Jaywant Shikshan Prasarak Mandal's

RAJARSHI SHAHU COLLEGE OF ENGINEERING's

POLYTECHNIC

S.No.80, Pune-Mumbai Bypass Highway, Tathawade Campus, Pune.

Approved By AICTE & Govt. of Maharashtra, Affilated to MSBTE

NBA ACCREDITED

DTE Code – 6141 **MSBTE Inst. Code - 1620**

"MASTER SOLUTIONS"

(MSBTE Question Paper with Solution as per I Scheme)

Class –Third Year Computer Engineering

Semester – CO5I

Subject: Software Testing

Subject Code – 22518

DEPARTMENT OF COMPUTER ENGINEERING

JSPM's

RAJARSHI SHAHU COLLEGE OF ENGINEERING, POLYTECHNIC, TATHAWADE, PUNE-33

DEPARTMENT OF COMPUTER ENGINEERING

JSPM's RSCOE Polytechnic **Institute Vision**	JSPM's RSCOE Polytechnic **Institute Mission**
"To satisfy the aspirations of youth force, who want to lead the nation towards prosperity throughtechno-economic development."	"To provide, nurture and maintain an environment of high academic excellence, research and entrepreneurship for all aspiring students, which will prepare them to face global challenges maintaining high ethical and moral standards."
JSPM's RSCOE Polytechnic **Department of Computer Engineering** **Vision**	**JSPM's RSCOE Polytechnic** **Department of Computer Engineering** **Mission**
To impart value based technical education for developing competent computer engineers fulfilling expectations of industry and society.	**M1-**To provide sound theoretical education, practical knowledge and to train the students in association with industry. **M2-**To improve self awareness and ethical values among students along with technical proficiency. **M3**-To promote awareness about life-long learning and problem solving among students.
Program Specific Outcomes (PSOs)	**Program Outcomes (POs)**
PSO1.Computer Software and Hardware Usage; Use state of-the-art technologies for operation and application of computer software and hardware. **PSO2.Computer engineering Maintenance**; Maintain computer engineering related software and hardware system.	**1. Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and Engineering fundamentals and engineering specialization to solve the engineering problems. **2. Problem analysis:** Identify and analyse well-defined engineering problems using codified standard methods. **3. Design/ development of solutions:** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs. **4. Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements. **5. Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices. **6. Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities. **7. Life-long learning:** Ability to analyse individual needs and engage in updating in the context of technological changes.
Program Educational Objectives (PEOs)	
PEO1: Provide socially responsible, environment friendly solutions to Computer Engineering related broad-based problems adapting professional ethics. **PEO2**: Adapt state-of-the-art Computer engineering broad-based technologies to work in multi-disciplinary work environments. **PEO3**: Solve broad-based problems individually and as a team member communicating effectively in the world of work.	

Maharashtra State Board of Technical Education, Mumbai

Teaching and Examination Scheme for Post S.S.C. Diploma Courses

Program Name : Computer Engineering Groups

Program Code : CO/CM/CW — **With Effect From Academic Year: 2017 - 18**

Duration of Program : 6 Semesters — **Duration : 16 Weeks**

Semester : Fifth — **Scheme : I**

S. N.	Course Title	Course Abbreviation	Course Code	Teaching Scheme			Credit (L+T+P)	Examination Scheme												Grand Total
								Theory						Practical						
				L	T	P		Exam Duration in Hrs.	ESE		PA		Total		ESE		PA		Total	
									Max Marks	Min Marks	Max Marks	Min Marks	Max Marks	Min Marks	Max Marks	Min Marks	Max Marks	Min Marks	Max Marks	Min Marks	
1	Environmental Studies	EST	22447	3	-	-	3	90 Min	70*#	28	30*	00	100	40	--	--	--	--	--	--	100
2	Operating Systems	OSY	22516	3	-	2	5	3	70	28	30*	00	100	40	25@	10	25	10	50	20	150
3	Advanced Java Programming	AJP	22517	3	1	2	6	3	70	28	30*	00	100	40	25#	10	25	10	50	20	150
4	Software Testing	STE	22518	3	-	2	5	3	70	28	30*	00	100	40	25@	10	25	10	50	20	150
	Elective (Any One)																				
4	Client Side Scripting Language	CSS	22519	3	-	2	5	3	70	28	30*	00	100	40	25#	10	25	10	50	20	150
	Advanced Computer Network	ACN	22520	3	-	2	5	3	70	28	30*	00	100	40	25#	10	25	10	50	20	150
	Advanced Dabase Management Systems	ADM	22521	3	-	2	5	3	70	28	30*	00	100	40	25#	10	25	10	50	20	150
5	Industrial Training	ITR	22049	-	-	6	6	--	--	--	--	--	--	--	75#	30	75~	30	150	60	150
6	Capstone Project Planning	CPP	22050	-	-	2	2	--	--	--	--	--	--	--	25@	10	25~	10	50	20	50
			Total	**15**	**1**	**16**	**32**	--	**350**	--	**150**	--	**500**	--	**200**	--	**200**	--	**400**	--	**900**

Student Contact Hours Per Week: **32 Hrs.** Medium of Instruction: **English**

Theory and practical periods of 60 minutes each. Total Marks : **900**

Abbreviations: ESE- End Semester Exam, PA- Progressive Assessment, L - Lectures, T - Tutorial, P - Practical

@ Internal Assessment, # External Assessment, *# On Line Examination, ^ Computer Based Assessment

* Under the theory PA, Out of 30 marks, 10 marks are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the cognitive domain LOs required for the attainment of the COs.

~ For the courses having ONLY Practical Examination, the PA marks Practical Part - with 60% weightage and Micro-Project Part with 40% weightage

- **If Candidate not securing minimum marks for passing in the "PA" part of practical of any course of any semester then the candidate shall be declared as "Detained" for that semester.**
- **Evalution of Industrial Training and its reports is to done during this semester. Credit of Industrial Training will not affect the framing of time table.**

Program Name : **Computer Engineering Program Group**

Program Code : **CO/CM/CW**

Semester : **Fifth**

Course Title : **Software Testing**

Course Code : **22518**

1. RATIONALE

In today's software environment writing bug-free code is challenging task, which make software testing important tool to get the quality software. Testing techniques include the process of executing a program or application with the intent of finding software bugs and verifying that the software product is fit for use. Students will learn the way to find bugs by applying types, levels and methods of software testing on applications with effective test planning approach. It also covers manual testing.

2. COMPETENCY

The aim of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- **Apply types, levels and methods of software testing on applications.**

3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following ***industry oriented*** COs associated with the above mentioned competency:

a. Apply various software testing methods.
b. Prepare test cases for different types and levels of testing.
c. Prepare test plan for an application.
d. Identify bugs to create defect report of given application.
e. Test software for performance measures using automated testing tools.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme											
				Theory						Practical					
L	T	P		Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
3	-	2	5	3	70	28	30*	00	100	40	25@	10	25	10	50	20

(*): Under the theory PA; Out of 30 marks, 10 marks of theory PA are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.

***Legends:** L-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** - Practical; **C** – Credit, **ESE** - End Semester Examination; **PA** - Progressive Assessment.*

5. COURSE MAP (with sample COs, PrOs, UOs, ADOs and topics)

This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the

course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

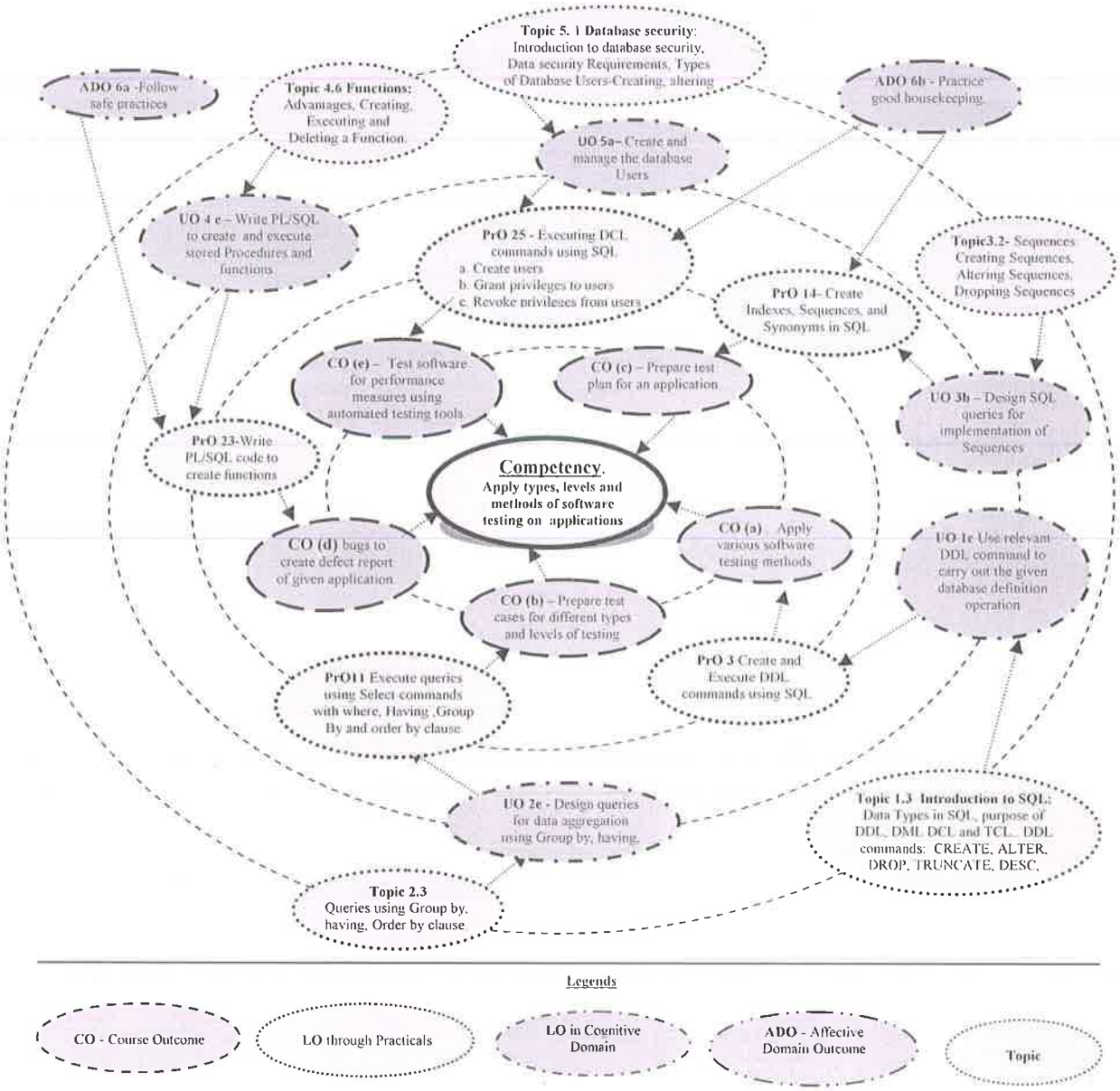


Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Identify system specification & design test cases for purchase order Management.	I	02*
2	Identify system specification & design test cases for Inventory management	I	02*
3	Design test cases for simple calculator application.(BB Testing)	I	02*
4	Design test cases for railway reservation form	II	02*
5	Design test cases for e-commerce (Flipkart, Amazon) login form	II	02*

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
6	Design test cases for Web Pages Testing any Web Sites	II	02*
7	Write program and design test cases for the following Control and decision making statement. 1) For... Loop 2) Switch...case 3) Do...While 4) If...else	II	02*
8	Prepare test plan for an identified Mobile application.	III	02*
9	Design test plan and test cases for Notepad (MS Window based) Application.	III	02*
10	Prepare defect report after executing test cases for library management system	IV	02*
11	Prepare defect report after executing test cases for Withdrawn of amount from ATM Machine.	IV	02
12	Prepare defect report after executing test cases for any login form.	IV	02
13	Design and run test cases for WordPad (MS Windows based). Using an Automated tool.	V	02*
14	Design and run test cases for MS Word application using an Automation Tool.	V	02
15	Project Assignment		04*
	Total		**32**

Note

i. A suggestive list of PrOs is given in the above table. More such PrOs can be added to attain the COs and competency. A judicial mix of minimum 12 or more practical need to be performed, out of which, the practicals marked as '' are compulsory, so that the student reaches the 'Precision Level' of Dave's 'Psychomotor Domain Taxonomy' as generally required by the industry.*

ii. The 'Process' and 'Product' related skills associated with each PrO is to be assessed according to a suggested sample given below:

S. No.	Performance Indicators	Weightage in %
1	Preparation of system specification, designing test plan using MS Excel.	50
2	Preparation of defect report	10
3	Execution of test cases using automation tool.	20
4	Answer to sample questions	10
5	Submit report in time	10
Total		**100**

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:

a. Follow safety practices.
b. Practice good housekeeping.
c. Work as a leader/a team member.
d. Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. However, acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs

according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1st year
- 'Organising Level' in 2nd year and
- 'Characterising Level' in 3rd year.

