢ When the client and server are communicating with each other, there exists a possibility of breaking of the communication due to various reasons or failure of either client or server or link connecting them.

➢ It may involve testing the scenario of such failure at different points in the system and action taken by the system in each case.

Testing for extended periods

➢ In client server application it may be expected that server is running 24*7 for extended period.

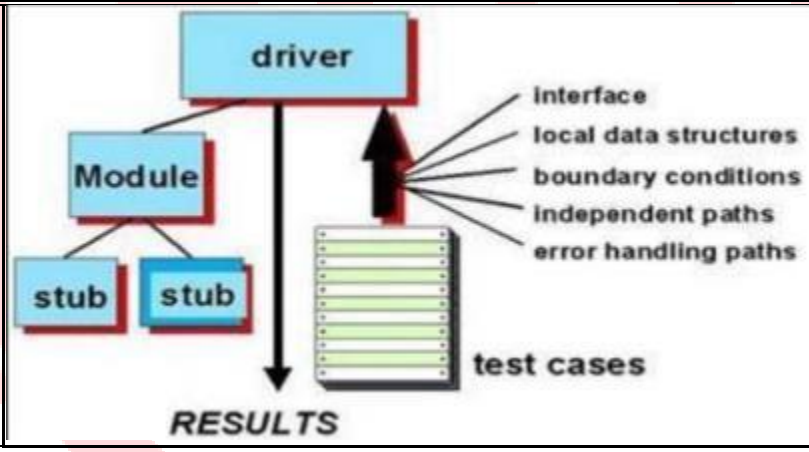
➢ one need to conduct testing over an extended period to understand if service level of network and server deteriorates over time due to some reasons.

Compatibility Testing

		➢ Servers may be in different hardware, software or operating environment than the recommended one. ➢ Client may differ significantly from the expected environmental variables.	

		➢ Testing must ensure that performance must be maintained on the range of hardware and software configurations.	
	c)	**How to select a testing tool? Explain in detail.**	**6 M**

	Ans	 **Selection criteria for testing tool** **1. Meeting requirements** • There are plenty of tools available in the market, but rarely do they meet all the requirements of a given product or a given organization. • Evaluating different tools for different requirements involve significant effort, money, and time. • The tool must match its intended use. • Wrong selection of a tool can lead to problems like lower efficiency and effectiveness of testing may be lost. Selection criteria for testing tool **2. Technology expectations:** • Test tools in general may not allow test developers to extends/modify the functionality of the framework • So, extending the functionality requires going back to the tool vendor and involves additional cost and effort. • Different phases of a life cycle have different quality-factor requirements. Tools required at each stage may differ significantly. Selection criteria for testing tool **3. Training/skills:** • While test tools require plenty of training, very few vendors provide the training to the required level. • Organization level training is needed to deploy the test tools. • As the user of the test suite are not only the test team but also the development team and other areas like configuration management.	6 M for correct explanation

		• If the testers do not have proper training and skill, then they may not be able to work effectively. Selection criteria for testing tool **4. Management aspects:** • A test tool increases the system requirement and requires the hardware and software to be upgraded. • This increases the cost of the already- expensive test tool. • Select affordable tools. Cost and benefits of various tools must be compared before making final decision.	
6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Explain the need of stubs and drivers with diagram and its importance in software testing.**	**6 M**
	Ans	**Drivers**:  **Stubs:**	2 M for diagram, 2 M for explanation of stub and Driver and 2 M for

		Importance: module where the required inputs for the module under test are f module or unit testing is known as a Driver module. The drive ret the result produced by the module under test. odule under testing may also call some other module which is n g. There is need of dummy modules required to simulate for te ules. These are called stubs. Stubs and Drivers works as a substitute for the missing or una are specifically developed, for each module, having different fu velopers and unit testers are involved in the development of stub	importanc e of stub and driver

		Their most common use may be seen in the integration incremental testing, where stubs are used in top bottom approach and drivers in a bottom-up approach.	
	b)	**Explain in detail, how to prepare a test plan with suitable example.**	**6 M**
	Ans	**Note:** Consider any relevant example • Test plan gives insight into testing activity completely. • Test Plan Ensures all Functional and Design Requirements are implemented as specified in the documentation. • The test plan addresses various levels of testing for Unit testing, Integration testing System Testing, and acceptance testing. • It explains who does testing, why test is performed, how tests are performed, how test is conducted and when test is scheduled. **Following is the test plan for 'Cam Scanner' which is installed on mobile.**	Template - 2 M, Correct explanatio n - 4 M

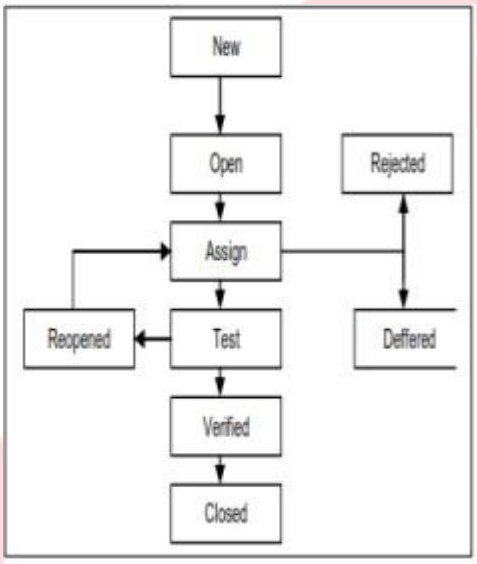
1	**Test Plan Identifier**	TP_10
2	**Introduction**	The purpose of this document is to create and test plan for camscanner. The purpose of testing this program is to check the correct operation of its functionality and ease of use.
3	**Test Items**	Working with CAMSCANNER

4	**Features to be tested**	• SCAN ID CARD DOCS BOOK QR CODE • PROCESS FILES • PDF TOOLS • TOOLS • SHARE • MOVE/COPY • RENAME • DELETE
5	**Approach**	• On the test object: o functional o nonfunctional • According to the requirements o positive o negative • By degree of preparedness - intuitive testing (ad hoc)

6	**Item Pass/Fail Criteria**	All test cases with high priority are closed with the result - pass. The test coverage is checked and sufficient, where the criterion of sufficiency is not less than 99% of the coverage of requirements by tests. The test report was compiled and approved by the team lead and customer.
7	**Suspension Criteria:** **Resumption Criteria:**	The appearance and entering the bug-tracking system of blocking bugs. Closing the blocking bug in the bug tracking system
8	**Test Deliverables**	Test plan, test case specification, test case, test summary report
9	**Test Tasks**	• Writing a test plan • Writing test cases • Development of criteria for the success of testing. • Conducting the testing and evaluation of the results • Creating test reports
10	**Environmental needs**	Cam Scanner Mobile Application Internet Mobile/Computer Android OS
11	**responsibilities**	**Functionality and Responsibilities** / **Responsible** SCAN / Test Engineer 1

				PROCESS FILES	Test Engineer 2
				PDF TOOLS	Test Engineer 3
				TOOLS	Test Engineer 4
				SHARE	Test Engineer 5
				MOVE/COPY	Test Engineer 5
				RENAME	Test Engineer 6
				DELETE	Test Engineer 6
		12	**Staffing and Training Needs**	To perform the tasks, you need to have the following knowledge and skills: • practical knowledge application of the WhatsApp is needed. • knowledge and ability to apply in practice the basic techniques of test design. • Knowledge of various types of testing including functional and non-functional.	
		13	**Schedule**	The deadline for completion of all works and delivery of the project is 01/07/2024 by 5.00pm	
		14	**Risks and Contingencies Possible risks during testing**	• Insufficient human resources for testing the application in deadlines. • Changing the requirements for the product	
		15	**Approvals Team**	Lead Test engineer 1 Test Manager Quality Manager	
	c)	**Draw a diagram for defect life cycle and write example for defect template.**			**6 M**

	Ans	**Defect life cycle**	Diagram of defect life cycle - 2 M and

			Defect Template Example - 4 M

Example of defect template for withdrawing an amount from ATM

ID	Def_01
Project	ATM Simulator
Product	Cash Simulator ATM
Release Version	v1.0
Module	Home Page > Simulator
Detected Build Version	v1.1
Summary	Limited denomination options in cash withdrawal function, restricting cash withdrawal only till 3000.
Description	No option of withdrawing of amount excess of 3000

Steps to Replicate	1) Open the website 2) Select our programs 3) Proceed to Digital Inclusion tools and select cash machine simulator (ATM) 4) Select language and skip to simulator 5) Enter the card
	6) Select the account type 7) Go to Other functions and select cash withdrawal
Actual Results	It has displaying limited options of denominations in cash withdrawal option.
Expected Results	It should add more options in denominations in withdrawal function or it should take amount input from the user
Attachments	

Remarks	Causes inconvenience to the user in terms of limited cash withdrawal options.
Defect Severity	High
Defect Priority	High
Reported By	abc
Assigned To	xyz
Status	Assigned

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

Winter – 19 EXAMINATION

Subject Name: Software Testing **Model Answer** **Subject Code: 22518**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1.		**Attempt any Five of the following:**	**10 M**
	a	**Define static and dynamic testing.**	**2M**
	Ans	**Static testing**: In static testing code is not executed. Rather it manually checks the code, requirement documents, and design documents to find errors. Main objective of this testing is to improve the quality of software products by finding errors in early stages of the development cycle. **Dynamic testing:** The dynamic testing is done by executing program. Main objective of this testing is to confirm that the software product works in conformance with the business requirements.	1 M for each definition
	b	**State any two examples of integration testing.**	**2M**

	Ans	1. Verifying the interface link between the login page and the home page i.e. when a user enters the credentials and logs it should be directed to the homepage 2. Check the interface link between the Login and Mailbox module 3. Check the interface link between the Mailbox and Delete Mails Module. 4. Verifying the interface link between the home page and the profile page i.e. profile page should open up.	Any two similar example:2M

4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

	c	**Enlist any two activities involved in test planning.**	**2M**
	Ans	**1. Scope Management:** Deciding what features to be tested and not to be tested. **2. Deciding Test approach /strategy:** Which type of testing shall be done like configuration, integration, localization etc. **3. Setting up criteria for testing:** There must be clear entry and exit criteria for different phases of testing. The test strategies for the various features and combinations determined how these features and combinations would be tested. **4. Identifying responsibilities, staffing and training needs.**	Any two activities 2M
	d	**Enlist objectives of software testing.**	**2M**
	Ans	**Objectives of software testing are as follows:** **1.** Finding defects which may get created by the programmer while developing the software. **2.** Gaining confidence in and providing information about the level of quality. **3.** To prevent defects. **4.** To make sure that the end result meets the business and user requirements. **5.** To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications. **6.** To gain the confidence of the customers by providing them a quality product.	Any two Objectives 2M
	e	**Define Defect.**	**2M**
	Ans	It refers to the several troubles with the software product, with its external behavior or its internal features. **OR** A defect is an error in coding that causes a program to fail or to produce incorrect /unexpected results.	Correct Definition 2M
	f	**State any four advantages of using tools.**	**2M**

	Ans	**Save Time /Speed:** Due to advanced computing facilities, automation test tools prevail in speed of processing the tests. Automation saves time as software can execute test cases faster than human. **Reduces the tester's involvement in executing tests:** It relieves the testers to do some other work. **Repeatability/Consistency:** The same tests can be re-run in exactly the same manner eliminating the risk of human errors such as testers forgetting their exact actions, intentionally omitting steps from the test scripts, missing out steps from the test script, all of which can	Any 4 advantages : ½ M for each
		result in either defects not being identified or the reporting of invalid bugs (which can again, be time consuming for both developers and testers to reproduce) **Simulated Testing:** Automated tools can create many concurrent virtual users/data and effectively test the project in the test environment before releasing the product. **Test case design:** Automated tools can be used to design test cases also through automation, better coverage can be guaranteed than if done manually. **Reusable:** The automated tests can be reused on different versions of the software, even if the interface changes. **Avoids human mistakes:** Manually executing the test cases may incorporate errors. But this can be avoided in automation testing**.** **Internal Testing:** Testing may require testing for memory leakage or checking the coverage of testing. Automation can done this easily. **Cost Reduction:** If testing time increases cost of the software also increases. Due to testing tools time and therefore cost is reduced**.**	
	g	**Define Bug, Error, Fault, and Failure.**	**2M**

	Ans	**Bug:** A bug can be defined as the initiation of error or a problem due to which fault, failure, incident or an anomaly occurs. **Error:** A human action that produces an incorrect result. **Fault:** An incorrect step, process, or data definition in a computer program. **Failure:** A failure is said to occur whenever the external behavior of a system does not conform to that prescribed in the system specification. A software fault becomes a software failure only when it is activated.	½ M for each definition
2.		**Attempt any Three of the following:**	**12M**
	a	**Define Boundary value analysis with suitable example.**	**4M**
	Ans	Most of the defects in software products hover around conditions and boundaries. By conditions, we mean situations wherein, based on the values of various variables, certain actions would have to be taken. By boundaries, we mean —limits of values of the various variables. • This is one of the software testing technique in which the test cases are designed to include values at the boundary. • If the input data is used within the boundary value limits, then it is said to be Positive Testing. If the input data is	Explanation:2M and 2 M for Example

		picked outside the boundary value limits, then it is said to be Negative Testing. • Boundary value analysis is another black box test design technique and it is used to find the errors at boundaries of input domain rather than finding those errors in the center of input. • Each boundary has a valid boundary value and an invalid boundary value. Test cases are designed based on the both valid and invalid boundary values. Typically, we choose one test case from each boundary. • Boundary value analysis is a black box testing and is also applies to white box testing. Internal data structures like arrays, stacks and queues need to be checked for boundary or limit conditions. When there are linked lists used as internal structures, the behavior of the list at the beginning and end has to be tested thoroughly. • Boundary value analysis help identify the test cases that are most likely to uncover defects. **Example 1:** A system can accept the numbers from 1 to 10 numeric values. All other numbers are invalid values. Under this technique, boundary values 0, 1,2,9,10,11 can be tested. **Example 2**: The exam has a pass boundary at 40 percent, merit at 75 percent and Distinction at 85 percent. The Valid Boundary values for this scenario will be as follows: • 49, 50 - for pass • 74, 75 - for merit • 84, 85 - for distinction Boundary values are validated against both the valid boundaries and invalid boundaries. The Invalid Boundary Cases for the above example can be given as follows: • 0 - for lower limit boundary value • 101 - for upper limit boundary value	
	b	**Differentiate between drivers and stub (any four points).**	**4M**

	Ans		1 M for each valid point

			Stubs	**Drivers**	
			Stubs are dummy modules that always used to simulate the low level modules.	Drivers are dummy modules that always used to simulate the high level modules.	
			Stubs are the called programs.	Drivers are the calling programs.	
			Stubs are used when sub programs are under construction.	Drivers are only used when main programs are under construction.	
			Stubs are used in top down approach.	Drivers are used in bottom up integration.	
	c	**State the contents of 'Test Summary Reports' used in test reporting.**			**4M**