7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.

S. No.	Equipment Name with Broad Specifications	PrO S. No.
1.1	Computer system (Any computer system with basic configuration)	All
1.2	Selenium	V
1.3	Mantis Bug Tracker	IV
1.4	IBM Rational Functional Tester	V
1.5	Spreadsheet Package	I, II, III
1.6	Bugzila	IV

8. UNDERPINNING THEORY COMPONENTS

The following topics are to be taught and assessed in order to develop the sample UOs given below for achieving the COs to attain the identified competency. More UOs could be added.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit – I Basics of Software Testing and Testing Methods	1a. Identify errors and bugs in the given program. 1b. Prepare test case for the given application. 1c. Describe the Entry and Exit Criteria for the given test application. 1d. Validate the given application using V model in relation with quality assurance. 1e. Describe features of the given testing method.	1.1 Software Testing, Objectives of Testing. 1.2 Failure, Error, Fault, Defect, Bug Terminology. 1.3 Test Case, When to Start and Stop Testing of Software (Entry and Exit Criteria). 1.4 Verification and Validation (V Model), Quality Assurance, Quality Control. 1.5 Methods of Testing: Static and dynamic Testing 1.6 The box approach : White Box Testing: Inspections, Walkthroughs, Technical Reviews, Functional Testing, Code Coverage Testing, Code Complexity Testing. 1.7 Black Box Testing: Requirement Based Testing, Boundary Value Analysis, Equivalence Partitioning,
Unit– II Types and Levels of Testing	2a Apply specified testing level for the given web based application. 2b Apply Acceptance testing for given web based application. 2c Apply the given performance testing for the specified application.	2.1 Levels of testing 2.1 Unit Testing: Driver, Stub 2.2 Integration Testing: Top-Down Integration, Bottom-Up Integration, Bi-Directional Integration 2.3 Testing on Web Application: Performance Testing:Load Testing,

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
	2d Generate test cases for the given application using regression and GUI testing.	Stress Testing, Security Testing. Client-Server Testing 2.4 Acceptance Testing: Alpha Testing and Beta Testing, Special Tests: Regression Testing, GUI Testing,
Unit III- Test Management	3a. Prepare test plan for the given application. 3b. Identify the resource requirement of the given application. 3c. Prepare test cases for the given application. 3d. Prepare test report of executed test cases for given application.	3.1 Test Planning : Preparing a Test Plan, Deciding Test Approach, Setting Up Criteria for Testing, Identifying Responsibilities, Staffing, Resource Requirements, Test Deliverables, Testing Tasks 3.2 Test Management: Test Infrastructure Management, Test People Management. 3.3 Test Process: Base Lining a Test Plan, Test Case Specification. 3.4 Test Reporting: Executing Test Cases, Preparing Test Summary Report.
Unit-IV Defect Management	4a. Classify defects on the basis estimated impact. 4b. Prepare defect template on the given application. 4c. Apply defect management process on the given application. 4d. Write procedure to find defect using the given technique.	4.1. Defect Classification, Defect Management Process. 4.2. Defect Life Cycle, Defect Template 4.3. Estimate Expected Impact of a Defect, Techniques for Finding Defects, Reporting a Defect.
Unit –V Testing Tools and Measurements	5a. Improve testing efficiency using automated tool for given application. 5b. Identify different testing tools to test the given application. 5c. Describe Metrics and Measurement for the given application 5d. Explain Object oriented metrics used in the given testing application	5.1 Manual Testing and Need for Automated Testing Tools 5.2 Advantages and Disadvantages of Using Tools 5.3 Selecting a Testing Tool 5.4 When to Use Automated Test Tools, Testing Using Automated Tools. 5.5 5.6 Metrics and Measurement: Types of Metrics, Product Metrics and Process Metrics, Object oriented metrics in testing.

Note: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'.

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Basics of Software Testing and Testing Method	10	04	04	06	14
II	Types and Levels of Testing	12	04	06	08	18
III	Test Management	10	04	04	06	14
IV	Defect Management	08	04	02	06	12
V	Testing Tools and Measurements	08	02	04	06	12
	Total	**48**	**18**	**20**	**32**	**70**

Legends: *R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)*
Note*: This specification table provides general guidelines to assist student for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.*

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related ***co-curricular*** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

a) Prepare journals based on practical performed in laboratory.
b) Give seminar on relevant topic.
c) Undertake micro-projects.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various learning outcomes in this course:

a) Massive open online courses (***MOOCs***) may be used to teach various topics/sub topics.
b) ***'L' in item No. 4*** does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
c) About ***15-20% of the topics/sub-topics*** which is relatively simpler or descriptive in nature is to be given to the students for ***self-directed learning*** and assess the development of the COs through classroom presentations (see implementation guideline for details).
d) With respect to item No.10, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.
e) Guide student(s) in undertaking micro-projects.
f) Demonstrate students thoroughly before they start doing the practice.
g) Encourage students to refer different websites to have deeper understanding of the subject.
h) Observe continuously and monitor the performance of students in Lab.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be individually undertaken to build up the skill and confidence in every student to become problem solver so

that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should not exceed three.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than 16 (sixteen) student engagement hours during the course. The student ought to submit micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects are given here. Similar micro-projects could be added by the concerned faculty:

a) Library Management: book issue /book stock system.
b) Any other micro-projects suggested by subject faculty on similar line.

13. SUGGESTED LEARNING RESOURCES :

S. No.	Title of Book	Author	Publication
1	Software Testing: Principles and Practices	Srinivasan Desikan Gopalaswamy Ramesh	PEARSON Publisher: Pearson India 2005, ISBN: 9788177581218,
2	Software Testing: Principles, Techniques and Tools	Limaye M. G.	Tata McGraw Hill Education, New Delhi., 2007 ISBN 13: 9780070139909
3	Software Testing: Principles and Practices	Chauhan Naresh	Oxford University Press Noida –
4	Software Testing	Singh Yogesh	Cambridge University Press, Bangluru. ISBN 978-1-107-65278-1

Note: Other available testing tools can be used at institute level.

14. SOFTWARE/LEARNING WEBSITES

a. http://www.selenium.com
b. http://en.wikipedia.org/wiki/Test_automation
c. http://en.wikipedia.org/wiki/Software_testing#Testing_tools
d. http://www.softwaretestingsoftware.com
e. www.toolsqa.com

JSPM's
Rajarshi Shahu College of Engineering,
Polytechnic, Tathawade, Pune-33

Academic Year 2024-25

MSBTE QUESTION PAPER ANALYSIS

Course: Software Testing
Course Code: 22518

Unit No.	Marks as per Teaching Scheme	Marks weightage in S-2024 QP	Marks weightage in W-2023 QP	Marks weightage in S-2023 QP	Marks weightage in W-2022 QP	Marks weightage in S-2022 QP	Marks weightage in W-2019 QP
Unit No. 1	14	10	20	08	20	14	18
Unit No. 2	18	26	28	30	26	26	16
Unit No. 3	14	32	20	32	24	30	36
Unit No. 4	12	14	20	16	12	12	12
Unit No. 5	12	20	14	16	20	20	20
Total Marks	70	102	102	102	102	102	102

Subject In-charge **DAC** **HOD**

JSPM's

Rajarshi Shahu College of Engineering, Polytechnic Tathawade

Department of Computer Engineering

Academic Year: 2024-2025

Course Name: Software Testing **Course code:** 22518

Semester: Odd **Name of the Faculty:** Prof. Suvarna A. Amle

UNIT NO. 1 Basic software testing And Testing Method

CO1: Apply various software testing Methods

Q. No.	Questions	Year	Marks
1	**Define static And dynamic software testing**	W-19	**2**
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.		
2	**-Enlist Objectives of Software testing.** **-Gives Objectives of Software testing.**	W-19 W-22 W-23	**2**
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.		
3	**Define Following terms –** **i) Error.** **ii) Fault.** **iii) Bug.** **iv) Failure**	W-19, S-22, W-22, S-23 W-23	**2**
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 behaviour 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.		
4	**-Describe Boundary value analysis with suitable example.** **-What is Describe Boundary value analysis? Explain With suitable example**	W-19 W-23	**4**
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 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 centre 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 behaviour 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		
5	**Describe the use of decision table in black box testing with the help of suitable example.**	W-19	**4**
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. 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.		

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
Example : X→ No action (Don't care)

6	**Define verification and validation with respect to software testing.**	S-22	**2**
Ans	**Verification** is the process of checking that software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfils the requirements that we have. Verification is static testing. **Validation** is the process of checking whether the software product is up to the mark or in other words product has high-level requirements. It is the process of checking the validation of the product i.e. it checks what we are developing is the right product. It is validation of the actual and expected products. Validation is dynamic testing.		
7	**Describe Code Complexity testing.**	S-22	**4**
Ans	**Program Statements and Line Coverage (Code Complexity Testing)** The most straightforward form of code coverage is called statement coverage or line coverage. If you're monitoring statement coverage while you test your software, your goal is to make sure that you execute every statement in the program at least once. With line coverage the tester tests the code line by line giving the relevant output. For example 1. #include		

	2. void main()
	3. {
	4. int i , fact= 1, n;
	5. printf(—enter the number —);
	6. scanf(—%d‖, &n);
	7. for(i =1 ;i <=n; i++)
	8. fact = fact * i;
	9. printf("\n Factorial of number is %d",fact);
	10.
	}
	Branch Coverage (Code Complexity Testing)
	Attempting to cover all the paths in the software is called path testing. ii. The simplest form of path testing is called branch coverage testing. iii. To check all the possibilities of the boundary and the sub boundary conditions and it's branching on those values.
	Test coverage criteria requires enough test cases such that each condition in a decision takes on all possible outcomes at least once, and each point of entry to a program or subroutine is invoked at least once.
	Condition Coverage (Code Complexity Testing)
	i. Just when you thought you had it all figured out, there's yet another Complication to path testing.
	ii. Condition coverage testing takes the extra conditions on the branch statements into account.
	iii.		

8	**Differentiate between Quality assurance and quality control.**	S-22 S-23	**4**

Ans

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.
Relates to all products that will ever be created by a process	Relates to specific product
Activities of QA are Process Definition and Implementation, Audits and Training	Activities of QC are Reviews and Testing
Verification is an example of QA	Validation/Software Testing is an example of QC
Preventive activities.	It is a corrective process.
Quality assurance is a proactive process	Quality control is a reactive process.
QA is a managerial tool	QC is a corrective tool

9	**Compare Verification And validation (any 2 points)**	W-22 W-23	**2**

Ans

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 non-functional 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.

10	**State the entry and exit criteria's for the software testing**	W-22 W-23	**4**

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.