	Ans	Test reporting is a means of achieving communication throughout the testing cycle. There are 3 types of test reporting. **1. Test incident report:** **2. Test cycle report:** **3. Test summary report:** **Test summary Report**: The final step in a test cycle is to recommend the suitability of a product for release. A report that summarizes the result of a test cycle is the test summary report. There are two types of test summary report: 1. Phase wise test summary, which is produced at the end every of phase. 2. Final test summary report, which has all the details of done testing by all phases. A Summary report should present 1. Test Summary Report Identifier 2 Description: Identify the test items being reported in this report with test id 3 Variances: Mention any deviation from test plans, test procedures, proced if any. 4 Summary of results: All the results are mentioned here with the resolved incidents and their solutions. 5 Comprehensive assessment and recommendation for release should include: Fit for release assessment and recommendation of release.	Explanation4 M

	d	**State any eight limitations of manual testing.**	**4M**

	Ans	1. Manual testing is slow and costly. 2. It is very labor intensive; it takes a long time to complete tests. 3. Manual tests don't scale well. As the complexity of the software increases the complexity of the testing problem grows exponentially. This leads to an increase in total time devoted to testing as well as total cost of testing. 4. Manual testing is not consistent or repeatable. Variations in how the tests are performed as inevitable, for various reasons. One tester may approach and perform a certain test differently from another, resulting in different results on the same test, because the tests are not being performed identically. 5. Lack of training is the common problem. 6. GUI objects size difference and color combinations are not easy to find in manual testing. 7. Not suitable for large scale projects and time bound projects. 8. Batch testing is not possible, for each and every test execution Human user interaction is mandatory. 9. Comparing large amount of data is impractical. 10. Processing change requests during software maintenance takes more time.	Any 8 points 1/2 M for each point
3.		**Attempt any Three of the following:**	**12M**
	a	**Describe the use of decision table in black box testing with the help of suitable example.**	**4M**

	Ans	I.Decision table testing is black box test design technique to determine the test scenarios for complex business logic. ii. Decision tables provide a systematic way of stating complex business rules, which is useful for developers as well as for testers. iii. Decision tables can be used in test design whether or not they are used in specifications, as they help testers explore the effects of combinations of different inputs and other software states that must correctly implement business rules. iv. It helps the developers to do a better job can also lead to better relationships with them. v. Testing combinations can be a challenge, as the number of combinations can often be huge. vi. Testing all combinations may be impractical if not impossible. vii. We have to be satisfied with testing just a small subset of combinations but making the choice of which combinations to test and which to leave out is also important.	Use of decision table in black box testing with example 4M

Viii.If you do not have a systematic way of selecting combinations, an arbitrary subset will be used and this may well result in an ineffective test effort.

Importance of Decision Table: Essentially it is a structured exercise to formulate requirements when dealing with complex business rules. Decision tables are used to model complicated logic. They can make it easy to see that all possible combinations of conditions have been considered and when conditions are missed, it is easy to see.

Example :

Conditions	**TC1**	**TC2**	**TC3**	**TC4**
Request login	**0**	**1**	**1**	**1**
Valid username entered	**X**	**0**	**1**	**1**
Valid password entered	**X**	**X**	**0**	**1**
Actions				
Offer recover credentials	**0**	**1**	**1**	**0**
Activate entry box username	**0**	**1**	**1**	**0**
Activate entry box Password	**0**	**0**	**1**	**0**
Enter privilege area	**0**	**0**	**0**	**1**

Where 0→ False
1→ True
X→ No action (Don't care)

	b	**Describe standards included in Test management.**	**4M**

	Ans	**Internal standards are:** 1. Naming and storage conventions for test artifacts. 2. Document standards 3. Test coding standards 4. Test reporting standards. **1. Naming and storage conventions for test artifacts:** Every test artifact (test specification, test case, test results and so on) have to be named appropriately and meaningfully. It enables a) Easy identification of the product functionality. b) Reverse mapping to identify the functionality corresponding to a given set of tests. E.g. modules shall be M01, M02. Files types can be .sh, .SQL.	Standards included in Test management4M

		2. Documentation standards: a) Appropriate header level comments at the beginning of a file that outlines the functions to be served by the test. b) Sufficient inline comments, spread throughout the file c) Up-to-Date change history information, reading all the changes made to the test file. **3. Test coding standards**: a) Enforce right type of initialization b) Stipulate ways of naming variables. c) Encourage reusability of test artifacts d) Provide standard interfaces to external entities like operating system, hardware and so on. 4. **Test reporting standard**: All the stakeholders must get a consistent and timely view of the progress of tests. It provides guidelines on the level of details that should be present in the test report, their standard formats and contents. **5.External Standards:** These are the standards made by an entity external to an organization. These standards are standards that a product should comply with, are externally visible and are usually stipulated by external parties. The three types of external standards are: • Customer standard: refer to something defined by the customer as per his/her business requirement for the given product. • National Standard: refer to something defined by the regulatory entities of the country where the supplier / customer resides. • International Standard: are defined at international level and these are applicable to all customers across the globe.	
	c	**Enlist different techniques for finding defects and describe any one technique with an example.**	**4M**

	Ans	Different techniques for finding defects are as given below: **a) Quick Attacks:** **i. Strengths** • The quick-attacks technique allows you to perform a cursory analysis of a system in a very compressed timeframe. • Even without a specification, you know a little bit about the software, so the time spent is also time invested in developing expertise.	List of any relevant techniques 1M, explanation of 1 technique with example 3M

- The skill is relatively easy to learn, and once you've attained some mastery your quick-attack session will probably produce a few bugs.
- Finally, quick attacks are quick.
- They can help you to make a rapid assessment. You may not know the requirements, but if your attacks yielded a lot of bugs, the programmers probably aren't thinking about exceptional conditions, and it's also likely that they made mistakes in the main functionality.
- If your attacks don't yield any defects, you may have some confidence in the general, happy-path functionality. **ii. Weaknesses**
- Quick attacks are often criticized for finding "bugs that don't matter"— especially for internal applications.
- While easy mastery of this skill is strength, it creates the risk that quick attacks are "all there is" to testing; thus, anyone who takes a two day course can do the work.

b) Equivalence and Boundary Conditions

i. Strengths

- Boundaries and equivalence classes give us a technique to reduce an infinite test set into something manageable.
- They also provide a mechanism for us to show that the requirements are "covered". **ii. Weaknesses**
- The "classes" in the table in Figure 1 are correct only in the mind of the person who chose them.
- We have no idea whether other, "hidden" classes exist—for example, if a numeric number that represents time is compared to another time as a set of characters, or a "string," it will work just fine for most numbers.

c) Common Failure Modes

i. Strengths

- The heart of this method is to figure out what failures are common for the platform, the project, or the team; then try that test again on this build.
- If your team is new, or you haven't previously tracked bugs, you can still write down defects that "feel" recurring as they occur—and start checking for them. **ii. Weaknesses**
- In addition to losing its potency over time, this technique also entirely fails to find "black swans"—defects that exist outside the team's recent experience.

		• The more your team stretches itself (using a new database, new programming language, new team members, etc.), the	

riskier the project will be—and, at the same time, the less valuable this technique will be.

d) State-Transition Diagrams

In this technique the state transition diagram is prepared with respect to the applied inputs and produced output. It clearly shows how the state transition of software takes place from one to another and hence can be useful to find the defects.

One of the example is as shown in the diagram below:

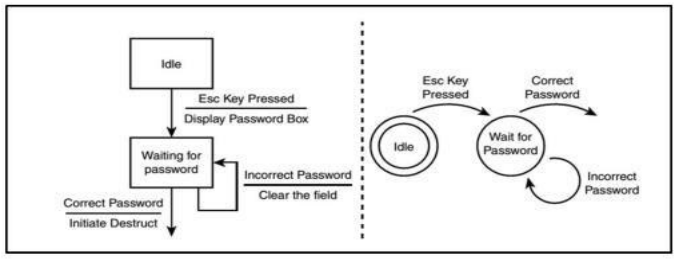


Figure 4: State Transition Map

i. Strengths

- Mapping out the application provides a list of immediate, powerful test ideas.
- Model can be improved by collaborating with the whole team to find "hidden" states—transitions that might be known only by the original programmer or specification author.
- Once you have the map, you can have other people draw their own diagrams, and then compare theirs to yours.
- The differences in those maps can indicate gaps in the requirements, defects in the software, or at least different expectations among team members. **ii. Weaknesses**
- The map you draw doesn't actually reflect how the software will operate; in other words, "the map is not the territory."
- Drawing a diagram won't find these differences, and it might even give the team the illusion of certainty.
- Like just about every other technique on this list, a statetransition diagram can be helpful, but it's not sufficient by itself to test an entire application.

e) Use Cases and Soap Opera Tests

Use cases and scenarios focus on software in its role to enable a human being to do something. **i. Strengths**

- Use cases and scenarios tend to resonate with business customers, and if done as part of the requirement process,

		they sort of magically generate test cases from the requirements.	

- They make sense and can provide a straightforward set of confirmatory tests. Soap opera tests offer more power, and they can combine many test types into one execution. **ii. Weaknesses**
- Soap opera tests have the opposite problem; they're so complex that if something goes wrong, it may take a fair bit of troubleshooting to find exactly where the error came from!

f) Code-Based Coverage Models

Imagine that you have a black-box recorder that writes down every single line of code as it executes. **i. Strengths**

- Programmers love code coverage. It allows them to attach a number— an actual, hard, real number, such as 75%—to the performance of their unit tests, and they can challenge themselves to improve the score.
- Meanwhile, looking at the code that isn't covered also can yield opportunities for improvement and bugs!

ii. Weaknesses

- Customer-level coverage tools are expensive, programmerlevel tools that tend to assume the team is doing automated unit testing and has a continuous-integration server and a fair bit of discipline.
- After installing the tool, most people tend to focus on statement coverage—the least powerful of the measures.
- Even decision coverage doesn't deal with situations where the decision contains defects, or when there are other, hidden equivalence classes; say, in the third-party library that isn't measured in the same way as your compiled source code is.
- Having code-coverage numbers can be helpful, but using them as a form of process control can actually encourage wrong behaviors. In my experience, it's often best to leave these measures to the programmers, to measure optionally for personal improvement (and to find dead spots), not as a proxy for actual quality.

g) Regression and High-Volume Test Techniques

People spend a lot of money on regression testing, taking the old test ideas described above and rerunning them over and over. This is generally done with either expensive users or very expensive programmers spending a lot of time writing and later maintaining those automated tests. **i. Strengths**

- For the right kind of problem, say an IT shop processing files through a database, this kind of technique can be extremely powerful.
- Likewise, if the software deliverable is a report written in SQL, you can hand the problem to other people in plain English, have them write their own SQL statements, and compare the results.
- Unlike state-transition diagrams, this method shines at finding the hidden state in devices. For a pacemaker or a missile-launch device, finding those issues can be pretty important. **ii. Weaknesses**
- Building a record/playback/capture rig for a GUI can be extremely expensive, and it might be difficult to tell whether the application hasn't broken, but has changed in a minor way.
- For the most part, these techniques seem to have found a function in IT/database work, at large companies like Microsoft and AT&T, which can have programming testers doing this work in addition to traditional testing, or finding large errors such as crashes without having to understand the details of the business logic.
- While some software projects seem ready-made for this approach, others aren't.
- You could waste a fair bit of money and time trying to figure out where your project falls.

OR

Different techniques for finding defects are:

1. Static technique
2. Dynamic technique
3. Operational technique

1. Static Techniques: Static techniques of quality control define checking the software product and related artifacts without executing them. It is also termed desk checking/verification /white box testing. It may include reviews, walkthroughs, inspection, and audits here; the work product is reviewed by the reviewer with the help of a checklist, standards, any other artifact, knowledge and experience, in order to locate the defect with respect to the established criteria. Static technique is so named because it involves no execution of code,

		product, documentation, etc. This technique helps in establishing conformance to requirements view. **2. Dynamic Testing:** Dynamic testing is a validation technique which includes dummy or actual execution of work products to	

		evaluate it with expected behavior. It includes black box testing methodology such as system testing and unit testing. The testing methods evaluate the product with respect to requirements defined; designs created and mark it as pass or fail. **3.Operational techniques:** Operational techniques typically include auditing work products and projects to understand whether the processes defined for development /testing are being followed correctly or not, and also whether they are effective or not. It also includes revisiting the defects before and after fixing and analysis. Operational technique may include smoke testing and sanity testing of a work product.	
	d	**Enlist factors considered for selecting a testing tool for test automation.**	**4M**

	Ans	**The following factors are important during tool selection:** i. **Assessment of the organization's maturity** (e.g. readiness for change); ii. Identification of the areas within the organization where **tool support will help to improve testing processes;** iii. **Evaluation of tools against clear requirements and objective criteria;** iv. **Proof-of-concept to see whether the product works as desired** and meets the requirements and objectives defined for it; v. **Evaluation of the vendor (training, support and other commercial aspects**) or open-source network of support; vi. **Identifying and planning internal implementation** (including coaching and mentoring for those new to the use of the tool). **OR** **The industry experts have suggested following four major criteria for selection of testing tools.** 1) Meeting requirements. 2) Technology expectations. 3) Training / skills. 4) Management aspects. **1) Meeting Requirements:** a) There are many tools available in the market today but rarely do they meet all the requirements of given product or a given organization. Evaluating different tools for different requirements involves lot of effort, money and time. Huge delay is involved in selecting and implanting test tools. b) Test tools may not provide backward or forward compatibility with the product-under-test (PUT). c) Test tools may not go through the same amount of evaluation for new requirements. For example: some tools had Y2K-problem.	Any relevant factors minimum 4M

d) A number of test tools cannot distinguish between a product failure and a test failure. This increases analysis time and manual testing. The test tools may not provide the required amount of trouble-shooting/debug/error messages to help in analysis.
For example, in case of GUI testing, the test tools may determine the results based on messages and screen coordinates at run-time. Hence, if the screen elements of the product are changed, it requires the test suite to be changed. The test tool must have some intelligence to proactively find out the changes that happened in the product and accordingly analyze the results.

2) Technology Expectations:

a) In general, test tools may not allow test developers to extend / modify the functionality of the framework. So, it involves going back to the tool vendor with additional cost and effort. Very few tools available in market provide source code for extending functionality or fixing some problems. Extensibility and customization are important expectations of a test tool.

b) A good number of test tools require their libraries to be linked with product binaries. When these libraries are linked with the source code of the product, it is called as the "instrumented code". This causes portion of testing be repeated after those libraries are removed, as the results of certain types of testing will be different and better when those libraries are removed. For example, the instrumented code has a major impact on the performance testing since the test tools introduce an additional code and there could be a delay in executing the additional code.

c) Finally, test tools are not 100% cross-platform. They are supported only on some O.S. platforms and the scripts generated from these tools may not be compatible on other platforms. Moreover, many of the test tools are capable of testing only the product, not the impact of the product/test tool to the system or network. When there is an impact analysis of the product on the network or system, the first suspect is the test tool and it is uninstalled when such analysis starts.