<table>
<tr><td></td><td>• 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</td><td></td><td></td></tr>
<tr><td>11</td><td>Differentiate between white box testing and black box testing (any 4 Points)</td><td>W-22</td><td>4</td></tr>
<tr><td>Ans</td><td>
<table>
<tr><th>Sr. No.</th><th>Black Box Testing</th><th>White Box Testing</th></tr>
<tr><td>1.</td><td>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.</td><td>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.</td></tr>
<tr><td>2.</td><td>It can be referred to as outer or external software testing.</td><td>It is the inner or the internal software testing.</td></tr>
<tr><td>3.</td><td>It is a functional test of the software.</td><td>It is a structural test of the software.</td></tr>
<tr><td>4.</td><td>This testing can be initiated based on the requirement specifications document.</td><td>This type of testing of software is started after a detailed design document.</td></tr>
<tr><td>5.</td><td>It is the behavior testing of the software.</td><td>It is the logic testing of the software.</td></tr>
<tr><td>6.</td><td>It is also called closed testing.</td><td>It is also called clear box testing.</td></tr>
</table>
</td><td></td><td></td></tr>
<tr><td>12</td><td>Design test cases for simple calculator application. (black box testing) (any six Point)</td><td>W-22</td><td>4</td></tr>
</table>

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 ten-digit 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
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

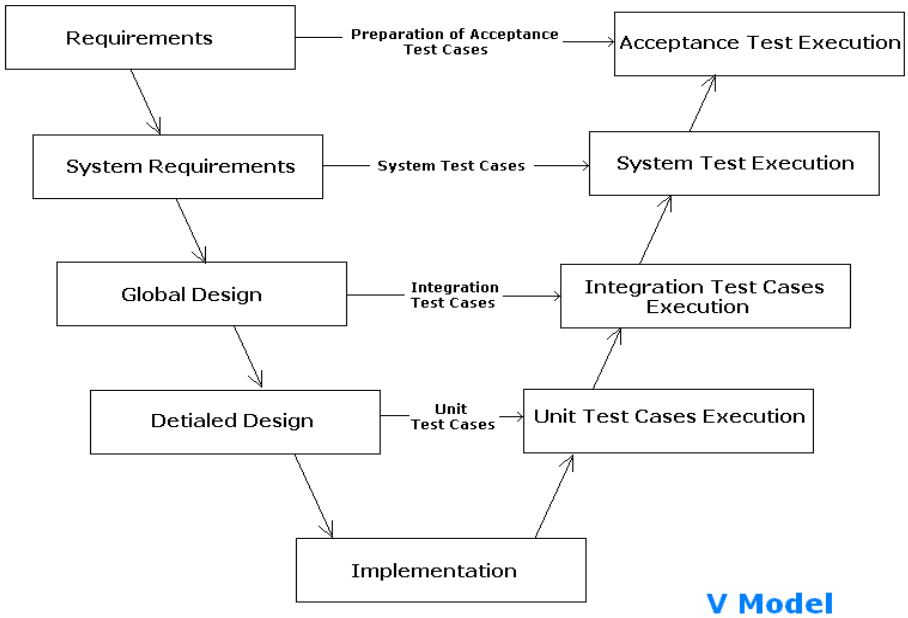
13	**List types of white box testing. Describe any two types of white box testing with neat diagram**	S-23	**4**
Ans	1) Unit testing 2) Integration testing 3) Loop testing 4) Static analysis 5) Dynamic analysis **Unit testing:** Checks if each part or function of the application works correctly. Ensures the application meets design requirements during development. **Integration testing:** Examines how different parts of the application work together. Done after unit testing to make sure components work well both alone and together		
14	**Describe V- Model with labelled diagram**	W-23	**4**
Ans	**Verification Phase:** 1. **Overall Business Requirement:** In this first phase of the development cycle the product requirements are understood from customer perspective. This phase involves detailed communication with the customer to understand his		

expectations and exact requirements. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing.

2. **Software Requirement:** Once the product requirements are clearly known, the system can be designed. The system design comprises of understanding & detailing the complete hardware, software & communication set up for the product under development. System test plan is designed based on system design. Doing this at earlier stage leaves more time for actual test execution later.
3. **High level design:** High level specification are understood & designed in this phase. Usually more than one technical approach is proposed & based on the technical & financial feasibility, the final decision is taken. System design is broken down further into modules taking up different functionality.
4. **Low level design:** In this phase the detailed integral design for all the system modules is specified. It is important that the design is compatible with the other modules in the system & other external system. Components tests can be designed at this stage based on the internal module design,
5. **Coding:** The actual coding of the system modules designed in the design phase is taken up in the coding phase. The base suitable programming language is decided base on requirements. Coding is done based on the coding guidelines & standards.

Validation Phase:

1. Unit Testing: Unit testing designed in coding are executed on the code during this validation phase. This helps to eliminate bugs at an early stage.
2. Components testing: This is associated with module design helps to eliminate defects in individual modules.
3. **Integration Testing:** It is associated with high level design phase & it is performed to test the coexistence & communication of the internal modules within the system
4. **System Testing:** It is associated with system design phase. It checks the entire system functionality & the communication of the system under development with external systems. Most of the software & hardware compatibility issues can be uncovered using system test execution.
5. **Acceptance Testing:** It is associated with overall & involves testing the product in user Environment. These tests uncover the compatibility issues with the other systems available in the user environment. It also uncovers the non-functional issues such as load & performance defects in the actual user environment.


Requirements
Preparation of Acceptance Test Cases
Acceptance Test Execution
System Requirements
System Test Cases
System Test Execution
Global Design
Integration Test Cases
Integration Test Cases Execution
Detialed Design
Unit Test Cases
Unit Test Cases Execution
Implementation
V Model

JSPM's

Rajarshi Shahu College of Engineering, Polytechnic Tathawade

Department of Computer Engineering

Academic Year: 2024-2025

Course Name: Software Testing **Course code:** 22518

Semester: Odd **Name of the Faculty:** Prof. Suvarna A. Amle

UNIT NO. 2 Types and Level of Testing

CO2: Prepare test cases for different types and levels of testing.

Q. No.	Questions	Year	Marks
1	**State any Two Examples of integration testing**	**W-19**	**2**
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.		
2	**Differentiate between drivers and stub (any four points)**	**W-19**	**4**

Ans

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.

3	**Differentiate between alpha testing and beta testing.**	**W-19** **W-23**	**4**

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.
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

4	**State the testing approaches that are considered during client-server testing**	**W-19**	**4**

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 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 behaviour 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 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		
5	**With respect to GUI testing write the test cases for Amazon login form**	**W-19**	**6**

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	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

6	**State the process of performance testing.**	**S-22**	**2**

Ans

Performance Testing

♣ Performance testing is testing that is performed, to determine how fast some aspect of a system performs under a particular workload. It can serve different purposes like it can demonstrate that the system meets performance criteria.

1)Stress Testing: 2)Load Testing:

	Examples of load testing include: Downloading a series of large files from the internet. Running multiple applications on a computer or server simultaneously		
7	**Differentiate between Smoke Testing and Sanity Testing. (Any four points).**	**S-22**	**4**
Ans	<table><tr><th>Smoke Testing</th><th>Sanity Testing</th></tr><tr><td>Whenever a new feature comes into the picture, we perform Smoke Testing.</td><td>Whenever Code changes or Bug fixes comes into the picture, we perform Sanity Testing.</td></tr><tr><td>It is a part of Acceptance testing</td><td>It is a part of Regression testing.</td></tr><tr><td>It is shallow and wide.</td><td>It is deep and narrow.</td></tr><tr><td>Smoke testing is documented.</td><td>Sanity testing is not documented.</td></tr><tr><td>Smoke Testing is performed mostly in unstable builds.</td><td>Sanity Testing is performed in stable builds only.</td></tr><tr><td>Smoke Testing can be performed either manually or by using Automation tools.</td><td>Sanity Testing is commonly executed manually, not by using Automation approach.</td></tr></table>		
8	**Describe Unit Testing.**	**S-22**	**4**
Ans	Unit Testing: Software product is made up of many units, each unit needed to be tested to find whether they have implemented the design correctly or not. Additional Requirements: The module under consideration might be getting inputs from another module or the module is calling some another module. Some interface modules has to be simulated if required like drivers and stubs Drivers: The module where the required inputs for the module under test are simulated for the purpose of module or unit testing is known as a Driver module. The driver module may print or interpret the results produced by the module under test. Stubs: The module under testing may also call some other module which is not ready at the time of testing. There is need of dummy modules required to simulate for testing, instead of actual modules. These are called stubs. The objective of Unit Testing are follows To isolate a section of code. To verify the correctness of the code. To test every function and procedure. To fix bugs early in the development cycle and to save costs.		

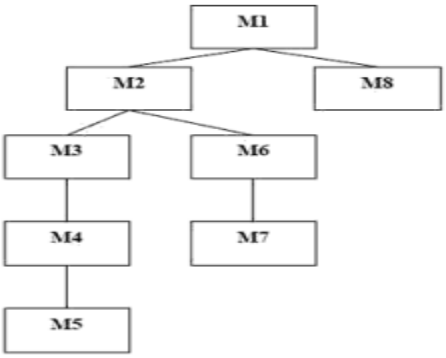
	To help the developers understand the code base and enable them to make changes quickly. To help with code reuse.		
9	**Explain following concepts related to security testing –** **i) Authorization. ii) Access Control.**	**S-22**	**4**
Ans	**Authorization –** ensuring sensitive systems or data properly control access for authenticated users according to their roles or permissions. Controlling user or client privileges or access levels to system resources, such as files, services, computer programs, data, and application features, authorization is a security mechanism. Authorization is used to restrict the user in accordance with the permissions they have been granted. The authorization process typically involves the usage of an access control list, user roles, user groups, and the definition of permissions and limits for a given user group as well as the granting and cancelling of user rights. **Access control** is a security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization. Access Control, also known as Authorization — is mediating access to resources on the basis of identity and is generally policy-driven (although the policy may be implicit). It is the primary security service that concerns most software, with most of the other security services supporting it. For example, access control decisions are generally enforced on the basis of a user-specific policy, and authentication is the way to establish the user in question. Similarly, confidentiality is really a manifestation of access control, specifically the ability to read data. Since, in computer security, confidentiality is often synonymous with encryption, it becomes a technique for enforcing an access-control policy.		
10	**With respect to client-server testing design test cases for Online Payment Transfer at banking system**	**S-22**	**6**

Ans

Test ID	Test case	Input	Expected output	Actual output	Status
TC1	Verify the bank website url	Click on url	Login page is opened	Login page is opened	pass
TC2	Verify the login page has username and password	---------	Username and password appeared	Username and password appeared	pass

		TC3	Verify the user if the user has access to the online money transfer feature	Click the online money transfer	The different payment options appeared	Different payment options appeared	pass		
		TC4	Check the fund transfer has 2 stp authentication	Click transfer button	Successfully transferred	Successfully transferred	pass		
		TC5	Check if the amount transferred notification is sent	-------	Notification is sent	Notification is sent	pass		
		TC6	Check the amount gets transfer to exact person		Got transferred to the appropriate person	Got transferred to the appropriate person	pass		
11	**With respect to GUI testing, write test cases for Flipkart login form**							**S-22**	**6**
		Test ID	Test case	Input	Expected Output	Actual output	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 button	It should redirect to password page	It redirects to password page	pass		
		TC4	Readability of font	---------	Content should be readable	Content is readable	pass		
		TC5	Testing of hyperlink	Move the mouse on hyperlink	It should redirect to respective page	It redirect To the specific page	pass		
12	**List the Level testing.**							**W-22**	**2**
Ans	Following are the levels of testing: a) Unit test b) Integration test c) System test d) Acceptance test								
13	**State the describe Top-down approach of integration testing with diagram.**							**W-22**	**4**
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.		
14	**Describe graphical user interface (GUI) testing and its important traits**	**W-22**	**4**
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.

15 **Design test cases for Web pages testing of any Web site (take a suitable example)** **W-22** **4**

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
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

16	**Write program for calculating even numbers from 1 to 20 And design test cases for the same**	**W-22**	**6**

Ans	Program :
	#include
	int main() {
	for(int i=0;i<=20;i=i+2)
	{
	printf("%d\n",i);
	}
	return 0;
	}		

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**	1-Jan-2023	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites :
1	C program using for loop

S #	Test Data

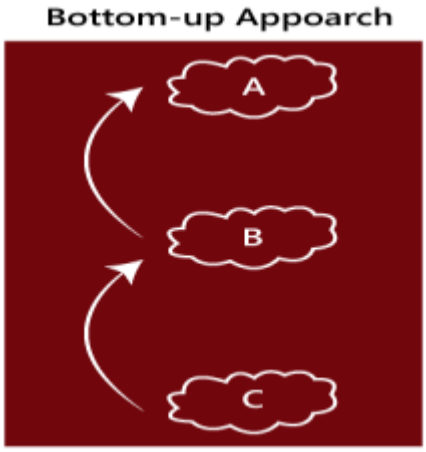
Test Scenario	Verify the even number

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

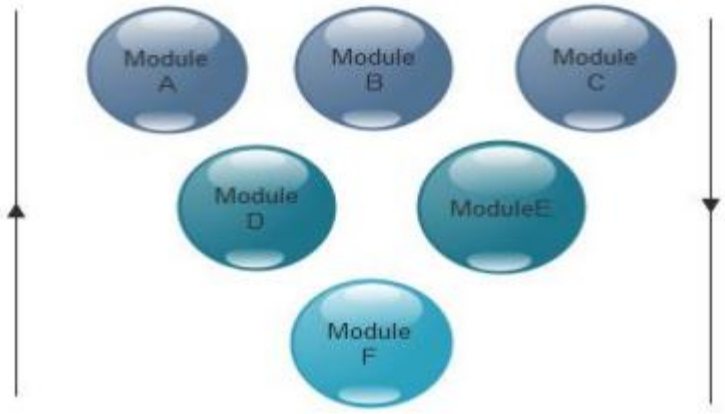
	3	Check the increment operator	Increment operator should increment by 2	Counter is incremented by 2	Pass	
	4	Check output	Even number is displayed on output screen	It is displaying even number	Pass	
17	**Enlist any two advantages of acceptance testing**					**S-23**	**2**
Ans	1) This testing helps the project team to know the further requirements from the users directly as it involve 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						
18	**Describe the concept of driver and stub with suitable example**					**S-23**	**4**
Ans	**Stubs**: Stubs are developed by software developers to use them in place of modules, if the respective modules aren't developed, missing in developing stage, or are unavailable currently while Top-down testing of modules. A Stub simulates module which has all the capabilities of the unavailable module. Stubs are used when the lower-level modules are needed but are unavailable currently. Stubs are divided into four basic categories based on what they do : -Shows the traced messages, -Shows the displayed message if any, -Returns the corresponding values that are utilized by modules, -Returns the value of the chosen parameters (arguments) that were used by the testing modules. **Drivers:** Drivers serve the same purpose as stubs, but drivers are used in Bottom-up integration sting and are also more complex than stubs. Drivers are also used when some modules are missing and unavailable at time of testing of a specific module because of some unavoidable reasons, to act in absence of required module. Drivers are used when highlevel modules are missing and can also be used when lower-level modules are missing. Ex: Suppose, you are told to test a website whose corresponding primary modules						

	are, where each of them is interdependent on each other, as follows: Module-A: Login page website, Module-B: Home page of the website Module-C: Profile setting Module-D: Sign-out page It's always considered good practice to begin development of all modules parallelly because as soon as each gets developed they can be integrated and could be tested further as per their corresponding interdependencies order with a module. But in some cases, if any one of them is in developing stage or not available in the testing process of a specific module, stubs or drivers could be used instead. Assume Module-A is developed. As soon as its developed, it undergoes testing, but it requires Module-B, which isn't developed yet. So in this case, we can use the Stubs or Drivers that simulate all features and functionality that might be shown by actual Module-B. So, we can conclude that Stubs and drivers are used to fulfil the necessity of unavailable modules. Similarly, we may also use Stubs or Drivers in place of Module-C and Module-D if they are too not available. Do both drivers and Stubs serve the same functionality? Yes, we can say both serve the same feature and are used in the absence of a module(M1) that has interdependencies with another module(M2) that is need to be test, so we use drivers or stubs in order to fulfil module(M1)"s unavailability's and to serve its functionality		
19	**Describe the types of integration testing with neat diagram.**	**S-23**	**4**
Ans	1. Top down Integration The top-down testing strategy deals with the process in which higher level modules are tested with lower level modules until the successful completion of testing of all the modules. Major design flaws can be detected and fixed early because critical modules tested first. In this type of method, we will add the modules incrementally or one by one and check the data flow in the same order.		

2. Bottom up Integration The bottom to up testing strategy deals with the process in which lower level modules are tested with higher level modules until the successful completion of testing of all the modules. Top level critical modules are tested at last, so it may cause a defect. Or we can say that we will be adding the modules from bottom to the top and check the data flow in the same order

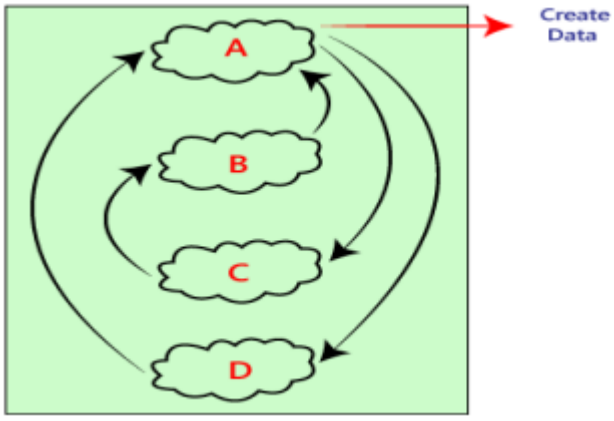


3. Hybrid testing Integration In this approach, both Top-Down and Bottom-Up approaches are combined for testing. In this process, top-level modules are tested with lower level modules and lower level modules tested with high-level modules simultaneously. There is less possibility of occurrence of defect because each module interface is tested.



Hybrid Method

4. Bing Bang Integration In this approach, testing is done via integration of all modules at once. It is convenient for small software systems, if used for large software systems identification of defects is difficult. Since this testing can be done after completion of all modules due to that testing team has less time for execution of this process so that internally linked interfaces and high-risk critical modules can be missed easily.



5. _____

20	**Explain the testing approaches that are considered during web application testing**	**S-23**	**4**
Ans	**Functional Testing:** Unit Testing: Testing individual units or components of the application to verify they work correctly. Integration Testing: Testing interfaces between components to ensure they work together as expected. System Testing: Testing the entire system as a whole to ensure all components work together in the actual environment. **Non-Functional Testing:** Performance Testing: Testing how the application performs under various conditions such as load, stress, and scalability.		

	Security Testing: Testing to identify vulnerabilities and ensure the application is secure against unauthorized access and attacks. Usability Testing: Testing to evaluate how user-friendly and intuitive the application is for end-users. Compatibility Testing: Testing to ensure the application works across different browsers, devices, and operating systems. Automated Testing: Regression Testing: Running tests to ensure that recent changes haven't adversely affected existing functionality. Functional Testing: Automating tests for repetitive tasks such as form submission, user login, etc., to increase efficiency and reduce manual effort. User Interface (UI) Testing: UI Testing: Testing the graphical interface to ensure it meets design specifications and functions correctly for users. Database Testing: Database Testing: Testing to ensure data integrity, data validity, and database performance. Accessibility Testing: Accessibility Testing: Testing to ensure that the application can be used by people with disabilities and complies with accessibility standards		
21	**Describe the regression testing. State entry and exit criteria for regression testing.**	**S-23**	**4**
Ans	**Regression Testing:** Definition: Regression testing is the practice of re-executing previously conducted tests on an application after changes have been made to ensure that existing functionalities continue to work correctly. Purpose: The primary goal of regression testing is to uncover any defects introduced or re-introduced due to changes in the software or its environment. Key Points: Automation: Regression tests are often automated to ensure efficiency and consistency in testing repetitive scenarios. Scope: It focuses on testing impacted areas of the software to verify that new changes have not adversely affected existing features. Continuous Process: Regression testing is typically an ongoing process throughout the **software development lifecycle (SDLC).** **Entry and Exit Criteria:** Entry Criteria: Definition: Entry criteria specify the conditions that must be met before testing can commence for a specific phase of the SDLC or a testing cycle. Purpose: They ensure that the project is sufficiently prepared for testing activities, reducing the likelihood of wasted effort and resources. Examples: Entry criteria may include completion of development, availability of necessary test resources (like test data and environments), and adherence to coding		

standards.

Exit Criteria:

Definition: Exit criteria define the conditions that must be satisfied before a testing phase or activity can be considered complete and exited.

Purpose: They help assess whether the testing phase has achieved its goals and whether the software is ready to progress to the next phase or release.

Examples: Exit criteria may include achieving a certain level of test coverage, resolving critical defects, and obtaining approval from stakeholders.

Relationship: Regression testing often occurs multiple times throughout the SDLC, with entry criteria ensuring the environment and prerequisites are met before each regression testing cycle begins. Exit criteria for regression testing ensures that the tests have adequately verified the impact of changes and that the software can proceed confidently to the next stage or release.

22 **Write test cases for facebook login form with respect to graphical user interface (GUI) testing.** **S-23** **6**

Ans

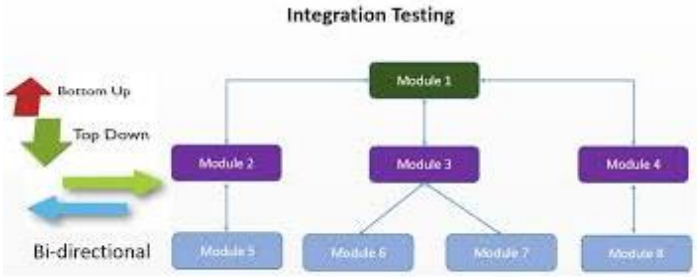
Test ID	Test case	Input	Expected output	Actual output	Status
TC1	Verify the login with correct username	Enter correct password	Login successfully	Login successfully	pass

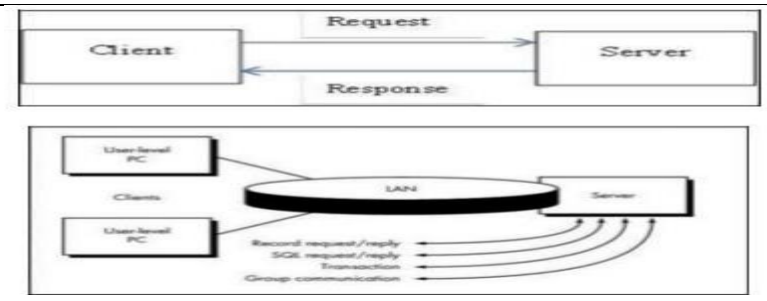
		and password						
	TC2	Verify login with correct username but incorrect password	Enter correct username and incorrect password	Incorrect password Enter again	Incorrect password Enter again	pass		
	TC3	Verify login with incorrect username and password	Enter incorrect username and password	User doesn't exist	User doesn't exist	pass		
	TC4	Check if the username and password appeared properly in login form		Username and password Displayed	Username and password Displayed	pass		
	TC5	Verify if the password is empty	Don't enter password	Please enter password	Please enter password	pass		
	TC6	Verify the username isn't filled	Don't enter username	Username is not entered please enter	Username is not entered please enter	pass		
23	**Design test cases for online railway reservation applications**						**S-23**	**6**