3) Training Skills:

Test tools require plenty of training, but very few vendors provide the training to the required level. Organization-level training is needed to deploy the test tools, as the users of the test suite are not only the test team but also the development team and other areas like SCM (Software Configuration Management). Test tools expect the users to learn new language/scripts and may not use standard

		languages/scripts. This increases skill requirements for automation and increases the need for a learning curve inside the organization. **4) Management Aspects:**	

A test tool increases the system requirement and requires the hardware and software to be upgraded. This increases the cost of the already-expensive test tool. When selecting the test tool, it is important to note the system requirements and the cost involved in upgrading the software and hardware needs to be included with the cost of the tool. Migrating from one test tool to another may be difficult and requires a lot of effort. Not only is this difficult, as the test suite that is written cannot be used with other test tools but also because of the cost involved. As the tools are expensive and unless the management feels that the returns on investment (ROI are justified, changing tools are generally not permitted.
Deploying a test tool requires as much effort as deploying a product in a company. However, due to project pressures, test tools effort at deploying gets diluted, not spent. Thus, later it becomes one of the reasons for delay or for automation not meeting expectations The support available on the tool is another important point to be considered while selecting and deploying the test tool.

4. **Attempt any THREE of the following.** **12M**

a **Differentiate between alpha and beta testing. (four points)** **4M**

Ans

Alpha Testing	Beta Testing
Alpha testing performed by Testers who are usually internal employees of the organization.	Beta testing is performed by Clients or End Users who are not employees of the organization.
Alpha Testing performed at developer's site.	Beta testing is performed at a client location or end user of the product.
Reliability and Security Testing are not performed in-depth Alpha Testing.	Reliability, Security, Robustness is checked during Beta Testing.
Alpha testing involves both the white box and black box techniques.	Beta Testing typically uses Black Box Testing.

4 differences 4M, 1M each. Any other relevant differences shall be given Marks.

			Alpha testing requires a lab environment or testing environment.	Beta testing doesn't require any lab environment or testing environment. The software is made available to the public and is said to be real time environment.	
			Long execution cycle may be required for Alpha testing.	Only a few weeks of execution are required for Beta testing	
			Critical issues or fixes can be addressed by developers immediately in Alpha testing.	Most of the issues or feedback is collected from Beta testing will be implemented in future versions of the product.	
			Alpha testing is to ensure the quality of the product before moving to Beta testing	Beta testing also concentrates on the quality of the product, but gathers users input on the product and ensures that the product is ready for real time users.	
	b	**Describe test infrastructure management.**			**4M**

	Ans	**Test infrastructure management** This Testing requires a robust infrastructure to be planned upfront. infrastructure is made up of three essential elements. es all **1. A test case database (TCDB):** A test case database tion. captur the relevant information about the test cases in an organiz ble Some of the entities and the attributes are given in following t A test case database captures all the relevant information the bout such test cases in an organization. Some of the entities and attributes in each of the entities in a TCDB are: • Test case • Test case-product cross reference • Test case run history • Test case- defect cross reference **2. Defect repository**	Test infrastructure management description :4M

Sr. No.	Test Case	Purpose	Attributes
1	Test case	Records all static information about tests.	1)Test case Id 2) Test case name (File name) 3) Test case owner 4) Associated files for test case.
2	Test case product cross reference	Provide mapping between the tests and the corresponding product features, enables identification of test cases for given feature.	Test case Id Module Id
3	Test case run history	Gives the history of when the test case was run and what was result , provided inputs on selection of test for regression runs	1) Test case Id 2) Run date 3) Time taken 4) Run status(Success/ Failure)
4	Test casedefect crossreference	Gives details of test cases introduced to test certain specific defects detected in the product, provides inputs on the selection of test for regression runs.	1) Test case Id 2) Defect reference

It captures all the relevant information of defect repository for a product. The information that a defect repository includes

- Defect details
- Defect test detail
- Fix details
- Communication

2. Defect repository

It captures all the relevant information of defect repository for a product. The
information that a defect repository includes

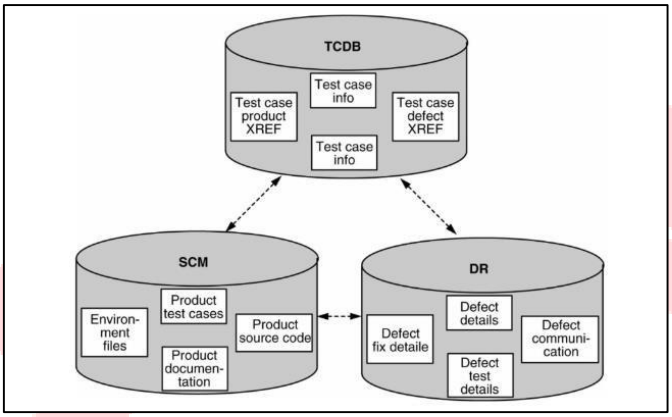
- Defect details
- Defect test detail
- Fix details
- Communication

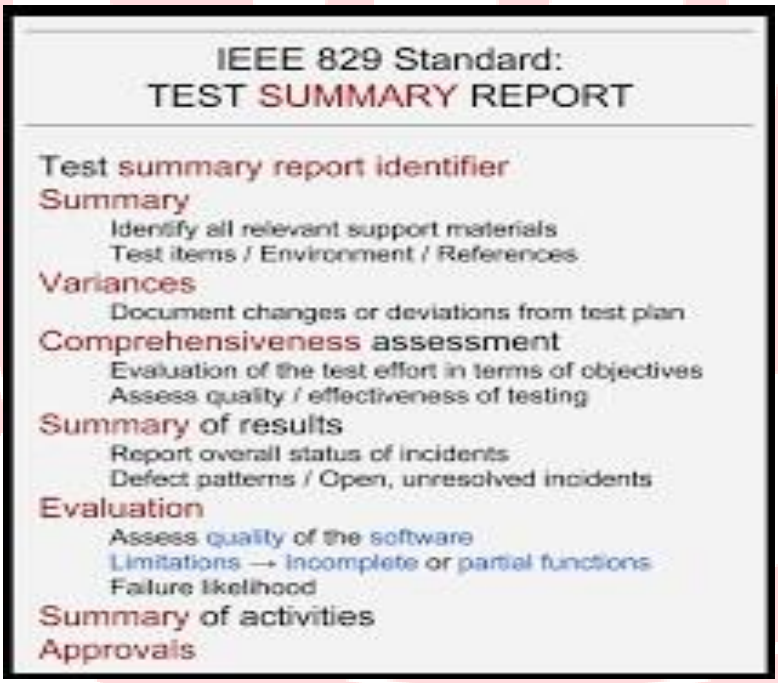
3. Configuration Management (CM) repository and **tool**

Software Configuration Management is defined as a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle.

It keeps track of change control and version control of all the files/entities that make up a software product. Change control ensures that

- Changes to test files are made in a controlled fashion and only with proper approvals
- Change are made by one test engineer are not accidently lost or overwritten by other changes
- Each change produces distinct version of the file that is re-creatable at any point of time
- Everyone gets access to only the most recent version of the test files.


TCDB
Test case info
Test case product XREF
Test case defect XREF
Test case info
SCM
Product test cases
Environ-ment files
Product source code
Product documen-tation
DR
Defect details
Defect fix detaile
Defect communi-cation
Defect test details

	c	**Describe the process of preparing summary report in test planning.**	**4M**
	Ans	**Preparing test summary report** At the completion of a test cycle, a test summary report is produced. This report gives insights to the senior management about the fitness of the product for release. There are two types of reports that are required: 1. The Incident Report 2. Test Cycle Report 3. Test Summary Report A summary report should present the following things: 1. A summary of the activities carried out during the test cycle; 2. Variance of the activities carried out from the activities planned; 3. Summary of results should include tests that failed and severity of impact of defect; 4.Comprehensive assessment and recommendation for release should include "Fit for release" assessment and Recommendation of release  	Process of preparing summary report in test planning 4M , any other relevant answer shall be given Marks.
	d	**Describe object oriented metrics in testing.**	**4M**

	Ans	**Object oriented metrics in testing:** OBJECT-ORIENTED METRICS AND MEASURES As object-oriented approach emerged to support major applications, the effectiveness of applying traditional software metrics to	Any 4 object oriented metrics in testing 4M; Relevant answer shall be given Marks.

		object-oriented systems was challenged. The object-oriented design approach gives opportunity to classify metrics naturally. The classification captures object-oriented software features and properties hierarchically. It begins with the high-level characteristics of an object-oriented system and moves down to the low-level characteristics. **Source code size metrics**: Traditional metrics which are applied to object oriented software give insight into an overall system size and allow comparing systems and evaluating productivity. They can also be used as a refactoring effectiveness indicator. **Lines of Code (LOC) metric** is most common software project measure. The metric becomes a baseline to measure the degree of work performed on a project and it is used to create time and cost estimates. **Effective Lines of Code Metric (eLOC)** is a measure of all lines that are not comments, blanks or standalone braces or parenthesis. This metric more closely represents the quantity of work performed. **Comment Line and Comment Percent** (or Comment to Code Ratio) is a degree of commenting within the source code. It measures the care taken by programmers to make the source code and algorithms understandable. Poorly commented code makes the maintenance activities an extremely expensive. Recommended minimum is 20%. **Blank Line and White Space Percent Metric** is the number of blank lines within source code. It indicates the readability of product. And File Count Metric counts the files processed and generates metrics based on the file extension. It provides the distribution of the source code types, source code types and distribution of the specifications to the implementations. **Procedural metrics**: Cyclomatic Complexity is a popular procedural (called also function) software metric equal to the number of decisions that can be taken in a procedure A decision is defined as an occurrence of keywords such as: "while", "for", "for each", "continue", "if", "case", "go to", "try" and "catch" within the function. Cyclomatic Complexity is the sum of these constructs. That metric helps to identify software need of inspection or redesign, and also to allocate resources for evaluation and test. **Class metrics**: Class metrics describe structure of a class and relationship between classes. The volume of a class is a basic size measure connected with the amount of information inside it. The	

		class volume can be measured by Number of Variables and by Number of Methods. Also Average LOC per Class and per Method metrics can provide insight into the average module size in the system.	

		Method metrics are used to estimate effort for testing early. Those metrics can be measured by Number of Parameters per Method, Weighted Methods per Class, Maximum Nesting Level, and Method Rank. Number of Parameter per Method counts parameters of a method and also references. **Afferent Coupling and Efferent Coupling** at method level are another object coupling metrics. Afferent Coupling for a particular method is the number of methods that depends directly on it and the Efferent Coupling for a particular method is the number of methods it directly depends on. Afferent Coupling is an indicator for the responsibility. The higher this value is the higher is the element's responsibility. Efferent Coupling means that a element depends on several other implementation details and it makes it instable. Therefore it is good practice to keep the Efferent Coupling for all artefacts at a minimum. **Inheritance metrics** :The inheritance relationships characteristic between classes and their parents indicate to a designer where changes would improve the development. The metrics connected to classes inheritance should take into account both the depth and breadth of the relationships. The Height of Inheritance Tree metric is counted as the maximum number of nodes from the class node to the root of the inheritance hierarchy. The deeper within the hierarchy, the more methods the class can inherit, increasing its complexity.	
	e	**State the testing approaches that are considered during client server testing.**	**4M**

	Ans	Testing approaches of client server system: • **Component Testing**: One need to define the approach and test plan for testing client and server individually. When server is tested there is need of a client simulator, whereas testing client a server simulator, and to test network both simulators are used at a time. • **Integration testing**: After successful testing of server, client and network, they are brought together to form system testing. • **Performance testing:** System performance is tested when number of clients is communicating with server at a time. Volume testing and stress testing may be used for testing, to test under maximum load as well as normal load expected. Various interactions may be used for stress testing. • **Concurrency Testing:** It is very important testing for clientserver architecture. It may be possible that multiple users may be accessing same record at a time, and concurrency	Testing approaches of client server testing 4 approaches 4 marks;1 M each

testing is required to understand the behavior of a system in this situation.

- **Disaster Recovery /Business continuity testing:** When the client server are communicating with each other , there exist a possibility of breaking of the communication due to various reasons or failure of either client or server or in connecting them. The requirement specifications must describe the possible expectations in case of any failure.

- **Testing for extended periods**: In case of client server applications generally server is never shutdown unless there is some agreed Service Level Agreement (SLA) where server may be shut down for maintenance. It may be expected that server is running 24X7 for extended period One needs to conduct testing over an extended period to understand if service level of network and server deteriorate over time due to some reasons like memory leakage. **Compatibility Testing**: Client server may be put in different environments when the users are using them in production. Server may be in different hardware, software, or operating system environment than the recommended. Other testing such as security testing and compliance testing may be involved if needed, as per testing and type of system.

5.		**Attempt any Three of the following:**	**12M**
	a	**Design test cases for railway reservation system.**	**4M**
	Ans	**Test cases for railway reservation system:**	Any 6 valid test cases :6 M, 1 M each Any other relevant test Cases shall be considered

Test case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Login field	Any valid login name (abcxyz)	It should accept the login name	It accepted the login name	Pass

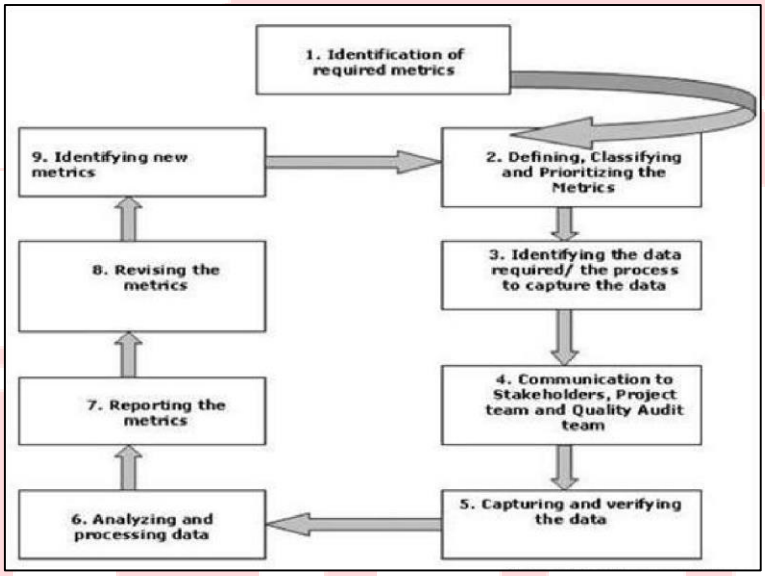
			TC2	Password field	Valid password	It should accept the valid password	It accepted the valid password; successful login message	Pass	
			TC3	Password field	Invalid password	It should not accept the valid password	Message displayed as invalid login or wrong password.	Pass	
			TC4	Date of journey	Date format not before the current date	It should accept date	Accepted the date	Pass	
			TC5	Date of return journey	Date format, date greater than the date of journey	It should accept the date	Accepted the date	Pass	

			TC6	Boarding station	Valid boarding station	It should accept	Accepted the boarding station	Pass	
			TC7	Train number	Valid train number	It should accept the valid train number	Train number accepted	Pass	
	b	**With respect to GUI testing write the test cases for Amazon login form.**							**4M**
	Ans		**Test case ID**	**Test case objective**	**Input data**	**Expected result**	**Actual result**	**Status**	Any 6 valid test cases :6M, 1M each Any other relevant test Cases shall be considered
			TC1	Check cursor position at email or mobile number field	Click on email or mobile number field	Cursor should be placed on the field	Placed the cursor on the field	Pass	
			TC2	Check cursor position at password field	Click on password field	Cursor should be placed on the password field	Placed the cursor on the password field	Pass	

			TC3	Check the continue button	Click on continue button	It should redirect to password page	It redirected to the password page.	Pass	
			TC4	Readability of font	Try to read the contents on login page	Contents should be readable	Contents are readable	Pass	
			TC5	Testing of spelling of login	Check the spelling of login	Login spelling should be correct	Spelling of Login is correct	Pass	

			TC6	Testing of hyperlink	Hover the mouse on hyperlink	It should change the cursor and should redirect to respective page on click	Cursor changed and redirects to other page.	Pass	
	c	**Elaborate the term metrics and measurement and write the need of software measurement.**							**4M**

	Ans	**Metrics and measurement :** A Metric is a measurement of the degree that any attribute belongs to a system, product or process. For example the number of errors per person hours would be a metric. Thus, software measurement gives rise to software metrics. A measurement is an indication of the size, quantity, amount or dimension of a particular attribute of a product or process. For example the number of errors in a system is a measurement. A Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. Metrics can be defined as "STANDARDS OF MEASUREMENT". Software Metrics are used to measure the quality of the project. Simply, Metric is a unit used for describing an attribute. Metric is a scale for measurement. **Need of Software measurement:** 1. Establish the quality of the current product or process. 2. To predict future qualities of the product or process. 3. To improve the quality of a product or process. 4. To determine the state of the project in relation to budget and schedule.	