Ans

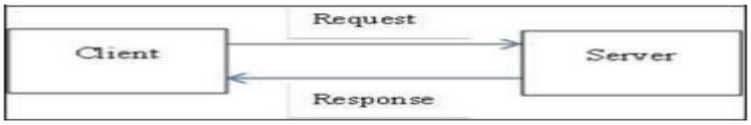
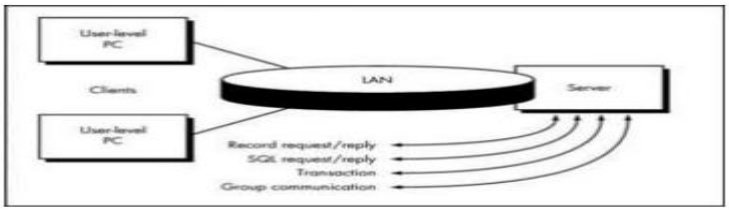
Test ID	Test case	Input	Expected output	Actual output	Status
TC1	Verify the login with correct username and password	Enter correct username and password	Login successfully	Login successfully	pass
TC2	Verify the login with incorrect username and password	Enter incorrect username and password	User doesn't exist	User doesn't	pass
TC3	Verify the tickets book option	Enter no. of tickets	Tickets booked	Tickets booked	pass
TC4	Verify the payment option	Click the payment	Payment successfully	Payment successfully	pass
TC5	Verify search train option	Search for trains	Trains details appeared	Trains details appeared	pass
TC6	Verify the passenger details	Click passenger details	Passenger details appeared	Passenger Details appeared	pass

24	**What is GUI testing? Give one example.**	**W-23**	**2**
Ans	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. Example: - : Consider any website like MSBTE, Google, yahoo or any login form or GUI of any application to be tested.		
25	**Define Driver and Stub.**	**W-23**	**2**
Ans	**Driver:** Drivers are used in bottom-up integration testing approach. It can simulate the behaviour of upper-level module that is not integrated yet. Driver's modules act as the temporary replacement of module and act as the actual products. Drivers are also used for interact with external system and usually complex than stubs. Driver: Calls the Module to be tested. **Stub:** Stubs are used in top down integration testing. It can simulate the behaviour of lower-level module that is not integrated. They are act as a temporary replacement of module and provide same output as actual product. When needs to intact with external system then also stubs are used. Stub: Is called by the Module under Test.		

26	**Illustrate process of bi-directional integration testing. State its two advantages & disadvantages.**	**W-23**	**4**
Ans	• Bi-directional integration is a combination of the top down and bottom up integration testing approaches used together to derive the integration process. • It is also called as **sandwich integration**. • **Process** of bi-directional testing is as follows - o In this type of testing, the testing begins from the middle layer. o When the bottom up approach is used then testing is done from middle layer to the top layer. o When the top down approach is used then testing is done from middle layer to bottom layer.  **Advantages:** 1. Sandwich approach is useful for very large projects having several subprojects. 2. Both Top-down and Bottom-up approach starts at a time as per development schedule. **Disadvantages:** 1. It require very high cost for testing because one part has Top-down approach while another part has bottom-up approach. 2. It cannot be used for smaller system with huge interdependence between different modules. It makes sense when the individual subsystem is as good as complete system.		
27	**Describe with one example each** **load testing**	**W-23**	**6**

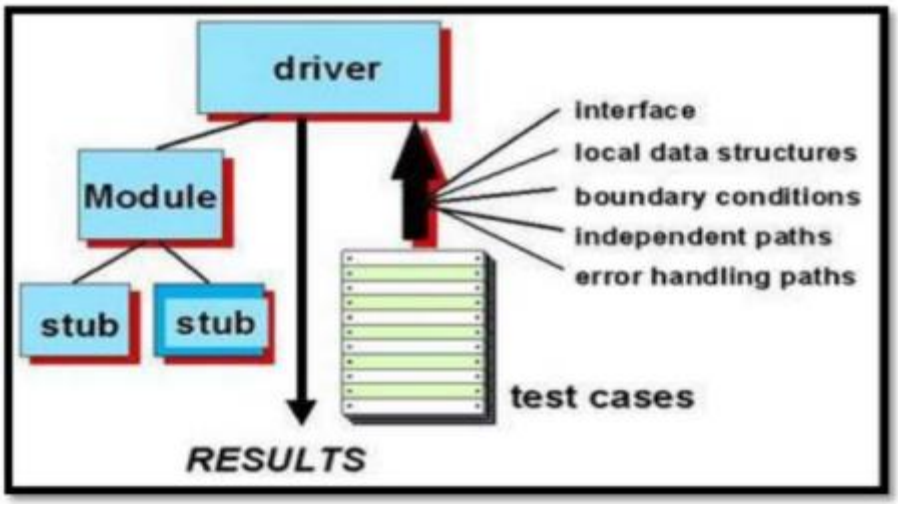
	Stress testing		
Ans	**Stress testing: -** is testing the software under less than ideal conditions. So subject your software to low memory, low disk space, slow cpus, and slow modems and so on. Look at your software and determine what external resources and dependencies it has. Stress testing is simply limiting them to 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 had a greater potential to expect a bug. Setting the values to zero or near zero will make the software execute different path as it attempt to handle the tight constraint. Ideally the software would run without crashing or losing data. **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 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's should be able to run forever without being restarted. So Time acts as a important variable. Stress testing and load testing can be best applied with the help of automation tools. Stress testing and load testing are the types of performance testing. The Microsoft stress utility program allows you to individually set the amounts of memory, disk space, files and other resources available to the software running on the machine. Example: Open many number of browsers in the windows simultaneously. Connect more than the specifies clients to the server. Connect more than one printer to the system.		
28	**Explain client-server testing with suitable diagram**	**Winter-23**	**6**
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. **Testing approaches of client server system** **1. 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. 3. 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. **4. 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. **5. 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 connecting them. The requirement specifications must describe the possible expectations in case of any failure. **6. 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. **7. 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.		
29	**Describe Acceptance testing with its advantages.**	**S-24**	**4**

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.		
30	**With the help of diagram, Describe client server testing.**	**S-24**	**6**
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.		

	Testing approaches of client server system **1. 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. **2. Integration testing:** After successful testing of server, client and network, they are brought together to form system testing. 3. 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. **4. 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. **5. 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 connecting them. The requirement specifications must describe the possible expectations in case of any failure. **6. 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. **7. 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.		
31	**Explain the need of stubs and drivers with diagram and its importance in software testing.**	**S-24**	**6**
Ans	**Drivers:** The module where the required inputs for the module under test are simulated for the purpose of module or unit testing is known as a Driver module. The driver module may print or interpret the result produced by the module under test. **Stubs:** The module under testing may also call some other module which is not		

ready at the time of testing. There is need of dummy modules required to simulate for testing, instead of actual modules. These are called stubs.



For example if a developer is developing a loop for searching functionality of an application which is a very small unit of the whole code of that application then to verify that the particular loop is working properly or not is known as unit testing.

Importance:

Stubs and Drivers works as a substitute for the missing or unavailable module. They are specifically developed, for each module, having different functionalities. Generally, developers and unit testers are involved in the development of stubs and drivers. 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.

JSPM's

Rajarshi Shahu College of Engineering, Polytechnic Tathawade

Department of Computer Engineering

Academic Year: 2024-2025

Course Name: Software Testing **Course code:** 22518

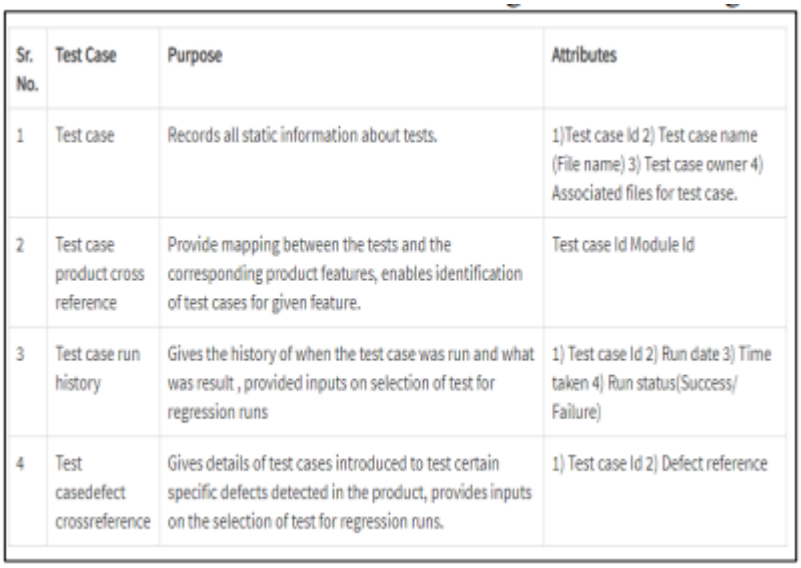
Semester: Odd **Name of the Faculty:** Prof. Suvarna A. Amle

UNIT NO. 3 Test Management

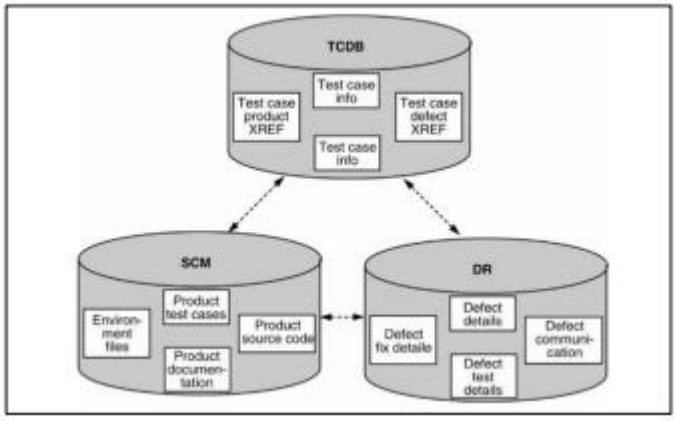
CO3: Prepare test plan for an application.

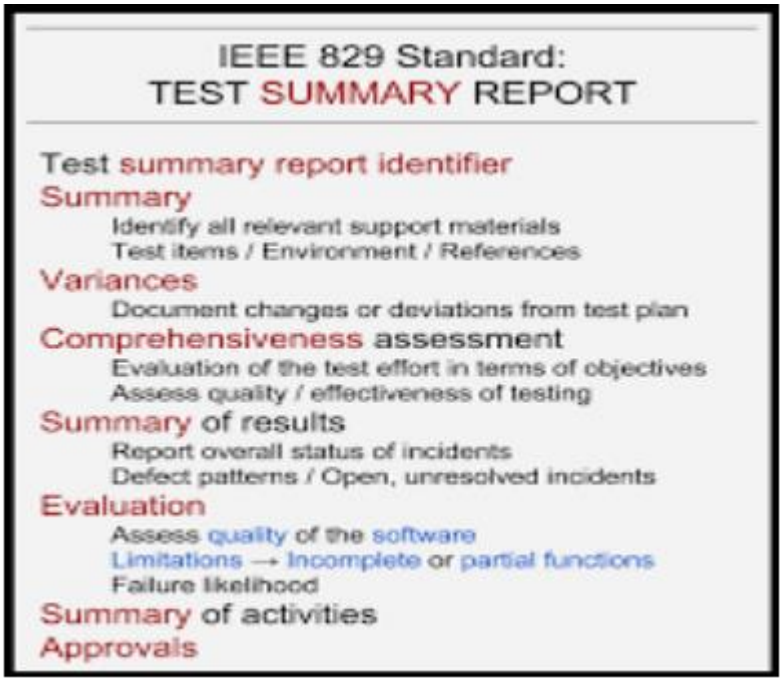
Q. No.	Questions	Year	Marks
1	**Enlist any two activities involved in test planning**	**W-19**	**2**
Ans	1. Scope Management: Deciding what features to be tested and not to be tested. 2. Deciding Test approach /strategy: Which type of testing shall be done like configuration, integration, localization 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.		
2	**State the contents of 'Test Summary Reports' used in test reporting.**	**W-19, W-22**	**4**
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 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. 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, 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.		
3	**Describe standards included in Test Management.**	**W-19**	**4**
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. 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. 6. 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 reside. • International Standard: are defined at international level and these are applicable to all customers across the globe.		
4	**Describe test infrastructure management.**	**W-19**	**4**
Ans	Test infrastructure management Testing requires a robust infrastructure to be planned upfront. This infrastructure is made up of three essential elements. 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 following table  A test case database captures all the relevant information about the test cases in an		

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

	organization. Some of the entities and attributes in each of the entities in such a TCDB are: • Test case • Test case-product cross reference • Test case run history **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 **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.  		
5	**Describe the process of preparing summary report in test planning**	**W-19**	**4**

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.
6	**Design test cases for railway reservation system.**	**W-19**	**6**
Ans	Design test cases for railway reservation system.		

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

7	**Design test cases for hostel admission form of your institute**	**Winter-19**	**6**

Ans

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

8	**Design a test plan along with the test cases for edit function in notepad**	**W-19**	**6**

Ans

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

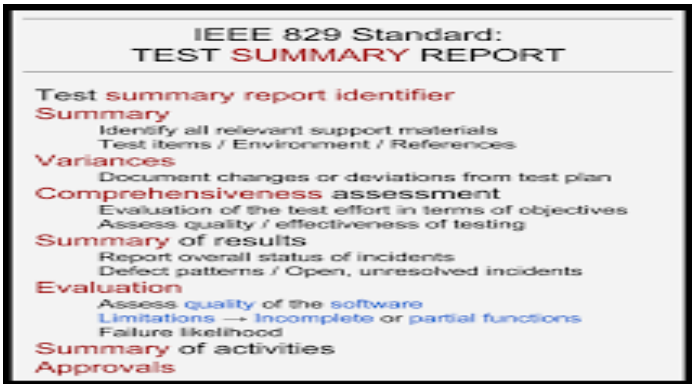
Test engineer 1

Test engineer 2

Test engineer 3

	Test engineer 4		
9	**Enlist different types of test deliverables.**	**S-22**	**2**
Ans	1. Test plan 2. Test cases 3. Test data 4. Test summary report 5. Traceability metrics		
10	**Enlist any four skills for software tester**	**S-22**	**2**
Ans	1. Communication skills 2. Project management skills 3. Adaptability 4. Programming 5. Knowledge of Test Management tools 6. Analytical skills		
11	**Explain test management with Test Infrastructure Management and Test People Management.**	**S-22**	**4**
Ans	**Test Management Process:** It is a software process that manages the start to the end of all software testing activities. This management process provides planning, controlling, tracking, and monitoring facilities throughout the whole group cycle, this process includes several activities like test case design and test execution, test planning, etc. It also gives an initial plan and discipline specifications for the software testing process. **Responsibilities:** Works in collaboration with test analyst and technical test analyst to select and customize the appropriate templates and also establish standards. Provides all facilities to keep track and control the testing throughout the project. Gives a clear concept of understanding the testing activity of the prior upcoming project and also posts one The test management process has two main parts of test Management Process: Planning: Risk analysis Test Estimation Test planning **Test infrastructure management**		

	Testing requires a robust infrastructure to be planned upfront. This infrastructure is made up of three essential elements. **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 following table A test case database captures all the relevant information about the test cases in an organization. Some of the entities and attributes in each of the entities in such a TCDB are: □ Test case □ Test case-product cross reference □ Test case run history □ Test case- defect cross reference		
12	**Explain test reporting with executing test cases and preparing test summary report**	**S-22**	**4**
Ans	**Test reporting** is a crucial process in software testing that revolves around gathering, analysing, and presenting essential test data and results to stakeholders. At its core, it serves as a vital communication channel, providing insights into a software application's progress, quality, and readiness throughout the testing lifecycle. By consolidating test outcomes, test reporting empowers teams to identify patterns, make informed decisions based on data, and proactively tackle potential issues. The fundamental purpose of test reporting is to keep all project stakeholders informed and aligned. This includes developers, testers, product owners, project managers, and key team members. By sharing a comprehensive view of the software's testing status, test management reporting fosters collaboration and makes sure everyone involved is on the same page. This level of transparency is essential in complex development environments, as it promotes a shared understanding of the software's health and assists in streamlining the development process. Test Execution is the process of executing the tests written by the tester to check whether the developed code or functions or modules are providing the expected result as per the client requirement or business requirement. Test Execution comes under one of the phases of the Software Testing Life Cycle (STLC). In the test execution process, the tester will usually write or execute a certain number of test cases, and test scripts or do automated testing. If it creates any errors then it will be informed to the respective development team to correct the issues in the code. If the text execution process shows successful results, then it will be ready for the deployment		

	phase after the proper setup for the deployment environment. **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 		
13	**Explain test case specification.**	**S-22**	**4**
Ans	A test case specification in software testing is a critical document that outlines the precise steps and conditions for testing a specific aspect of a software application. It acts as a detailed blueprint, outlining precisely what scenarios need testing, how to conduct those tests, and how frequently they should be performed for a specific software feature. Each test case within the specification is uniquely identified, described, and often linked to preconditions and dependencies. Test runners rely on these specifications to		

determine which test suites to execute, making them crucial instructions for testing teams. By following these meticulously crafted instructions, software testers can systematically evaluate software functionality.

Test case specifications not only enhance the quality of testing but also provide a clear reference for developers when addressing identified issues, ultimately contributing to the overall reliability and robustness of the software.

A test specification is important for several reasons:

Test Specifications provide clear and precise instructions on what scenarios to test, how to test them, and what the expected outcomes should be. This clarity ensures that testers have a comprehensive understanding of their testing objectives, reducing ambiguity and potential errors in the testing process.

A well-documented specification can be reused for regression testing, future releases, or similar projects. They serve as a valuable reference point, helping testing teams maintain consistency in testing procedures over time and across different testing cycles.

When a test case fails, it acts as a crucial reference for identifying the root cause of the failure. By comparing the expected outcomes in the specification with the actual results, testers can pinpoint the issues and facilitate efficient debugging and issue resolution.

It provides a common language and framework for collaboration between QA teams, developers, and other stakeholders.

14 **Design test cases for Online Mobile Recharge (Data filed are mobile number, state, email-id, recharge amount.)** **S-22** **6**

Ans

Test case ID	Test case	Input	Expected output	Actual output	status
TC1	Check valid number	Enter mobile number	Number is valid	Number is valid	pass
TC2	Choose correct plan	Enter correct recharge plan	Recharge plan is valid	Recharge plan is valid	pass
TC3	Check valid email id	Enter email id	Email id is valid	Email id is invalid	fail
TC4	Check invalid number	Enter incorrect number	Number is not exist	Number is not exist	pass
TC5	Check payment	Enter amount	Recharge successfu	Recharge successfu	pass

			option		l	l			
		TC6	Choose incorrect plan	Enter incorrect plan	Plan doesn't exist	Plan doesn't exist	pass		
15	**Prepare a test plan along with test cases for creating a saving account at bank. (Test cases should be at least six)**							**S-22**	**6**
Ans		Test ID	Test cases	Input	Expected output	Actual output	Status		
		TC1	Verify that registered user can login with correct name and password	Enter correct username and password	Login successfully	Login successfully	pass		
		TC2	Verify login with incorrect username and password	Enter incorrect username and password	User doesn't exist	User doesn't exist	pass		
		TC3	Verify the user registration form accessible and display correctly in various browser.	Enter website in different browser	All controls appeared properly	All controls appeared properly	pass		
		TC4	Verify that user registered	Click register button	Registered successfully	Registered successfully	pass		
		TC5	Confirms that the applicatio		Account is saving	Account is saving	pas		

			n identifies and categorizes the account type i.e saving						
		TC6	Verify the user can access money transfer	Click transfer	Successfully transeferred	Successfully transeferred	pass		
16	**State any four needs to prepare a test plan.**							**W-22**	**2**
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.								
17	**Describe the 'Test Infrastructure' components with diagram**							**W-22** **S-23**	**4**
Ans	Testing requires a robust infrastructure to be planned upfront. This infrastructure is made up of three essential elements  Fig. Component 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.

2. 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

18	**Write the test cases for Notepad application. (any eight test case)**	**W-22**	**4**

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
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	If it is matching, then it highlights it.	Pass

19	**Design test cases for Web pages testing of any Web site (take a suitable example)**	**W-22**	**4**

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
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

20	**Define test plan with two advantages**	**S-23**	**2**
Ans	**Test plan: -** A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test		

	environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process. Improved project visibility: A well-defined test plan provides clear visibility into the testing process, objectives, and expectations. Better risk management: The test plan helps identify potential risks and plan for their mitigation, reducing the risk of project failure.		
21	**Write steps to prepare test cases**	**S-23**	**2**
	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.		
22	**Describe any four skills of software tester**	**S-23**	**4**
	Following are the various skill required to become a good software tester: · Communication Skills: Testers are expected to be good listeners as well as good presenters. A good software tester must have strong verbal and written communication skills. They require a good communication with developers before, during and after the project. The test reports, plans/cases which testers made should be easy to read and comprehend. They must be good listeners, good speakers, good writers, and good readers etc. all at the same time. Communication skills of a good software tester include his/her body language, their words, tone, writing styles, listening and attending others etc. **Domain Knowledge:** Testers should have the detailed knowledge about the software or application, whether they are not domain experts and this knowledge will help them to find such errors which a user can face while using the application. After testing the application/software, the tester should keep the domain in his/her mind while arranging the errors in order according to their priority. As the testers may be working in different domain, on different technologies so they should be aware of any challenges and complexities. And the tester which has this domain knowledge quality may also include a good user interface, able to differentiate between a trivial and critical issue and also the better understanding of issues as they are some pros of Domain Knowledge.		

Desire to Learn:

Testers should have the brief knowledge regarding the latest technologies, tools & techniques and they can also use them during testing. As there are various tools and techniques for development as well as testing & every tool or technique has some positive and negative effects. Testers must be able to learn new technologies & can also use them while testing. Working with latest tools/techniques might be difficult for tester but they can get something new.

Technical Skills:

A good software tester must have strong Technical skills. They must have proper knowledge about the coding skills in order to understand the application, good communication with developers and write test automation. The technical skills also include high proficiency in tools like MS Office, testing tools etc. These skills can be obtained by practicing and proper training.

Analytical Skill:

A good software tester should have to be able to check out how to reproduce the errors because only finding errors are not sufficient. For better understanding and creating of good test reports, analytical skill will break the complex software system into smaller units. Testers should have to analyze the situation of user while using the software or application. Testing report is a SWOT i.e. 'Strength', 'Weakness', 'Opportunity' and 'Threat' analysis of Software.

Planning: First of all testers should have to plan how to make the testing report. Testing should be in proper manner i.e. it should cover all the functionalists, requirements, features and critical aspects of business etc. The testing report should be made in an exact order i.e. according to the priority of the errors/bugs. For better judgment of testing report good planning is very necessary.

Integrity: Testers find the errors in applications/software with an assurance that developers will absolutely fix them. The testing report should have to show the priorities of the errors i.e. the report should be in various levels according to their priority level.

Curiosity: During testing or analyzing any software, the testers must know about various applications, various domains etc. As domain has its own specialty so testers must have the curiosity to understand the domain under testing. They should have an eagerness of understanding the complexity and expectations. · Think from Users Perspective: Each and every product is developed and designed for customers. Customers may not be having all technical skills a tester is having, if you fail to keep this in mind you might miss important bugs.

Be a Good Judge of Your Product: Last but not the least; you have to be a good judge of your product. Ask yourself questions whether the software meeting all the requirements it should be having. As now you are the judge and you have the powers to distinguish between right and wrong. Judge listens to all testers in team.