6.		**Attempt any Three of the following:**	**12M**
	a	**Design test cases for hostel admission form of your institute.**	**4M**
	Ans		6 test cases of test cases for hostel admission form of institute : 6 M; 1M each; any other valid test cases shall be considered

Test case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Student name field	Any valid alphabetical characters (John)	It should accept the name	Students name is accepted	Pass

			TC2	Date of birth field	Date format before the current date	It should accept the date less than the current date	It accepted the valid date	Pass	
			TC3	Gender field	Radio button should be selected. F or M	It should select the proper radio button	Proper radio button is selected	Pass	
			TC4	Date of admission	Date format not before the current date	It should accept date	Accepted the date	Pass	
			TC5	Age field	Any numerical data greater than or equal to 16	It should accept the number greater than or equal to 16	Accepted the age	Pass	
			TC6	Address field	Valid alpha numeric characters	It should accept the address	Accepted the address	Pass	

			TC7	Pin code	Valid 6 digits numeric format	It should accept the valid pin code	Pin code accepted	Pass	
	b	**Design a test plan along with the test cases for edit function in notepad.**							**4M**
	Ans								Any 3 valid test cases 3 M ; 1M each for edit function in notepad test plan 3 M

Test case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Test the select all option	Click on select all	All the text should be selected	All the text is selected	Pass
TC2	Cut option	Select the text and click on cut	Selected text should be cut	Selected text is cut	Pass
TC3	Paste option	Click on paste	Contents should be pasted	Contents are pasted	Pass
TC4	Delete option	Select text and click on delete	Contents should be deleted	Contents are deleted	Pass

Test plan :

Test Plan Identifier

TP_10

Introduction: The purpose of this document is to create an application test plan for EDIT option of Notepad. The purpose of testing this program is to check the correct operation of its functionality, ease of use.

Test Items: Working with the document (select, cut, copy etc.)

		Features to be tested • Select all text • Cut some text • Paste the text	

- Delete the text
- Copy the text
- Finding and replacing text

Features to be tested

- Working with Help
- Time and date option

Approach

- On the test object:
  - functional
  - non-functional
- According to the requirements
  - positive
  - negative
- By degree of preparedness - intuitive testing (ad hoc) **Item Pass/Fail Criteria:** All test cases with high priority are closed with the result - pass. The test coverage is checked and sufficient, where the criterion of sufficiency is not less than 99% of the coverage of requirements by tests. The test report was compiled and approved by the team lead and customer.

Suspension Criteria and Resumption Requirements

Criterion for interrupting testing:

- The appearance and entering into the bug-tracking system of blocking bugs. Criterion for continuation of testing:
- Closing the blocking bug in the bug tracking system. **Test Deliverables:** Test plan, test cases, test report.

Test Tasks

- Writing a test plan
- Writing test cases
- Development of criteria for the success of testing
- Conducting the testing and evaluation of the results
- Creating test reports

Environmental Needs

Notepad
Computer
Windows os

Responsibilities

Sr. no	Functionality and Responsibilities	Responsible
1	select all text	Test engineer 1
2	cut the text	Test engineer 1
3	paste the text	Test engineer 1
3	copy the text	Test engineer 1
5	find the text	Test engineer 2
6	replacing text	Test engineer 2
7	delete the selected text	Test engineer 2

Staffing and Training Needs

To perform the tasks, you need to have the following knowledge and skills:

- knowledge and practical application of the notepad;
- knowledge and ability to apply in practice the basic techniques of test design
- Knowledge of various types of testing including functional and non-functional.

Schedule

The deadline for completion of all works and delivery of the project is 06/12/2019 by 5.00pm

Risks and Contingencies

Possible risks during testing:

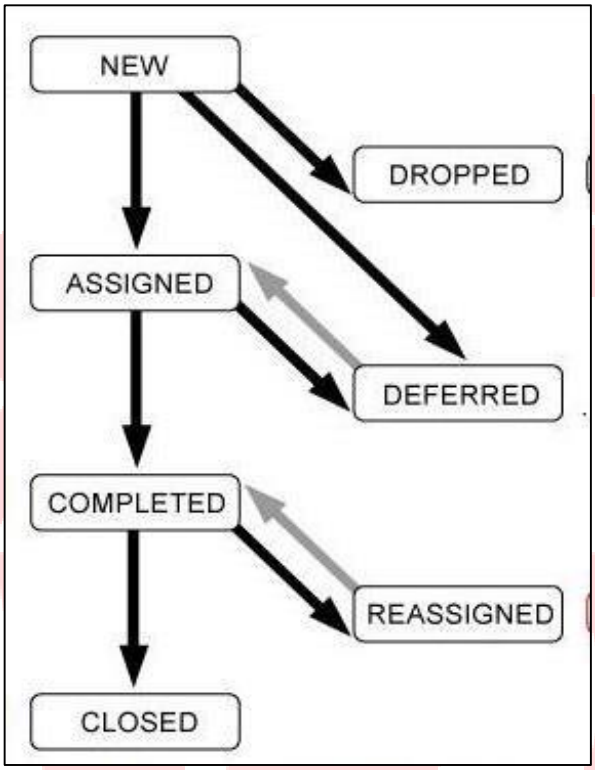
- Insufficient human resources for testing the application in deadlines.
- Changing the requirements for the product

Approvals

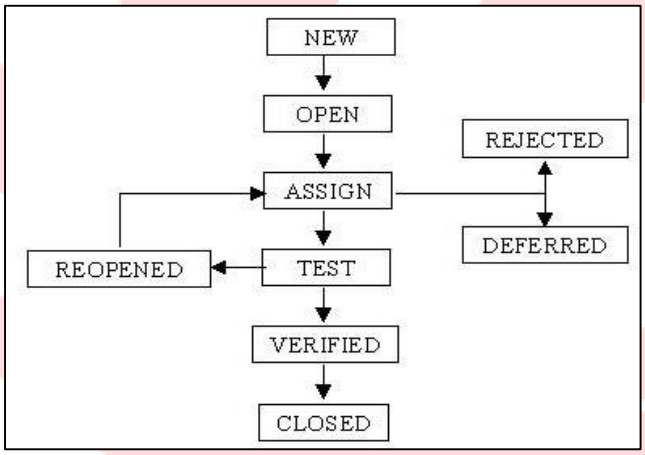
Team Lead
Test engineer 1
Test engineer 2
Test engineer 3
Test engineer 4

	c	**Draw a diagram for defect life cycle and write example for defect template.**	

	Ans	**Defect life cycle**	Defect life cycle diagram : 3 M; defect template : 3 M


NEW
DROPPED
ASSIGNED
DEFERRED
COMPLETED
REASSIGNED
CLOSED

OR


NEW
OPEN
REJECTED
ASSIGN
DEFERRED
REOPENED
TEST
VERIFIED
CLOSED

Defect template:

ID	Unique identifier given to the defect. (Usually Automated)
Project	Project name.
Product	Product name.
Release Version	Release version of the product. (e.g. 1.2.3)
Module	Specific module of the product where the defect was detected.
Detected Build Version	Build version of the product where the defect was detected (e.g. 1.2.3.5)
Summary	Summary of the defect. Keep this clear and concise.
Description	Detailed description of the defect. Describe as much as possible but without Repeating anything or using complex words. Keep it simple but comprehensive.
Steps to Replicate	Step by step description of the way to reproduce the defect. Number the steps.
Actual Result	The actual result you received when you followed the steps.
Expected Results	The expected results.
Attachments	Attach any additional information like screenshots and logs.
Remarks	Any additional comments on the defect.
Defect Severity	Severity of the Defect.

Example of Defect Template: (Varies defect wise):

ID	**R1**

Project	**Cash Simulator Cash (ATM)**
Product	http://www.motc.gov.qa/en/ditoolkit/migrantworkers/cash-machine-simulator-atm

Release Version	v1.0
Module	Home Page> Our Programs > Digital Inclusion tools
Detected Build Version	V1.1
Summary	Limited denomination options in cash withdrawal function, restricting cash withdrawal only till 3000.
Description	No option of withdrawing of amount excess of 3000.
Steps to Replicate	1) Open the website 2) Select our programs 3) Proceed to Digital Inclusion tools and select cash machine simulator (ATM) 4) Select language and skip to simulator 5) Enter the card 6) Select the account type 7) Go to Other functions and select cash withdrawal
Expected Results	It should add more options in denominations in withdrawal function or it should take amount input from the user.
Actual Results	It is displaying limited options of denominations in cash withdrawal option.

Attachments	Cash Machine Simulator (ATM) 
Remarks	Causes inconvenience to the user in terms of limited cash withdrawal options.
Defect Severity	High
Defect Priority	High
Reported By	Test Engineer1
Assigned To	XYZ
Status	Assigned

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

WINTER – 2022 EXAMINATION

Subject Name: Software Testing **Model Answer** **Subject Code:** 22518

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**Define the testing terminology i) Error ii) Fault iii) Defect iv) Bug**	**2 M**
	Ans	i) Error: : An error is a human action that produces the incorrect result ii) Fault: State of software caused by an error iii) Defect: A defect is an error or a bug, in the application which is created. A programmer while designing and building the software can make mistakes or error. These mistakes or errors mean that there are flaws in the software. These are called defects. iv) Bug: The presence of error at the time of execution of the software.	½ M for each definition
	b)	**List the levels of testing.**	**2 M**
	Ans	Following are the levels of testing: a) Unit test b) Integration test	½ M for each level

7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

		c) System test d) Acceptance test	
	c)	**State any four needs to prepare a test plan.**	**2 M**
	Ans	Need of test plan: ● Test Plan Ensures all Functional and Design Requirements are implemented as specified in the documentation. ● Test plan gives detail aspects such as test scope, test estimation, strategy, etc. ● Test plan determines the time, cost, and effort. ● It helps in determining the quality of software applications. ● Provide a schedule for testing activities. ● Test Plan Document can be used for similar projects. ● It helps to understand the test details.	1/2 M for each need
	d)	**Give the defect classification and its meaning.**	**2 M**

	Ans	**Requirement/Specification Defects:** Requirement-related defects arise in a product when one fails to understand what the customer requires. These defects may be due to the customer gap, where the customer is unable to define his requirements. Producer gap, where the developing team is not able to make a product as per requirements. **Design Defects:** Design defects occur when system components, interactions between system components, interactions between the outside software/hardware, or users are incorrectly designed. Design defects generally refer to the way of design creation or its usage while creating a product. **Coding Defects:** This defect arises when variables are not initialized properly or variables are not declared correctly or database is not created properly. Coding also needs adequate commenting to make it readable and maintainable in future. **Testing Defects:** These would encompass incorrect, incomplete, missing inappropriate test cases and test procedures.	½ M for each classification and meaning

	e)	**Compare verification and validation (any two points).**	**2 M**

	Ans		1 M for each point (2 point)

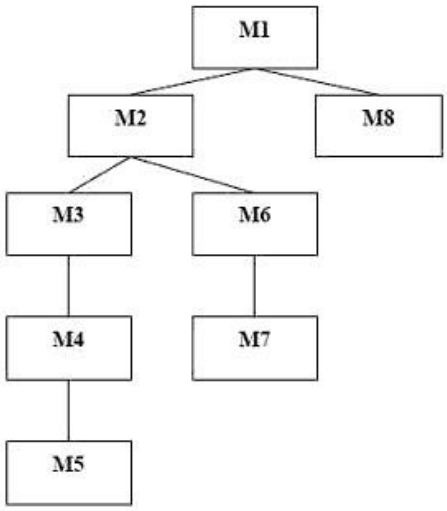
verification	**validation**
It includes checking documents, design, codes, and programs.	It includes testing and validating the actual product.
Verification is the static testing.	Validation is the dynamic testing.
It does not include the execution of the code.	It includes the execution of the code.
Methods used in verification are reviews, walkthroughs, inspections, and desk checking.	Methods used in validation are Black Box Testing, White Box Testing, and nonfunctional testing.
It checks whether the software conforms to specifications or not.	It checks whether the software meets the requirements and expectations of a customer or not
Quality assurance team does verification.	Validation is executed on software code with the help of testing team.

	f)	**State the need of automated testing tools.**	**2 M**

	Ans	• An automated testing tool can playback pre-recorded and predefined actions, compare the results to the expected behavior and report the success or failure of these to a test engineer. • Once automated tests are created, they can easily be repeated, and they can be extended to perform tasks impossible with manual testing. • Automated Software Testing Saves Time and Money. • Software tests must be repeated often during development cycles to ensure quality. • Every time source code is modified software tests should be repeated. • For each release of the software, it may be tested on all supported operating systems and hardware configurations. Manually repeating these tests is costly and time consuming • Once created, automated tests can be run repeatedly at no additional cost, and they are much faster than manual tests. • Testing Improves Accuracy, Even the most conscientious tester will make mistakes during monotonous manual testing. • Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results.	1/2 M for each need (any 4 should write)

		• They can even be run on multiple computers with different configurations. • Automated software testing can look inside an application and see memory contents, data tables, file contents, and internal program states to determine if the product is behaving as expected.	
	g)	**Give the objectives of software testing.**	**2 M**
	Ans	• To find any defects or bugs that may have been created when the software was being developed • To increase confidence in the quality of the software • To prevent defects in the final product • To ensure that end product meets customer requirements as well as specifications • To provide customers with a quality product and increase their confidence in the team.	1 point for each objective (any 2 points are required)

2.		**Attempt any THREE of the following:**	**12 M**
	a)	**State the Entry and Exit criteria's for the software testing.**	**4 M**
	Ans	**Entry criteria** Entry criteria are the condition or the set of conditions, which should exist or be met in order to start a process. Some of the conditions or situations, which may be seen as an entry criterion for the initiation of testing activities. • Requirements should be clearly defined and approved. • Test Design and documentation plan is ready. • Availability of the test environment supporting necessary hardware, software, network configuration, settings, and tools for the purpose of test execution. • Testers are trained, and necessary resources are available. • Availability of proper and adequate test data (like test cases). • It depends upon which software development model is used. **Exit criteria** Exit Criteria is often viewed as a single document concluding the end of a life cycle phase. Some of the conditions or situations which may be seen as an exit criterion for testing activities. • Testing Deadline • Completion of test case execution. • Completion of Functional and code coverage to a certain point.	2 M for entry criteria and 2 M for exit criteria
		• Bug rates fall below a certain level and no high priority bugs are identified. • Management decision.	
	b)	**State and describe top-down approach of integration testing with diagram.**	**4 M**

	Ans	**Top-down integration** ➢ Modules are integrated by moving downward through the control hierarchy, beginning with the main module. ➢ It takes help of dummy program called stub for testing. ➢ Subordinate modules are incorporated in either a depth-first or breadth-first fashion. Integration can be done in two ways: • Depth First Method: All modules on a major control path are integrated. • Breadth First method: All modules directly subordinate at each level are integrated.  **Fig-Top-down integration** **Incremental approach→ Top-down integration procedure** 1. Main control module used as a test driver and stubs are substitutes for components directly subordinate to it. 2. Subordinate stubs are replaced one at a time with real components. (Following the depthfirst or breadth-first approach). 3. Tests are conducted as each component is integrated. 4. On completion of each set of tests and other stub is replaced with a real component. 5. Regression testing may be used to ensure that new errors not introduced.	1 M for diagram 3 M for explanation
	c)	**Describe the "Test Infrastructure' components with diagram.**	**4 M**
	Ans	Testing requires a robust infrastructure to be planned upfront. This infrastructure is made up of three essential elements.	Component/ Diagram-1 M Explaination-3 M

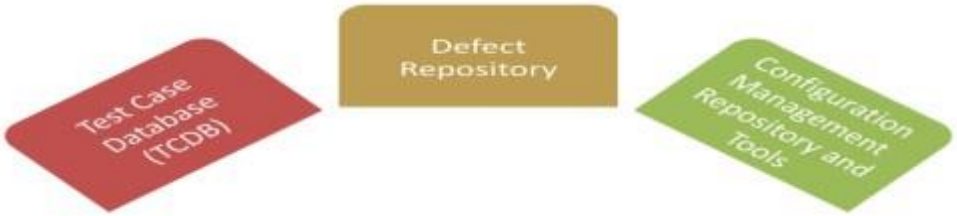


Fig: Components of Test Infrastructure

1. **A test case database (TCDB):** A test case database captures all the relevant information about the test cases in an organization. Some of the entities and the attributes are given in the following table.

Sr. No.	Test Case	Purpose	Attributes
1	Test case	Records all static information about tests.	1)Test case Id 2) Test case name (File name) 3) Test case owner 4) Associated files for test case.
2	Test case product cross reference	Provide mapping between the tests and the corresponding product features, enables identification of test cases for given feature.	Test case Id Module Id
3	Test case run history	Gives the history of when the test case was run and what was result, provided inputs on selection of test for regression runs	1) Test case Id 2) Run date 3) Time taken 4) Run status (Success/ Failure)

4	Test case defect cross reference	Gives details of test cases introduced to test certain specific defects detected in the product, provides inputs on the selection of test for regression runs.	1) Test case Id 2) Defect reference

2. **Defect Repository**: It captures relevant details of defects. It is a tool of communication. Defects matrices are derived from defect repository.
3. **Configuration management repository and tools**: They keep track of change control of all the files/entities that make up a software product. They keep track of version control of all files/entities that makeup a software product.

	d)	**State the limitations of manual testing.**	**4 M**
	Ans	i. Manual testing is slow and costly. ii. It is very labour intensive; it takes a long time to complete tests. iii. Manual tests don't scale well. As the complexity of the software increases the complexity of the testing problem grows exponentially. This leads to an increase in the total time devoted to testing as well as the total cost of testing. iv. One tester may approach and perform a certain test differently from another, resulting in different results on the same test, because the tests are not being performed identically. v. GUI objects size difference and color combinations are not easy to find in manual testing. vi. Not suitable for large scale projects and time bound projects	Write any 4 limitations for 4 M
3.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Differentiate between white box testing and black box testing (any four points).**	**4 M**
	Ans		Any 4 Points1 M each

Sr. No.	Black Box Testing	White Box Testing
1.	It is a way of software testing in which the internal structure or the program or the code is hidden, and nothing is known about it.	It is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software.
2.	It can be referred to as outer or external software testing.	It is the inner or the internal software testing.
3.	It is a functional test of the software.	It is a structural test of the software.

		4.	This testing can be initiated based on the requirement specifications document.	This type of testing of software is started after a detailed design document.	
		5.	It is the behavior testing of the software.	It is the logic testing of the software.	
		6.	It is also called closed testing.	It is also called clear box testing.	
		7.	Can be done by trial-and-error ways and methods.	Data domains along with inner or internal boundaries can be b tested. tter	
	b)	**State the contents of "Test Summary Reports" used in test reporting.**			**4 M**

	Ans	Test reporting is a means of achieving communication through the testing cycle. There are 3 types of test reporting. 1. Test incident report: 2. Test cycle report: 3. Test summary report: **Test summary Report**: The final step in a test cycle is to recommend the suitabilit product for release. A report that summarizes the result of a test cycle is the test sum report. **There are two types of test summary report:** 1. **Phase wise test summary**, which is produced at the end of every phase. 2. **Final test summary report**, which has all the details of testing done by all phas Summary report should be presented. 1. Test Summary Report Identifier 2 Description: Identify the test items being reported in this report with test id 3 Variances: Mention any deviation from test plans, test procedures, if any. 4 Summary of results: All the results are mentioned here with the resolved incidenttheir solutions. 5 Comprehensive assessment and recommendation for release should include: F release assessment and recommendation of release.	Types – 1 M Contents-3 M
	c)	**Prepare defect report after executing test cases for any login form.**	**4 M**
	Ans	Defect Report in Software Testing is a detailed document about bugs found in the soft application Following is Defect report after executing test cases for **Email-log in form.**	Format of defect report- 2 M

ID number	#123
Name	loginform - Unable to login Email
Reporter	Person's name (xyz)
Submit Date	03/01/2023
Summary	When I put my mail id and password, I am unable to

	login while login credentials are right.
URL	www.gmail.com
Screenshot	https://accounts.google.com/signin/
Platform	AngularJS
Operating System	OS X 10.12.0
Browser	Chrome 53
Severity	Major
Assigned to	/
Priority	High

ɾription

n I put mail id and password, I am unable to login while login credentials are .

s to reproduce go to

www.gmail.com

ɪck on login button

t Right mail id and password and click next.

take Screenshot.

ɛcted result

mail account should logged in after putting the right mail id and password.

ıal result

mail account is not logging in after putting the right details.

Description-2 M

	d)	**Enlist the factors considered for selecting a testing tool for test automation.**	**4 M**

	Ans	The following factors are important during tool selection: i. Assessment of the organization's maturity (e.g., readiness for change). ii. Identification of the areas within the organization where tool support will help to improve testing processes. iii. Evaluation of tools against clear requirements and objective criteria. iv. Proof-of-concept to see whether the product works as desired and meets the requirements and objectives defined for it. v. Evaluation of the vendor (training, support and other commercial aspects) or opensource network of support.	Any 4 factors- 1 M each

		vi. Identifying and planning internal implementation (including coaching and mentoring for those new to the use of the tool).	
4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**Describe graphical user interface (GUI) testing and its important traits.**	**4 M**

	Ans	GUI Testing ● There are two types of interfaces for a computer application. ● Command Line Interface is where you type text and the computer responds to that command. ● GUI stands for Graphical User Interface where you interact with the computer using images rather than text. ● GUI testing is the process of testing the system's Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc. ● GUI is what the user sees. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not. **GUI Testing Guidelines** 1. Check Screen Validations 2. Verify All Navigations 3. Check usability Conditions 4. Verify Data Integrity 5. Verify the object states 6. Verify the date Field and Numeric Field Formats **Advantages of GUI Testing:** ● Good GUI improves the feel and look of the application; it psychologically accepts the application by the user. ● GUI represents a presentation layer of an application. Good GUI helps an application due to better experience of the users. ● Consistency of the screen layouts and designs improves usability of an application.	Description - 2 M Importance -2 M

	b)	**Describe test deliverables in details.**	**4 M**

	Ans	1) Test Deliverables are the artifacts which are given to the stakeholders of a software project during the software development lifecycle. There are different test deliverables at every phase of the software development lifecycle. Some test deliverables are provided before the testing phase, some are provided during the testing phase and some after the testing cycle is over. The different types of Test deliverables are: ● Test cases Documents ● Test Plan ● Testing Strategy ● Test Scripts ● Test Data ● Test Traceability Matrix ● Test Results/reports ● Test summary report ● Install/config guides ● Defect Reports ● Release notes 2) The test plan describes the overall method to be used to verify that the software meets the product specification and the customer's needs. It includes the quality objectives, resource needs, schedules, assignments, methods, and so forth. 3) Test cases list the specific items that will be tested and describe the detailed steps that will be followed to verify the software. 4) Bug reports describe the problems found as the test cases are followed. These could be done on paper but are often tracked in a database 5) Test tools and automation are listed and described which are used to test the software. If the team is using automated methods to test software, the tools used, either purchased or written in-house, must be documented. 6) Metrics, statistics, and summaries convey the progress being made as the test work progresses. They take the form of graphs, charts, and written reports	Definition-1M Types-1 M any other 4 points - 2 M
	c)	**Describe load testing and stress testing with suitable example.**	**4 M**

	Ans	**Load Testing** ● Load Testing is a type of performance testing to check system with constantly increasing the load on the system until the time load reaches its threshold value. ● Here Increasing load means increasing number of concurrent users, transactions & check the behavior of the application under test. ● It is normally carried out underneath controlled environment to distinguish between two different systems. ● The main purpose of load testing is to monitor the response time and staying power of application when the system is performing well under heavy load. ● The successfully executed load testing is only if the specified test cases are executed without any error in allocated time. ● Load testing is testing the software under customer expected load.	Description of each- 1 M each

		● In order to perform load testing on the software you feed it all that it can handle. Operate the software with the largest possible data files. ● If the software operates on peripherals such as printer, or communication ports, connect as many as you can. ● If you are testing an internet server that can handle thousands of simultaneous connections, do it. With most software it is important for it to run over long periods. ● Some software should be able to run forever without being restarted. So, Time acts as an important variable. Load testing can be best applied with the help of automation tools. **Simple examples of load testing**: ● Testing printers by sending large jobs. ● Editing a very large document for testing of word processor ● Continuously reading and writing data into the hard disk. ● Running multiple applications simultaneously on the server. ● Testing of mail server by accessing thousands of mailboxes ● In case of zero-volume testing & system fed with zero load **Stress Testing** ● Stress Testing is performance testing type to check the stability of software when hardware resources are not sufficient like CPU, memory, disk space etc. ● It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum. ● Main parameters to focus during Stress testing are "Response Time" and "Throughput". ● Stress testing is Negative testing where we load the software with large number of concurrent users/processes which cannot be handled by the systems hardware resources. This testing is also known as **Fatigue testing.** Stress testing is testing the software under less-than-ideal conditions. So, subject your software with low memory, low disk space, slow CPU, slow modems and so on. Look at your software and determine what external resources and dependencies it has. Stress testing is simply limiting them to a bare minimum. With stress testing you starve the software. For e.g. Word processor software running on your computer with all available memory and disk space, it works fine. But if the system runs low on resources, you have a greater potential to expect a bug. Setting the values to zero or near zero will make the software execute a different path as it attempts to handle the tight constraint. Ideally the software would run without crashing or losing data	Example of each- 1 M each
	d)	**State the advantages and disadvantages of using tools.**	**4 M**

	Ans	**Advantages of using tools:** **Save Time /Speed:** Due to advanced computing facilities, automation test tools prevail in speed of processing the tests. Automation saves time as software can execute test cases faster than humans.	Advantages any 4 - 2 M

		Reduces the tester's involvement in executing tests: It relieves the testers to do some other work. **Repeatability/Consistency:** The same tests can be re-run in exactly the same manner eliminating the risk of human errors such as testers forgetting their exact actions, intentionally omitting steps from the test scripts, missing out steps from the test script, all of which can result in either defects not being identified or the reporting of invalid bugs (which can again, be time consuming for both developers and testers to reproduce) **Simulated Testing**: Automated tools can create many concurrent virtual users/data and effectively test the project in the test environment before releasing the product. **Test case design:** Automated tools can be used to design test cases also through automation, better coverage can be guaranteed than if done manually. **Reusable:** The automated tests can be reused on different versions of the software, even if the interface changes. **Avoids human mistakes**: Manually executing the test cases may incorporate errors. But this can be avoided in automation testing. **Internal Testing**: Testing may require testing for memory leakage or checking the coverage of testing. Automation can do this easily. **Cost Reduction:** If testing time increases, the cost of the software also increases. Due to testing tools time and therefore cost is reduced. **Disadvantages of using tools:** • Unrealistic expectation from the tool • People always make mistake by understanding time cost and effort for the initial introduction of the tool • People frequently miscalculate the time and effort needed to achieve significant and continuing benefits from the tools • Mostly people underestimate the effort required to maintain the test assets generated by the tool • People depend on the tool a lot. (Over reliance on the tool)	Disadvantages 2 M
	e)	**Write the test cases for Notepad application. (any eight test case)**	**4 M**

Ans

PROJECT:	**NOTEPAD**
MODULE:	**FIND AND REPLACE**
FUNCTIONAL SPECIFICATION:	**FIND AND REPLACE**
TEST CASE NO: -	**TC-FR-1**
TEST OBJECTIVE: -	**To Check functionality of " Find and Replace" in notepad.**

ENVIRONMENT:WIN 2k, Notepad.