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>23</td><td>Describe the factors considered to decide test approach.</td><td>S-23</td><td>4</td></tr>
<tr><td></td><td>1. Project Scope and Objectives: Understanding the project's goals, functionalities, and intended users helps in determining the appropriate testing depth and coverage.
2. Risk Assessment: Identify and prioritize the risks associated with the project. High-risk areas may require more rigorous testing and validation.
3. Timeline and Schedule: Consider the project timeline and release schedules. Test approaches should align with development milestones to ensure timely feedback and bug fixes.
4. Budget and Resources: Determine the available budget, human resources, tools, and infrastructure for testing. This influences the feasibility of different testing methods.
5. Quality Standards and Compliance: Adherence to industry standards, regulatory requirements, and quality benchmarks (e.g., ISO standards, GDPR compliance) may dictate specific testing methodologies.
6. Technology and Architecture: The technology stack, system architecture, and integration points impact the testing strategy. For instance, distributed systems may require testing for scalability and performance.
7. Development Methodology: Whether using Agile, Waterfall, DevOps, or another methodology affects when and how testing is integrated into the development lifecycle.
8. Types of Testing Needed: Identify the types of testing required (e.g., functional, non-functional, regression, performance, security) based on project requirements and risks.
9. Feedback Loop: Consider how feedback from testing will be incorporated into development. Continuous integration and continuous testing practices may be necessary for agile projects.
10. Customer or User Expectations: Understand user expectations and acceptance criteria to ensure the testing approach aligns with delivering a satisfactory product.</td><td></td><td></td></tr>
<tr><td>24</td><td>Design test cases for hostel admission form.</td><td>S-23</td><td>6</td></tr>
<tr><td>Ans</td><td>
<table>
<tr><td>Test ID</td><td>Test Case</td><td>Input</td><td>Expected output</td><td>Actual output</td><td>Status</td></tr>
<tr><td>TC1</td><td>Verify the all Information displays properly</td><td></td><td>All information appeared</td><td>All information appeared</td><td>pass</td></tr>
</table>
</td><td></td><td></td></tr>
</table>

	TC2	Verify the hostel website properly opened in different browser	Enter url In different browsers	Site opened with login page	Site opened with login page	pass		
	TC3	Check if the date is empty	Don't enter date	Fill the date message appeared	Fill the date message appeared	pass		
	TC4	Check if the income is empty	Don't enter father's income	Fill all the income value appeared	Fill all the income value appeared	pass		
	TC5	Verify the registration button	Click the registration button	Successfully registered	Successfully registered	pass		
	TC6	Verify login option with correct username and password	Click the login button	Successfully login	Successfully login	pass		
25	**Prepare test plan for Notepad application. (Windows based)**						**W-23**	**4**
Ans	**Test Plan** **Identifier TP_10** **Introduction:** The purpose of this document is to create an application test plan for 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

26 **What is test plan? What is its need? List test planning activities.** **W-23** **4**

Test plan: - A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

Need of test plan:

Test Plan functions as a roadmap for carrying out test cases, monitoring test advancement, and guaranteeing thorough coverage. Product Managers, who use test plans to make informed decisions about release timelines, resource allocation, and overall product quality.

List test planning activities:-

Analyze the product (learn product thoroughly)

- Develop test strategy -define scope of testing ,risk and issues
- Define objective of test
- Define test criteria
- Planning the resources
- Plan test environment
- Schedule and cost
- Test deliverables

27 **Prepare six test cases for marketing site www.flipkart.com.** **W-23** **6**

Sr. No	Test Case - ID	Test case Objective	Prerequisite	Steps	Input data	Expected Result	Actual Result	Remarks/ Status
1	TC-1	To check interest	Whether internet is available or not?	Test internet connection	w.w w. flipkart .com	Site home page display should	Home page displayed	Test to fail

						display on screen		
2	TC-2	User name	correct and valid user name should be registered on flipkart	Type correct and valid user name	User name given by flipkart	Should enter valid the user name	Goes to next page if user id and pass word verified and validated	Test to pas s
3	TC-3	Password	correct and valid password should be validate and verified on flipkart site	Type correct and valid password	Password selected by user and validated by flipkart	Should enter valid password	Goes to next page if user id and pass word verified and validated	Test to pass
4	TC-4	To check whether site home page opens or not	To display flipkart home page website on screen	Type site proper address as www.flipkart .com	w.w w. flipkart .com	Site home page display should display on screen	Home page displayed	Test to fail
5	TC-5	TC5 To search the product	Search the product	Click on product link	Mouse rollover and click	Shall display availability of product	Show availability of product	Test to pass
6	TC-6	Product details	Product shall be available on site	Click on product link	Mouse rollover and click	Shall display product details	displays product details of other produc	Test to fail

							t	
7	TC7	Product details	Product shall be available on site	Click on product link	Mouse rollover and click	Shall display product details	displays product details of product	Test to pass
8	TC8	Search for offers	Offer should be valid for that day	Click on Offer link	Select Offer icon and click	Shall display offer details	displays offer details of selected product	Test to pass

28	**Write important six test cases for the 'Login Form' of the Facebook website.**	**W-23**	**6**

Test cases for facebook login are as given below:

Step	Test step	Test data	Expected output	Actual output	Status
1	Navigate to login page of facebook	---	---	---	Pass
2	Provide valid username	Username abc@yahoo.co.in	Shall accept the username	Accepted user name	Pass
3	Provide password	Password co6g1234	Shall accept the password	Accepted the password	Pass
4	Click on submit	Press submit button	User should be able to login successfully	User successfully logged in	Pass
5	Go to the home page	Click on home button	Should display home page	Home page of the user displayed	Pass
6	Write the status	Type in the status in the area provided and press post	Should post the message typed in and the status of user should change	Status changed successfully	Pass

JSPM's

Rajarshi Shahu College of Engineering,
Polytechnic Tathawade

Academic Year: 2024-2025

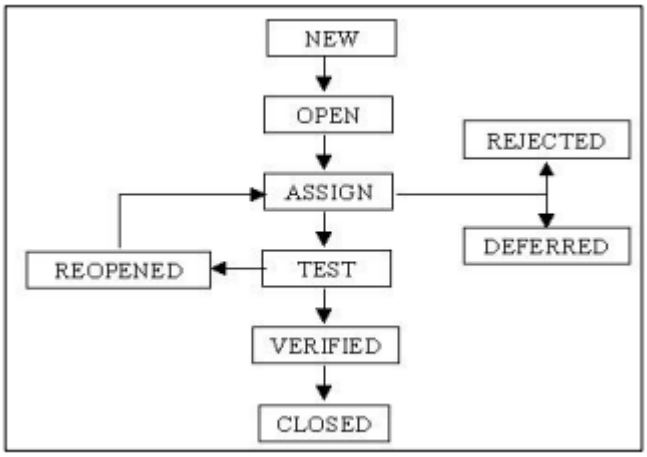
Course Name: Software Testing **Course code:** 22518

Semester: Odd **Name of the Faculty:** Prof. Suvarna A. Amle

UNIT NO. 4 **Defect Management**

CO4: Identify bugs to create defect report of given application.

Q. No.	Questions	Year	Marks
1	**Define Defect:**	**W-19**	**2**
Ans	A defect is an error in coding that causes a program to fail or to produce incorrect /unexpected results.		
2	**Enlist different techniques for finding defects and describe any one technique with an example.**	**W-19**	**4**
Ans	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		

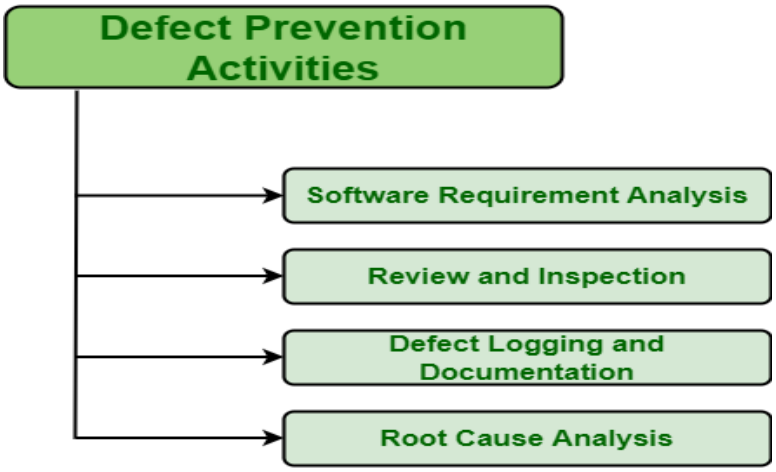
	execution of work products to evaluate it with expected behaviour. It includes black box testing methodology such as system testing and unit testing. The testing methods evaluate the product with respect to requirements defined; designs created and mark it as pass or fail. **3.Operational techniques:** 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.		
3	**Draw a diagram for defect life cycle and write example for defect template**	**W-19**	**6**
Ans	DEFECT LIFE CYCLE (Bug Life cycle) is the journey of a defect from its identification to its closure. The Life Cycle varies from organization to organization and is governed by the software testing process the organization or project follows and/or the Defect tracking tool being used.  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.
Defect Priority	Priority of the Defect.
Reported By	The name of the person who reported the defect.
Assigned To	The name of the person that is assigned to analyze/fix the defect.
Status	The status of the defect.
Fixed Build Version	Build version of the product where the defect was fixed (e.g. 1.2.3.9)

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		
4	**State any four defect reporting guidelines.**		**S-22**	**2**
Ans	1) Defect report forms the base quality measurement. 2) Correct the defect. 3) Report status of system. 4) Gather Statistics to predict Failure.			
5	**Draw defect prevention process cycle. State working of each phase.**		**S-22**	**4**
Ans	Defect prevention is basically defined as a measure to ensure that defects being detected so far, should not appear or occur again. For facilitating communication simply among members of team, planning and devising defect prevention guidelines, etc., Coordinator is mainly responsible. Coordinator is mainly responsible to lead defect prevention efforts, to facilitate meetings, to facilitate communication between team members and management, etc. DP board generally has a quarterly plan in which sets some goals at organization level. To achieve these goals, various methods or activities are mostly used and carried out to achieve and complete these goals.			



1. Software Requirement Analysis : The main cause of defects in software products is due to error in software requirements and designs. Software requirements and design both are important, and should be analyzed in an efficient way with more focus. Software requirement is basically considered an integral part of Software Development Life Cycle (SDLC). These are the requirements that basically describes features and functionalities of target product and also conveys expectations or requirement of users from software product. Therefore, it is very much needed to understand about software requirements more carefully, If requirements are not understood well by tester and developers, then there might be chance of occurring of issue or defect in further process. Therefore, it is essential to analyze and evaluate requirements in more appropriate and proper manner.

2. Review and Inspection : Review and inspection, both are essential and integral part of software development. They are considered as powerful tools that can be used to identify and remove defects if present before their occurrence and impact on production. Review and inspection come in different levels or stages of defect prevention to meet different needs. They are used in all software development and maintenance methods. There are two types of review i.e., self-review and peer-review.

3. Defect Logging and Documentation : After successful analysis and review, there should be records maintained about defects to simply complete description of defect. This record can be further used to have better understanding of defects. After getting knowledge and understanding of defect, then only one can take some effective and required measures and actions to resolve particular defects so that defect cannot be carried further to next phase.
4. Root Cause Analysis : Root because analysis is basically analysis of main cause of defect. It simply analysis what triggered defect to occur. After analyzing main cause of defect, one can find best way to simply avoid occurrence of such types of defects next time.

5. Static Code Analysis: To find any problems in the source code without running the program, use automated techniques for static code analysis. Before the code is even compiled, these tools can detect typical programming errors, coding standard violations and other problems.
6. Pair programming: It involves two programmers sharing a workstation, where one writes code while the other goes over each line as it is written. This real-time communication between team members promotes constant feedback and helps in the early detection of mistakes.
7. Test-Driven Development (TDD): Prior to writing any code, write automated tests. This method helps identify errors early in the development process and guarantees that the code complies with the requirements.
8. Training and Skill Development: Investing in continuous training and skill development for team members is recommended. Skilled developers are better able to follow best practices and are less prone to make frequent mistakes.
9. Checklists: To make sure that crucial actions are not missed during the various stages of development, use checklists. Developers and reviewers can quickly confirm that established standards are being followed by using checklists as a reference.