TC#	**Test Scenario**	**Pre-Condition**	**Test Steps**	**Test Data**		**Expected Result**	**Actual Result**	**Remarks**	
1&2	Check the availability of the Find Option.	-	Click the Edit menu from the menu bar.			After clicking on the **find**,the window should pop up	The search box available.	Pass	Any eight valid test cases ½ M each
3-a)	Check the navigation through Shortcut keys	-	Press Ctrl + F			After pressing CTRL + F , should produce the search box	The find box available.	Pass	
3-b)	Check the navigation through Shortcut keys	-	Press Ctrl + H			After pressing CTRL + H, should produce the replace box	The Replace box available.	Pass	

3-C)	Check the navigation through mouse	-	Click Edit menu from menu bar and then FIND		In the menu by clicking the edit and then find, the search box should open	The find Box is available.	Pass
4	The cursor default position.	Find box should be open.			The cursor should be present in the typing space box.	The cursor is Available	Pass
5	To Check find Button.	Find box should be open.	Press Find Button		Without typing anything, the **find** button should not be enabled and functional	It is not enabled.	Pass
6	To check the Search control in the page.	Find box should be open.	Press Find Button		The typed text in the search field should match, otherwise generate an error that word does not exist.	If it is matching, then it highlights it.	Pass
7	If the user want to search a single word, more than once. Then after competition of 1st search, the search button should be enable for next also	Find box should be open.	Press Find Button, Find Next		The Search/Find button should be enabled for the next search also.	The find button is available for next search also.	Pass

			8	To check the Replace control.	Find box should be open.	Click on Replace		The user should be asked before replacing any word.	The message is not coming.	Fail	
			9-a)	Replace the exact work	Find box should be open.	Click on Replace		By clicking "match whole word only", it should replace only the whole word.	It is replacing the whole word only.	Pass	
			9-b)	Don't replace when there is no text in replace space	Find box should be open.	Click on Replace		Don't replace when there is no text in replace space	It is replacing the find value with the blank space.	Fail	
			10	Check Cancel button functionality	Find box should be open.	Click Cancel		After pressing the cancel button the window should exit.	The window is exit.	pass	
5.		**Attempt any <u>TWO</u> of the following:**									**12 M**
	a)	**Design test cases for simple calculator application. (Black box testing.) (Any six points.)**									**6 M**

Ans

Test Case ID	Test case Objective	Input data	Expected Result	Actual Result	Status
TC-1	To add two integer and display the result on tendigit calculator	176 + 100	276	276	Pass
TC-2	To subtract two integer and display the result on ten-digit calculator	176 - 100	76	76	Pass
TC-3	To multiply two integer and display the result on ten-digit calculator	100 x 20	2000	2000	Pass
TC4	To divide two integer and display the result on ten-digit calculator	100/ 5	20	20	Pass
TC5	To clear the screen		Symbol "0" should appear on screen	Symbol "0" appears on screen	Pass

6 test cases of test cases for simple calculator application: 6 M; 1M each; any other valid test cases shall be considered

TC6	To delete digits one by one		One Digit should be deleted from right hand side	One Digit is deleted from right hand side	Pass

	b)	**Design test cases for Web pages testing of any Web site (take a suitable example).**	**6 M**

Ans

Test Case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Check cursor position at email or mobile number field	Click on email or mobile number field	Cursor should be placed on the field	Placed the cursor on the field	Pass
TC2	Check cursor position at password field	Click on password field	Cursor should be placed on the password field	Placed the cursor on the password field	Pass
TC3	Check the continue button	Click on continue button	It should redirect to password page	It redirected to the password page.	Pass
TC4	Readability of font	Try to read the contents on login page	Contents should be readable	Content s are readable	Pass

6 test cases of test cases for any web site: 6 M; 1M each; any other valid test cases shall be considered

			TC5	Testing of spelling of login	Check the spelling of login	Login spelling should be correct	Spelling of Login is correct	Pass	
			TC6	Testing of hyperlink	Hover the mouse on hyperlink	It should change the cursor and should redirect to respective page on click	Cursor changed and redirects to other page.	Pass	

	c)	**Design test cases for MS Word application using an Automation tool.**	**6 M**

Ans

Test Case ID	Test case objective	Input data	Expected result	Actual result	Status
TC1	Check whether Undo in Edit main menu undoes the previous action		Previous action should be undone	Previous action was undone	Pass
TC2	Checks whether the Undo button in right click context menu undoes the previous action		Previous action should be undone	Previous action was undone	Pass
TC3	Checks whether Undo button in the Edit main menu is disabled when there is not any previous actions		Undo Button should be disabled	Undo Button was disabled	Pass
TC4	Checks whether Undo button in right context menu is disabled when there are not any previous actions		Undo Button should be disabled	Undo Button remained disabled	Pass
TC5	Checks whether hotkey (CTRL+Z) response when there is no any of previous actions		No response is expected	No response	Pass
TC6	Checks whether the Cut options in Edit main menu cuts the selected text		Selected text should be cut	Selected text was cut	Pass

6 test cases of test cases for any function of MS Word: 6 M; 1M each; any other valid test cases shall be considered

TC7	Checks whether the Cut options in Edit Menu is disabled when no texts are selected		Cut Options should be disabled	Cut Option Was Disabled	Pass

Above test cases will be executed on any automation tool like autoIT, QTP etc.

6.		**Attempt any TWO of the following:**	**12 M**
	a)	**Write program for calculating even numbers from 1 to 20 And design the test cases for same.**	**6 M**
	Ans	Program : #include <stdio.h> int main() { for(int i=0;i<=20;i=i+2) { printf("%d\n",i); } return 0; }	Correct program: 2M; valid test cases shall be considered:4 M

Test Case ID	EV_001	**Test Case Description**	Test program prints the even number from 1 to 20		
Created By	ABC	**Reviewed By**	PQR	**Version**	2.1
Tester's Name	ABC	**Date Tested**	1Jan-2023	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites :
1	C program using for loop

S #	Test Data

Test Scenario	Verify the even number

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended

		1	Check initial condition of for loop	Initial value of For loop should be 0 or 1	Initial value of For loop is 0 or 1	Pass		
		2	Check final condition of for loop	Final condition should be "< 20" or "<=20"	Final condition is "< 20" or "<=20"	Pass		
		3	Check the increment operator	Increment operator should increment by 2	Counter is incremented by 2	Pass		
		4	Check output	Even number is displayed on output screen	It is displaying even number	Pass		
	b)	**Prepare test plan for "Cam Scanner' which is installed on mobile.**						**6 M**

	Ans	**Test plan for Cam Scanner:** **Test Plan Identifier** TP_10 **Introduction:** The purpose of this document is to create test plan for CamScanner application installed on mobile. The purpose of testing this program is to check the correct operation of its functionality, ease of use. **Test Items:** Working with the document (Scan document, Edit document, PDF conversion) **Features to be tested** ● Scan Document ● Edit Document ● PDF Conversion **Approach** ● On the test object: ○ functional ○ non-functional ● According to the requirements ○ positive ○ negative ● By degree of preparedness - intuitive testing (ad hoc) **Item Pass/Fail Criteria:** All test cases with high priority are closed with the result - pass. The test coverage is checked and sufficient, where the criterion of sufficiency is not less than 99% of the coverage of requirements by tests. The test report was compiled and approved by the team lead and customer. **Suspension Criteria and Resumption Requirements** Criterion for interrupting testing: ● The appearance and entering into the bug-tracking system of blocking bugs. Criterion for continuation of testing: ● Closing the blocking bug in the bug tracking system. **Test Deliverables:** Test plan, test cases, test report. **Test Tasks** ● Writing a test plan ● Writing test cases ● Development of criteria for the success of testing ● Conducting the testing and evaluation of the results ● Creating test reports	

Environmental Needs
Mobile Phone
CamScanner Installed
Responsibilities

Sr. no	Functionality and Responsibilities	Responsible
1	Scan Document	Test Engineer 1
2	Edit Document	Test Engineer 1
3	PDF Conversion of Document	Test Engineer 3

Staffing and Training Needs
To perform the tasks, you need to have the following knowledge and skills:

- knowledge and practical application of the camscaaner;
- knowledge and ability to apply in practice the basic techniques of test desig
- Knowledge of various types of testing including functional and non-functio

Schedule
The deadline for completion of all works and delivery of the project is 25/01/2 5.00pm
Risks and Contingencies
Possible risks during testing:

- Insufficient human resources for testing the application in deadlines. ●
  Changing the requirements for the product

Approvals
Team Lead
Test engineer 1
Test engineer 2

	c)	**Prepare defect report after executing test cases for withdrawn of amount from ATM machine.**	**6 M**
	Ans		any valid defect report related with withdrawal functionality shall be considered

ID	R1
Project	**Cash Simulator Cash (ATM)**
Product	http://www.motc.gov.qa/en/ditoolkit/migrantworkers/cash - machine-simulator-atm
Release Version	v1.0
Module	Home Page> Our Programs > Digital Inclusion tools
Detected Build Version	V1.1

Summary	Limited denomination options in cash withdrawal function, restricting cash withdrawal only till 3000
Description	No option of withdrawing of amount excess of 3000.
Steps to Replicate	1. Open the website 2. Select our programs 3. Proceed to Digital Inclusion tools and select cash machine simulator (ATM) 4. Select language and skip to simulator 5. Enter the card 6. Select the account type 7. Go to Other functions and select cash withdrawal
Expected Results	It should add more options in denominations in withdrawal function or it should take amount input from the user.
Actual Results	It is displaying limited options of denominations in cash withdrawal option.
Attachments	
Remarks	Causes inconvenience to the user in terms of limited cash withdrawal options.
Defect Severity	High
Defect Priority	High
Reported By	Test Engineer1
Assigned To	XYZ
Status	Assigned

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

WINTER – 2023 EXAMINATION
Model Answer – Only for the Use of RAC Assessors

Subject Name: Software Testing **Subject Code:** 22518

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.
8) As per the policy decision of Maharashtra State Government, teaching in English/Marathi and Bilingual (English + Marathi) medium is introduced at first year of AICTE diploma Programme from academic year 2021-2022. Hence if the students in first year (first and second semesters) write answers in Marathi or bilingual language (English +Marathi), the Examiner shall consider the same and assess the answer based on matching of concepts with model answer.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a)	**Compare Verification and validation. (any 2 points)**	**2 M**
	Ans		comparison of verification and validation: 2 points: 2 M,

Verification	Validation
Are we building the system, right?	Are we building the right system?

		Verification is the process of evaluating products of a development phase to find out whether they meet the specified requirements.	**Validation** is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements.	1 M each
		Execution of code is not comes under **Verification**.	Execution of code is comes under **Validation**.	
		Verification is carried out before the Validation.	**Validation** activity is carried out just after the Verification.	
		Cost of errors caught in **Verification** is less than errors found in Validation.	Cost of errors caught in **Validation** is more than errors found in Verification.	
		It is basically manually checking the of documents and files like requirement specifications etc.	It is basically checking of developed program based on the requirement specifications documents & files.	
	b)	**Define failure, error, fault, bug.**		**2 M**
	Ans	**Failure:** It is the inability of a system or component to perform the required function according to its specification. OR External behavior is incorrect. **Error:** Refers to difference between Actual Output and Expected output. OR An error is a human action that produces the incorrect result. **Fault:** It is a condition that causes the software to fail to perform its required function. OR Discrepancy in code that causes a failure. **Bug:** The presence of error at the time of execution of the software. OR A bug can be defined as the initiation of error or a problem due to which fault, failure, incident or an anomaly occurs.		Definition of failure, error, fault, bug: ½ M each, total 2 M
	c)	**List the objectives of software testing (any four).**		**2 M**

	Ans	**Objectives of software testing:** **1. Finding error:** Finding defects which may be created by the programmer while developing the software. **2. Quality improvement:** Gaining confidence in and providing information about the level of quality. **3. Creating good test cases**: Good test case is one that has a high probability of finding undiscovered error. **4. Meets users' requirements:** To make sure that the result meets the business and user requirements. **5. Satisfying Requirements:** To ensure that it satisfies the BRS that is Business Requirement Specification, and SRS that is System Requirement Specifications. **6.** To gain the confidence of the customers by providing them a quality product	2 objectives of software testing: 2 M; 1 M each
	d)	**Define driver and stub.**	**2 M**
	Ans	**Driver:** Drivers are dummy modules that are always used to simulate the high-level modules. Drivers are only used when main programs are under construction. Drivers are used in bottom-up integration. **Stub:** Stubs are dummy modules that always used to simulate the low-level modules. Stubs are used when sub programs are under construction. Stubs are used in top-down approach.	Definition of driver: 1 M; definition of stub: 1 M
	e)	**What is GUI testing? Give one example.**	**2 M**
	Ans	**GUI Testing:** GUI stands for Graphical User Interface where you interact with the computer using.	GUI Testing: 1 M; any valid

		images rather than text. ● GUI testing is the process of testing the system's Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes and windows, etc. GUI is what the user sees. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not. **Examples of GUI testing includes:** 1. Check Screen Validations 2. Verify All Navigations 3. Check usability Conditions 4. Verify Data Integrity 5. Verify the object states 6. Verify the date Field and Numeric Field Formats	example: 1 M; total 2 M
	f)	**Write any two root causes of defect.**	**2 M**
	Ans	Root causes of defect are: i. Miscommunication of requirements introduces error in code. ii. Lack of design Experience. iii. Lack of coding practice. iv. Unrealistic time schedule for development. v. Multiple changes in the requirements. **OR** **Any other valid answer shall be given marks.**	Two root causes of defect: 2 M; 1 M each
	g)	**Enlist any four software testing tools.**	**2 M**
	Ans	4 software testing tools are: 1. Selenium 2. Test complete 3. LoadRunner 4. Cucumber 5. Quick test professional (QTP) 6. Cypress **(Any valid software testing tool shall be given marks)**	any 4 software testing tools: 2 M; ½ M each

2.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**State the entry and exit criteria for software testing.**	**4 M**

	Ans	**Entry criteria** Entry criteria are the condition or the set of conditions, which should exist or be met to start a process. Some of the conditions or situations, which may be seen as an entry criterion for the initiation of testing activities. • Requirements should be clearly defined and approved. • Test Design and documentation plan is ready. • Availability of the test environment supporting necessary hardware, software, network configuration, settings, and tools for the purpose of test execution. • Testers are trained, and necessary resources are available. • Availability of proper and adequate test data (like test cases). • It depends upon which software development model is used. **Exit criteria.** Exit Criteria is often viewed as a single document concluding the end of a life cycle phase. Some of the conditions or situations which may be seen as an exit criterion for testing activities. • Testing Deadline • Completion of test case execution. • Completion of Functional and code coverage to a certain point. • Bug rates fall below a certain level and no high priority bugs are identified. • Management decision.	entry criteria: 2 M exit criteria: 2 M; total 4 M
	b)	**Illustrate process of bi-directional integration testing. State it's two advantages and disadvantages.**	**4 M**

	Ans	• Bidirectional integration testing strategy is a combination of Top Down and Bottom-up approaches. • Here, top modules are tested with lower modules at the same time lower modules are integrated with top modules and tested. • This strategy makes use of stubs as well as drivers. • bidirectional integration testing is a culmination of both incremental as well as non-incremental integration testing, wherein Bottom-Up approach is focused on middle to top layer, Top-Down approach is concerned about layers from middle to downwards and the Big Bang approach is followed for the middle layer. • This type of testing combines the advantages of all the three approaches and is mainly used to test large projects. **Advantages:** 1. Bidirectional integration testing is very useful for large enterprises and huge projects that further have several subprojects. 2. When development follows a spiral model and the module itself is as large as a system, then one can use Bidirectional integration testing. 3. Top-Down and Bottom-Up approach both start as per development schedule. 4. Units are tested and brought together to make a system. 5. Integration is done downwards. 6. The resources that are required are immense and big teams perform both topdown and bottom-up method of testing at a time or one after the other. Disadvantages:	Process of Bidirectional integration testing: 2 M; 2 advantages: 1 M; 2 disadvantages: 1 M; total 4 M

		1. As both Top-Down and Bottom-Up approaches are executed on the software, the cost of testing is very high. 2. It cannot be used for smaller systems with huge interdependence between the modules. 3. It only makes sense when the individual subsystem is as good as the completed system. 4. Different skill sets are required for testers at different levels.	
	c)	**Enlist four attributes of defect. Describe them with suitable example.**	**4 M**

	Ans	**Attributes of defect:** 1) Defect ID: Identifies defect as there are many defects might identified in system. a. i.e. D1, D2, etc. 2) Defect Name: Name of defect which explains the defect in brief. a. It must be short but descriptive. i.e. Login error. 3) Project Name: Indicates project name in which defect is found e.g.: Library management system 4) Module /Sub-module name: for which the defect is found. E.g. Login form 5) Phase introduced: Phase of life cycle to which the defect belongs to. e.g. 2 6) Phase found: Phase of project when the defect is found is added here. It is used to find defect leakage or stage. 7) Defect type: Defines defect type. e.g. security defects, functional defect, GUI defect etc. 8) Severity: Declared in test plan, e.g. high medium or low. 9) Priority: defines based on how the project decides a schedule to take the defects for fixing. e.g. High, low, Moderate 10) Summary: Describes short about the defect. 11) Description: Describes it in detail. 12) Status: dynamic field, open, assigned, resolved, closed, hold, deferred, or reopened, etc. 13) Reported by/ Reported on: Who found defect, and on what date. 14) Assigned to: The tester is being assigned to some testing team member.	4 attributes of defect with example: 4 M; 1 M each
	d)	**Describe any four factors for selecting a testing tool.**	**4 M**
	Ans	**4 factors for selection of testing tools are:** **The following factors are important during tool selection:** i. **Assessment of the organization's maturity** (e.g. readiness for change); ii. Identification of the areas within the organization where **tool support will help to improve testing processes;** iii. **Evaluation of tools against clear requirements and objective**	4 factors for selection of testing tools: 4 M; 1 M each

criteria;

iv. **Proof-of-concept to see whether the product works as desired** and meets the requirements and objectives defined for it;

v. **Evaluation of the vendor (training, support and other commercial aspects**) or open-source network of support; vi. **Identifying and planning internal implementation** (including coaching and mentoring for those new to the use of the tool).

OR (SOME CONTENTS FROM FOLLOWING FACTORS ALSO SHALL BE GIVEN MARKS)

The industry experts have suggested following four major criteria's for selection of testing tools.

1) Meeting requirements.
2) Technology expectations.
3) Training / skills.
4) Management aspects.

1.Meeting requirements

- There are plenty of tools available in the market but rarely do they meet all the requirements of a given product or a given organization.
- Evaluating different tools for different requirements involve significant effort, money, and time. • The tool must match its intended use.
- Wrong selection of a tool can lead to problems like lower efficiency and effectiveness of testing may be lost.

1. Technology expectations:

- Test tools in general may not allow test developers to extends/modify the functionality of the framework So, extending the functionality requires going back to the tool vendor and involves additional cost and effort.
- Different phases of a life cycle have different quality-factor requirements. Tools required at each stage may differ significantly.

2. Training/skills:

- While test tools require plenty of training, very few vendors provide the training to the required level. • Organization level training is needed to deploy the test tools.
- As the user of the test suite are not only the test team but also the development team and other areas like configuration management.

- If the testers do not have proper training and skill, then they may not be able to work effectively.

3. Management aspects:

- A test tool increases the system requirement and requires the hardware and software

to be upgraded.

- This increases the cost of the already- expensive test tool.
- Select affordable tools. Cost and benefits of various tools must be compared before making final decision.

3. Attempt any THREE of the following: **12 M**

a) Differentiate between Alpha Testing and Beta Testing. (any four points) **4 M**

Ans **Differences Between Alpha Testing and Beta Testing:** 1 M each Differentiations

Alpha Testing	Beta Testing
Alpha Testing performed at developer's site.	Beta testing is performed at a client location or end user of the product.
Alpha testing involves both the white box and black box techniques.	Beta Testing typically uses Black Box Testing.
Alpha testing performed by Testers who are usually internal employees of the organization.	Beta testing is performed by Clients or End Users who are not employees of the organization.
Alpha testing requires a lab environment or testing environment	Beta testing doesn't require any lab environment or testing environment. The software is made available to the public and is said to be real time environment
Alpha testing is to ensure the quality of the product before moving to Beta testing	Beta testing also concentrates on the quality of the product, but gathers users input on the product and ensures that the product is ready for real time users.
Critical issues or fixes can be addressed by developers immediately in Alpha testing.	Most of the issues or feedback is collected from Beta testing will be implemented in future versions of the product.
Long execution cycle may be required for Alpha testing.	Only a few weeks of execution are required for Beta testing

b) Prepare Test Plan for Notepad Application. (Windows based) **4 M**

	Ans	1			Consider any 8 points
			Test Plan Identifier	TP_10	
		2	**Introduction:**	The purpose of this document is to create and test plan for edit .functionality of notepad The purpose of testing this program is to check the correct operation of its functionality	

				and ease of use.	
		3	**Test Items**	Working with notepad	
		4	**Features to be tested.**	• Undo • Cut /Copy • Paste • DELETE • Find • Goto	
		5	**Approach**	• On the test object: o functional o non-functional • According to the requirements o positive o negative • By degree of preparedness - intuitive testing (ad hoc)	
		6	**Item Pass/Fail Criteria:**	All test cases with high priority are closed with the result - pass. The test coverage is checked and sufficient, where the criterion of sufficiency is not less than 99% of the coverage of requirements by tests. The test report was compiled and approved by the team lead and customer.	

7	**Suspension Criteria:** **Resumption Criteria:**	The appearance and entering the bug-tracking system of blocking bugs. Closing the blocking bug in the bug tracking system
8	**Test Deliverables**	Test plan, test case specification, test case, test summary report
9	**Test Tasks**	• Writing a test plan • Writing test cases • Development of criteria for the success of testing. • Conducting the testing and evaluation of the results • Creating test reports
10	**Environmental needs**	Notepad Laptop/Computer
11	**responsibilities**	

Functionality and Responsibilities	Responsible
Undo	Test Engineer 1
Cut/copy	Test Engineer 2
paste	Test Engineer 3
delete	Test Engineer 4
find	Test Engineer 5
goto	Test Engineer 5

12	**Staffing and Training Needs**	To perform the tasks, you need to have the following knowledge and skills: • practical knowledge application of the notepad is needed. • knowledge and ability to apply in practice the basic techniques of test design. • Knowledge of various types of testing including functional and non-functional.

		13	**Schedule**	The deadline for completion of all works and delivery of the project is 31/12/2023 by 5.00pm	
		14	**Risks and Contingencies Possible risks during testing**	• Insufficient human resources for testing the application in deadlines. • Changing the requirements for the product	
		15	**Approvals Team**	Lead Test engineer 1 Test Manager Quality Manager	
	c)	**Explain defect Management Process with suitable diagram.**			**4 M**
	Ans	 i. **Defect Prevention**-- Implementation of techniques, methodology and standard processes to reduce the risk of defects.			2 M for Diagram 2 M for Explanation

		ii. **Deliverable Baseline**-- Establishment of milestones where deliverables will be considered complete and ready for further development work. When a deliverable is base lined, any further changes are controlled. Errors in a deliverable are not considered defects until after the deliverable is base lined. iii. **Defect Discovery**-- Identification and reporting of defects for development team acknowledgment. A defect is only termed discovered when it has been documented and acknowledged as a valid defect by the development team member(s) responsible for the component(s) in error. iv. **Defect Resolution**-- Work by the development team to prioritize, schedule and fix a defect, and document the resolution. This also includes notification back to the tester to ensure that the resolution is verified. v. **Process Improvement** -- All problems are due to failure in the process involved in creating software. Defects give an opportunity to identify the problem with process used and update them. Better processes mean better product with less defect. vi. **Management Reporting** -- Analysis and reporting of defect information to assist management with risk management, process improvement and project management.	
	d)	**State and explain any four benefits of automation in testing.**	**4 M**

	Ans	• **Reduces time of testing** Software tests have to be repeated often during development cycles to ensure quality. Every time source code is modified software tests should be repeated. For each release of the software it may be tested on all supported operating systems and hardware configurations. Manually repeating these tests is costly and time consuming. Once created, automated tests can be run over and over again at no additional cost and they are much faster than manual tests. • **Greater accuracy:** Continuous testing increases the chances of errors while done manually, but in automated testing, repetitive tests can be performed with the same precision. • **Deliver the quality product** The exposure and expertise of a manual tester determine how thoroughly your applications get tested. Test automation, when implemented accurately, removes your dependency on these parameters, delivering expected results every time. Yet another aspect that impacts quality is manual errors. Irrespective of how meticulous a quality engineer is, there is always a possibility of missing a few steps, entering the wrong data, or any trivial manual error – leading to poor application quality. With test automation, you can ensure that your solution will execute the steps accurately without fail and report each step without bias • **Allow to run tests many time with different data** They can even be run on multiple computers with different configurations. Automated software testing can look inside an application and see memory contents, data tables, file contents, and internal program states to determine if the product is behaving as expected. Test automation can easily execute thousands of different complex test cases during every test run providing coverage that is impossible with manual tests.	1 M for each benefit

		• **Save resources or requires less** Manual testing, especially regression testing, can seem extremely tedious as you need to test every minor code change. Writing scripts, running them over and over again is nothing short of a nightmare. With no-code test automation, you need not write test cases or execute them manually when the codebase changes. Instead, your solution creates the test scripts which you can reuse and execute as required without additional efforts or costs.	
4.		**Attempt any <u>THREE</u> of the following:**	**12 M**
	a)	**What is boundary value analysis? Explain with suitable example.**	**4 M**

	Ans	• Most of the defects in software products hover around conditions and boundaries. • Boundary value analysis is another black box test design technique, and it is used to find the errors at boundaries of input domain rather than finding those errors in the center of input. • Each boundary has a valid boundary value and an invalid boundary value. • Test cases are designed based on both valid and invalid boundary values. Typically, we choose one test case from each boundary. The basic idea in boundary value testing is to select input variable values at their: 1. Minimum 2. Just below the minimum 3. Just above the minimum 4. Just below the maximum 5. Maximum 6. Just above the maximum **Example: Input Box should accept the Number 1 to 10** Here we will see the Boundary Value Test Cases	Explanation – 2 M, Example - 2 M

Test Scenario Description	Expected Outcome
Boundary Value = 0	System should NOT accept
Boundary Value = 1	System should accept
Boundary Value = 2	System should accept
Boundary Value = 9	System should accept
Boundary Value = 10	System should accept
Boundary Value = 11	System should NOT accept

	b)	**Explain the Regression Testing. State when the Regression testing shall be done?**	**4 M**

	Ans	Regression testing a black box testing technique that consists of re-executing those tests that are impacted by the code changes. ii. These tests should be executed as often as possible throughout the software development life cycle. Types of Regression Tests: i. **Final Regression Tests: -** A "final regression testing" is performed to validate the build that hasn't changed for a period of time. This build is deployed or shipped to customers. ii. **Regression Tests: -** A normal regression testing is performed to verify if the build has NOT broken any other parts of the application by the recent code changes for defect fixing or for enhancement. **When to Perform:** Regression testing should be performed whenever there is a change in the software code, configuration, or environment. For example, regression testing should be done after adding new features, fixing bugs, refactoring code, integrating components, updating libraries, or migrating to a different platform. Regression testing should also be done before releasing the software to the end-users or deploying it to the production environment.	2 M (for explanation related with regression testing) 1 M
	c)	**What is Test Plan? What is its need? List test planning activities.**	**4 M**

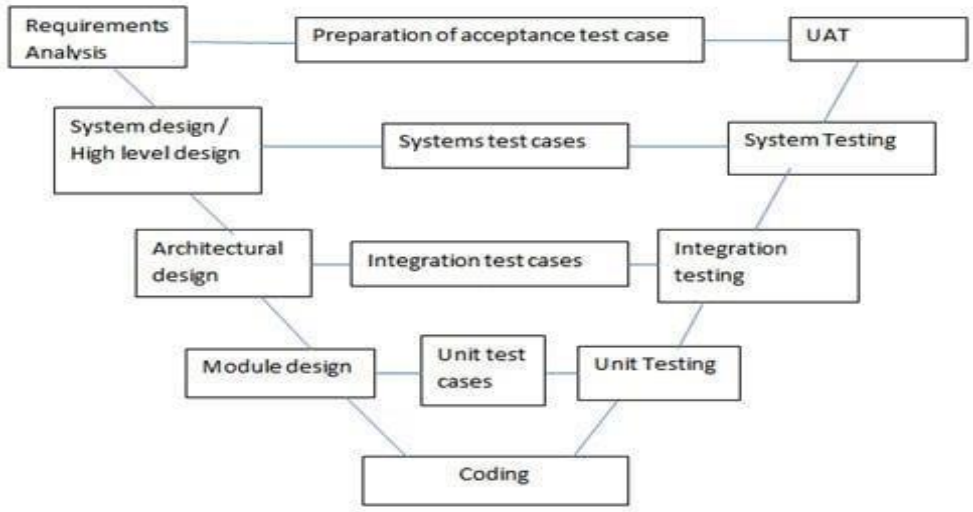
	Ans	**Test Plan:** i. Test plan is the project plan for the testing work to be done. ii. It is not a test design *specification,* a collection of *test cases* or a set of *test procedures;* in fact, most of our test plans do not address that level of detail. **Need:** i. Test Plan Ensures all Functional and Design Requirements are implemented as specified in the documentation. ii. To provide a procedure for Unit and System Testing. iii. To identify the documentation process for Unit and System Testing. iv. To identify the test methods for Unit and System Testing. **Activities:** 1. Scope Management: Deciding what features to be tested and not to be tested. 2. Deciding Test approach /strategy: Which type of testing shall be done like configuration, integration, localization etc. 3. Setting up criteria for testing: There must be clear entry and exit criteria for different phases of testing. The test strategies for the various features and combinations	1 M 1 M 2 M

		determined how these features and combinations would be tested. 4. Identifying responsibilities, staffing and training needs. 5. Identifying resource requirements. 6. Identifying test deliverables. 7. Testing tasks: size and effort estimation.	
	d)	**Prepare defect report for login field of email application.**	**4 M**

	Ans	A defect report is a document that describes a defect, including its severity, priority, and steps to replicate the problem. A defect report's primary purpose is to help the developers quickly reproduce and fix the fault. **Defect Report Template** • **Defect id :** • **Project Name :** • **Module Name :** • **Sub Module Name :** • **Type of Defect : (wrong, missing or extra)** • **Status : (New, open, assign, fix)** • **Severity : (high, medium, low)** • **Priority : (high, medium, low)** • **Summary :** • **Description : (Steps To Reproduce)** • **Expected Result :** • **Actual Results :** • **Reported By :** • **Assign To :** • **Date & Time:** **Example: Defect Report for User Login Page** • **Defect id:** D001 • **Project Name:** Web-Site Domain Name • **Module Name:** Login • **Sub Module Name:** User Login • **Type of Defect:** Missing • **Status:** New • **Severity:** High • **Priority :** • **Summary:** User Login Page Not Opening • **Description :** 1. Enter Url / Domain Name 2. Click On the User Login Option • **Expected Result:** User login page should get open	2 M 2 M

		• **Actual Results:** User login page does not get open • **Reported By: ABC Tester** • **Assign To: XYZ Developer** • **Date & Time: 12/28/2022**	
	e)	**State any four limitations of manual testing.**	**4 M**
	Ans	1. Manual testing is slow and costly. 2. It is very labor intensive; it takes a long time to complete tests. 3. Manual tests don't scale well. As the complexity of the software increases the complexity of the testing problem grows exponentially. This leads to an increase in total time devoted to testing as well as total cost of testing. 4. Manual testing is not consistent or repeatable. Variations in how the tests are performed as inevitable, for various reasons. One tester may approach and perform a certain test differently from another, resulting in different results on the same test, because the tests are not being performed identically. 5. Lack of training is the common problem. 6. GUI objects size difference and color combinations are not easy to find in manual testing. 7. Not suitable for large scale projects and time bound projects. 8. Batch testing is not possible, for each test execution Human user interaction is mandatory. 9. Comparing large amounts of data is impractical. 10. Processing change requests during software maintenance takes more time.	1 M for each limitation
5.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Describe V-model with labelled diagram.**	**6 M**

	Ans	• The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. • It is also known as Verification and Validation model. • It is based on the association of a testing phase for each corresponding development stage. • Development of each step is directly associated with the testing phase. • The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it. • V-Model contains Verification phases on one side of the Validation phases on the other side. • Verification and Validation phases are joined by coding phase in V-shape.	Diagram-2M, Explanation of Verification Phase- 2 M, Explanation of Validation Phase-2 M OR Answer with Relevant Contents



Verification Phase (Design Phase):

It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

1. **Requirement Analysis:** This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.
2. **System Design:** This phase contains the system design and the complete hardware and communication setup for developing product.
3. **Architectural Design:** System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood.
4. **Module Design:** In this phase the system breaks dowm into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

Validation (Testing Phases) :

It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

1. **Unit Testing:** Unit Test Plans are developed during module design phase. These Unit Test Plans are executed to eliminate bugs at code or unit level.
2. **Integration testing:** After completion of unit testing Integration testing is performed. In integration testing, the modules are integrated and the system is

tested. Integration testing is performed on the Architecture design phase. This test verifies the communication of modules among themselves.

3. **System Testing:** System testing test the complete application with its functionality, inter dependency, and communication. It tests the functional and non-functional requirements of the developed application.

		4. User Acceptance Testing (UAT): UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets user's requirement and system is ready for use in real world.	
	b)	**Describe with one example each : i. Load Testing ii. Stress Testing**	**6 M**

	Ans	**i. Load Testing** • Load Testing is a type of performance testing to check system with constantly increasing the load on the system until the time load reaches its threshold value. • Here Increasing load means increasing number of concurrent users, transactions & check the behavior of the application under test. • It is normally carried out underneath controlled environment to distinguish between two different systems. • The main purpose of load testing is to monitor the response time and staying power of application when the system is performing well under heavy load. • The successfully executed load testing is only if the specified test cases are executed without any error in allocated time. • Load testing is testing the software under customer expected load. • In order to perform load testing on the software you feed it all that it can handle. Operate the software with the largest possible data files. • If the software operates on peripherals such as printer, or communication ports, connect as many as you can. • If you are testing an internet server that can handle thousands of simultaneous connections, do it. With most software it is important for it to run over long periods. • Some software should be able to run forever without being restarted. So, Time acts as an important variable. Load testing can be best applied with the help of automation tools. **Examples of load testing:** • Downloading a series of large files from the internet. • Running multiple applications on a computer or server simultaneously. • Assigning many jobs to a printer in a queue. • Subjecting a server to a large amount of traffic. • Writing and reading data to and from a hard disk continuously **ii. Stress Testing** • It is a type of non-functional testing. • It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. • It is a form of software testing that is used to determine the stability of a given system. • It put greater emphasis on robustness, availability and error handling under a heavy load, rather than on what would be considered correct behaviour under normal circumstances. • The goals of such tests may be to ensure the software does not crash in conditions of insufficient computational resources (such as memory, disk space, network request etc.) • Stress testing is also called fatigue testing.	Description of each- 2 M Any Suitable example of each – 1 M

• **For Example:**

Word processor software running on your computer with all available memory and disk space, it works fine. But if the system runs low on resources, you have a great potential error to expect a bug. Setting the values to zero or near zero will make the software execute a failure different path as it attempts to handle the tight constraint. Ideally the software would run without crashing or losing data.

c) Prepare six test cases for marketing site www.flipkart.com **6 M**

Ans

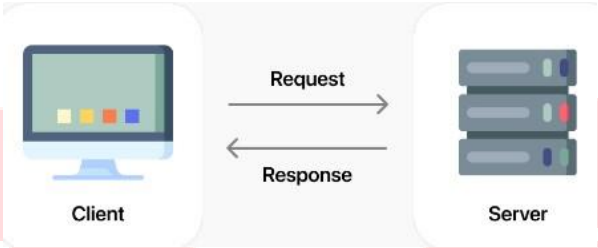
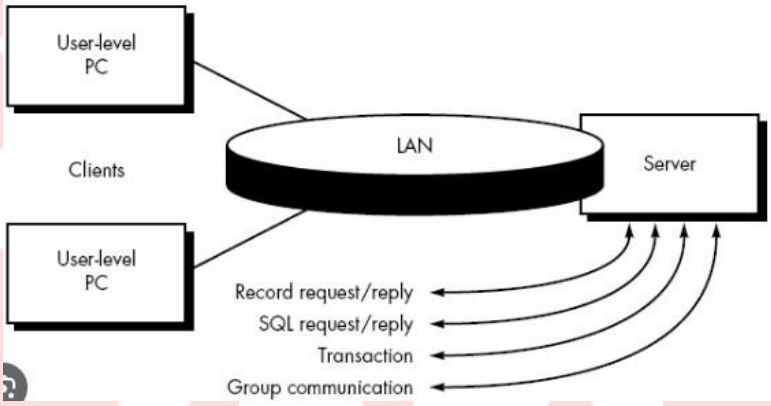
Test Case -ID	Steps	Input data	Expect ed Result	Actual Result	Status
TC-1	Type correct user name	Abc123	It Should accept user name	It accepts user name	pass
TC-2	Type correct and valid password	Co5i518	It Should accept password	It accepts password	pass
TC-3	Click on login button	-	Home page should be displayed after login and user name should be displayed on home page	Home page is displayed after login and user name should be displayed on home page	Pass
TC-4	Click on any product displayed on home page	-	User should be redirected to product specification page.	User redirected to product specification page	pass
TC-5	Click on add to cart for the product.		The product should be added to cart.	The product added to cart	Pass
TC-6	Click on go to cart button.		The total amount of all items in cart should be displayed.	The total amount of all items in cart is displayed.	pass
TC-7	Click on remove from cart button.	-	The item should be removed form cart.	The item is removed from the cart.	pass

6 test cases of test cases: 6 M; 1 M each;

any other valid test cases shall be considered

			TC-8	Click on Checkout button.	-	The checkout page should be displayed with payments options.	The checkout page is displayed with payments options.	Pass	
			TC-9	Make payment for the order.	-	User should get order details by message or email.	User gets order details by message or email.	Pass	

6.		**Attempt any <u>TWO</u> of the following:**	**12 M**
	a)	**Explain client-server testing with suitable example.**	**6 M**

	Ans	In Client-server testing there are several clients communicating with the server.   1. Multiple users can access the system at a time, and they can communicate with the server. 2. Configuration of client is known to the server with certainty. 3. Client and server are connected by real connection. 4. Testing approaches of client server system: • **Component Testing:** One need to define the approach and test plan for testing client and server individually. When server is tested there is need of a client simulator, whereas testing client a server simulator, and to test network both simulators are used at a time. • **Integration testing:** After successful testing of server, client and network, they are brought together to form system testing. • **Performance testing:** System performance is tested when number of clients is communicating with server at a time. Volume testing and stress testing may be used for testing, to test under maximum load as well as normal load expected. Various interactions may be used for stress testing. • **Concurrency Testing:** It is very important testing for client-server architecture. It may be possible that multiple users may be accessing same record at a time, and concurrency testing is required to understand the behavior of a system in this situation. • **Disaster Recovery Business continuity testing:** When the client server are communicating with each other, there exit a possibility of breaking of the communication due to various reasons or failure of either client or server or link	Description - 3 M, Any suitable example - 1 M

connecting them. The requirement specifications must describe the possible expectations in case of any failure.

- **Testing for extended periods:** In case of client server applications generally server is never shutdown unless there is some agreed Service Level Agreement (SLA) where server may be shut down for maintenance. It may be expected that server is running 24X7 for extended period. One needs to conduct testing over an extended period to understand if service level of network and server deteriorates over time due to some reasons like memory leakage.
- **Compatibility Testing:** Client server may be put in different environments when the users are using them in production. Servers may be in different hardware, software, or operating system environment than the recommended. Other testing such as security testing and compliance testing may be involved if needed, as per testing and type of system.

For example, the average number of users working simultaneously on a system must be quantified, since performance testing most commonly tests performance under workload stress. Testers should also determine maximum or peak user performance or how the system operates under maximum workloads. Bandwidth is another necessary bit of information, as is most users most frequent actions. Performance testing also validates and verifies other performance parameters such as reliability and scalability. Performance testing can establish that a product lives up to performance standards necessary for commercial release. It can compare two systems to determine which one performs better. Or they can use profilers to determine the programs behavior as it runs. This determines which parts of the program might cause the most trouble and it establishes thresholds of acceptable response times.

b) Write important six test cases for the "Login Form" of the Facebook website. **6 M**

Ans

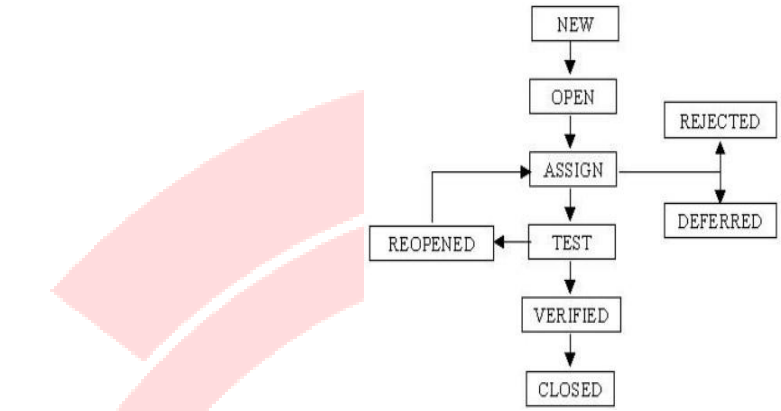
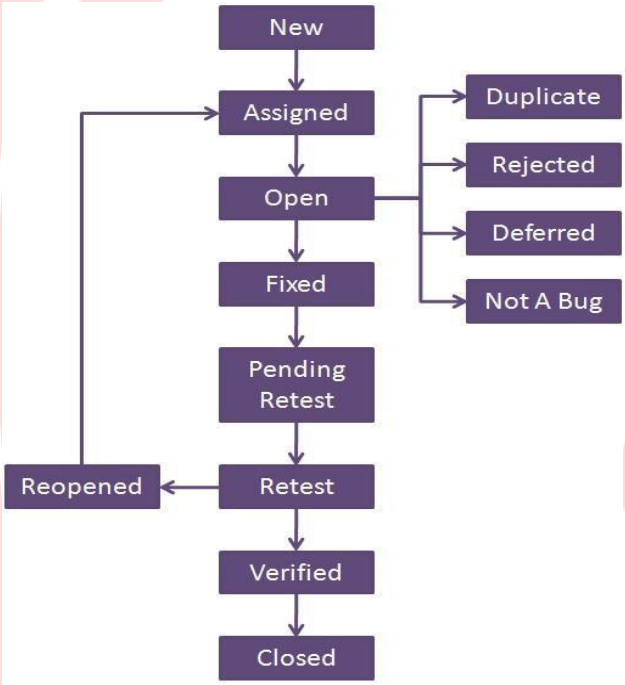
Test_case_no	Test step	Test data	Expected output	Actual output	Status
1	Username filed is left blank		It will display 'Enter Username'	It displays 'Enter Username'	Pass
2	Enter invalid user name	abc	It will prompt 'couldn't find your account' message	It prompt couldn't find your account.	Pass

6 test cases of test cases: 6 M;

1 M each;

any other valid test cases shall be considered

		3	Enter valid user name and invalid password	Username-abc123 Password – 123	It will display 'Wrong password' message.	It displays 'wrong password' message.	Pass	
		4	Enter Valid username and no password	Username-abc123 Password –	It will display 'Enter password'.	It displays 'Enter password'.	Pass	
		5	Enter Valid username and Password	Username-abc123 Password – co5i22518	It will display users' account's facebook page.	It displays users account's facebook page.	Pass	
		6	Click on 'Forgotten password?'	-	It will go to Find your account page.	It goes to Find your account page.	Pass	
	c)	**Describe defect life cycle with neat diagram.**						**6 M**

	Ans	 OR  **1. New:** When a defect is logged and posted for the first time. It's state is given as new.	Diagram: 2 M Description: 4 M

2. **Assigned:** After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. It's state given as assigned.
3. **Open:** At this state the developer has started analysing and working on the defect fix.
4. **Fixed:** When developer makes necessary code changes and verifies the changes then he/she can make bug status as 'Fixed' and the bug is passed to testing team.
5. **Pending retest:** After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.
6. **Retest:** At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.
7. **Verified:** The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to "verified".
8. **Reopen:** If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "reopened". The bug goes through the life cycle once again.
9. **Closed:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to "closed". This state means that the bug is fixed, tested and approved.
10. **Duplicate:** If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to "duplicate**".**
11. **Rejected:** If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to "rejected".
12. **Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
13. **Not a bug:** The state given as "Not a bug" if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and feel of the application like change of colour of some text then it is not a bug but just some change in the look of the application.