6 **Design any four test cases for withdraw an amount from ATM and prepare defeat report of it.** **S-22** **6**

Ans

Test case id	Test case	Input	Expected output	Actual output	Status
TC1	Check valid card	Insert valid card	Card inserted	Card inserted	pass
TC2	Check invalid card	Insert invalid card	Invalid card insert again	Invalid card insert again	pass
TC3	Check for valid amount	Enter amount to withdraw	Amount withdrew	Amount withdrew	pass
TC4	check pin option	Enter valid pin	Pin is correct	Incorrect pin	fail
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.	
7	**Give the defect classification and its meaning.**		**W-22**	**2**
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			

<table>
<tr><td></td><td>future.

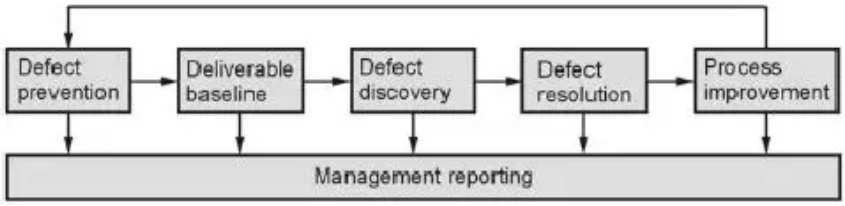
Testing Defects: These would encompass incorrect, incomplete, missing inappropriate test cases and test procedures.</td><td></td><td></td></tr>
<tr><td>8</td><td>Prepare defect report after executing test cases for any login form.</td><td>W-22</td><td>4</td></tr>
<tr><td>Ans</td><td>Defect Report in Software Testing is a detailed document about bugs found in the software application Following is Defect report after executing test cases for Email-log in form.

<table>
<tr><td>ID number</td><td>#123</td></tr>
<tr><td>Name</td><td>loginform - Unable to login Email</td></tr>
<tr><td>Reporter</td><td>Person’s name (xyz)</td></tr>
<tr><td>Submit Date</td><td>03/01/2023</td></tr>
<tr><td>Summary</td><td>When I put my mail id and password, I am unable to login while login credentials are right.</td></tr>
<tr><td>URL</td><td>www.gmail.com</td></tr>
<tr><td>Screenshot</td><td>https://accounts.google.com/signin/</td></tr>
<tr><td>Platform</td><td>AngularJS</td></tr>
<tr><td>Operating System</td><td>OS X 10.12.0</td></tr>
<tr><td>Browser</td><td>Chrome 53</td></tr>
<tr><td>Severity</td><td>Major</td></tr>
<tr><td>Assigned to</td><td>/</td></tr>
<tr><td>Priority</td><td>High</td></tr>
</table></td><td></td><td></td></tr>
</table>

Ans

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

9	**State any four attributes of defect.**	**S-23**	**2**
Ans	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 4) Module /Sub-module name: for which the defect is found. 5) Phase introduced: Phase of life cycle to which the defect belongs to. 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. I.e. security defect, functional defect, GUI defects etc.		
10	**Describe detect management process with neat diagram.** **Explain defect management process with suitable diagram.**	**S-23** **W-23** **S-24**	**4**
Ans	The process of finding defects and reducing them at the lowest cost is called as Defect Management Process.  **Defect Prevention:-** Implementation of techniques, methodology and standard processes to reduce the risk of defects. **Deliverable Baseline: -** Establishment of milestones where deliverables will be considered completes 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. D**efect 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. **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.		

<table>
<tr><td></td><td>Process Improvement: - Identification and analysis of the process in which a defect originated to identify ways to improve the process to prevent future occurrences of similar defects. Also the validation process that should have identified the defect earlier is analyzed to determine ways to strengthen that process.

Management Reporting: - Analysis and reporting of defect information to assist management with risk management, process improvement and project management.</td><td></td><td></td></tr>
<tr><td>11</td><td>Design test cases for online shopping website flipkart and prepare defect report of it</td><td>S-23</td><td>6</td></tr>
<tr><td>Ans</td><td>
<table>
<tr><td>Test ID</td><td>Test case</td><td>Input</td><td>Expected output</td><td>Actual output</td><td>Status</td><td></td></tr>
<tr><td>TC1</td><td>Verify the search product option</td><td>Search for products</td><td>Different products are appeared</td><td>Different products are appeared</td><td>pass</td><td></td></tr>
<tr><td>TC2</td><td>Verify the buy product option</td><td>Click buy button</td><td>Payment details appeared</td><td>Payment details appeared</td><td>pass</td><td></td></tr>
<tr><td>TC3</td><td>Verify the user account</td><td>Click account</td><td>User details displays</td><td>User details displays</td><td>pass</td><td></td></tr>
</table>
</td><td></td><td></td></tr>
<tr><td>12</td><td>Enlist any four attributes of defect. Describe them with suitable example.</td><td>W-23</td><td>4</td></tr>
<tr><td>Ans</td><td>Defect has following attributes: Any six of the following attributes shall be considered

1) Defect ID: Identifies defect as there are many defects might identified in system. a. i.e. D1, D2, etc.

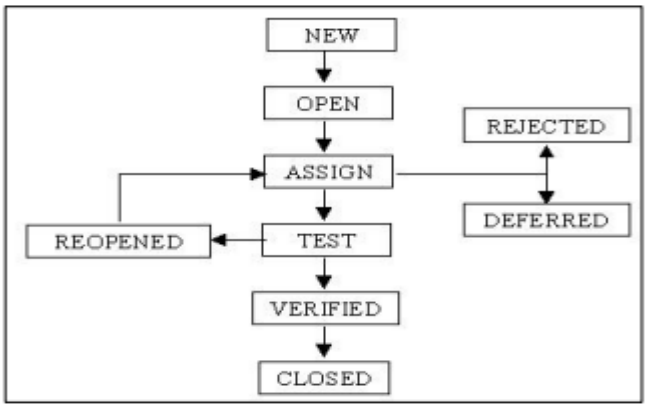
2) Defect Name: Name of defect which explains the defect in brief. a. It must be short but descriptive. i.e. Login error.

3) Project Name: Indicates project name in which defect is found

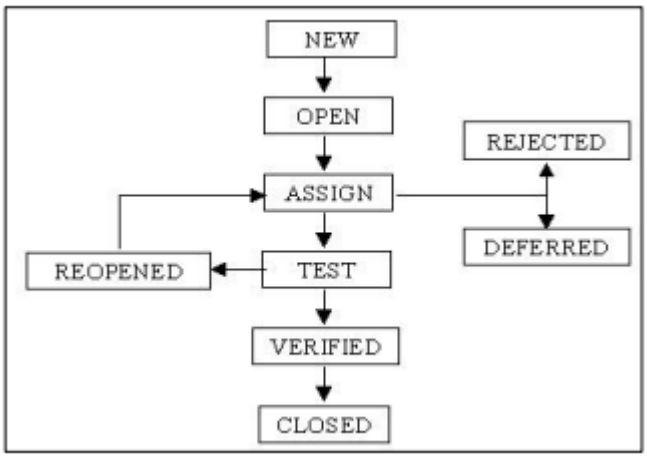
4) Module /Sub-module name: for which the defect is found.

5) Phase introduced: Phase of life cycle to which the defect belongs to.

6) Phase found: Phase of project when the defect is found is added here. It is used</td><td></td><td></td></tr>
</table>

	to find defect leakage or stage. 7) Defect type: Defines defect type. I.e. security defect, functional defect, GUI defects etc. 8) Severity: Declared in test plan, i.e. high medium or low. 9) Priority: defines on the basis of how the project decides a schedule to take the defects for fixing. 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.		
13	**Describe defect life cycle with neat diagram**	**W-23**	**6**
Ans	DEFECT LIFE CYCLE (Bug Life cycle) is the journey of a defect from its identification to its closure. The Life Cycle varies from organization to organization and is governed by the software testing process the organization or project follows and/or the Defect tracking tool being used.  • New: When the bug is posted for the first time, its state will be "NEW". This means that the bug is not yet approved. • Open: After a tester has posted a bug, the lead of the tester approves that the bug is genuine and he changes the state as "OPEN". • Assign: Once the lead changes the state as "OPEN", he assigns the bug corresponding developer or developer team. The state of the bug now is changed to "ASSIGN". • Test/Retest: Once the developer fixes the bug, he has to assign the bug to the		

	testing team for next round of testing. Before he releases the software with bug fixed, he changes the state of bug to "TEST". It specifies that the bug has been fixed and is released to testing team.// 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. • 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. • 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". • Verified: Once the bug is fixed and the status is changed to "TEST", the tester tests the bug. If the bug is not present in the software, he approves that the bug is fixed and changes the status to "VERIFIED". • Reopened: If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "REOPENED". The bug traverses the life cycle once again. • 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. • 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. • 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.		
14	**State Classification of defect.**	**S-24**	**2**
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. **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		
15	**Draw a diagram for defect life cycle and write example for defect template.**	**S-24**	**6**
Ans			

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

JSPM's

Rajashi Shahu College of Engineering, Polytechnic Tathawade

Department of Computer Engineering

Academic Year: 2024-2025

Course Name: Software Testing **Course code:** 22518

Semester: Odd **Name of the Faculty:** Prof. Suvarna A. Amle

UNIT NO. 5 Testing Tools and Measurements

CO5: Test software for performance measure using automation testing tools.

Q. No.	Questions	Year	Marks
1	**State any four advantages of using tools.**	**W-19**	**2**
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 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.		
2	**State any eight limitations of manual testing.**	**W-19** **S-22**	**4**
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 colour 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.		
3	**Enlist factors considered for selecting a testing tool for test automation.**	**W-19** **W-22**	**4**
Ans	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:

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.

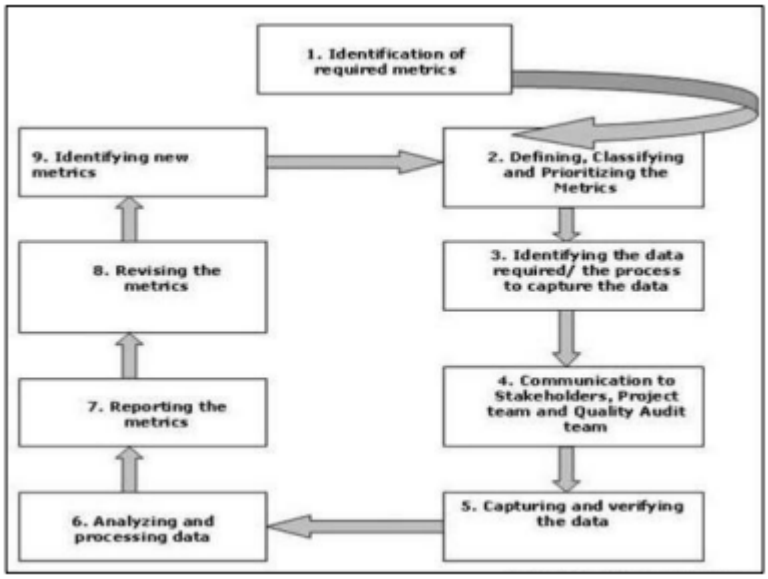
c) Finally, test tools are not 100% cross-platform. They are supported only on some O.S. platforms and the scripts generated from these tools may not be compatible on other platforms. Moreover, many of the test tools are capable of testing only the product, not the impact of the product/test tool to 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: some tools had Y2K-problemA 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	**Describe object oriented metrics in testing**	**W-19**	**4**
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 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 are 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.		
5	**Elaborate the term metrics and measurement and write the need of software measurement.**	**W-19**	**6**
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 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	**Enlist any four testing tools.**	**S-22**	**2**
Ans	Manual Testing Tool Automation Testing Tool Static Testing Tool Dynamic Testing Tool		
7	**Describe different factors for selecting testing tools.**	**S-22** **W-23**	**4**
Ans	The categories of criteria for selecting test tools are: 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.

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.		
8	**Describe need for Automated Testing tools.**	**S-22** **W-22**	**4** **2**
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. ● 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.		
9	**State the advantages and disadvantages of using tool**	**W-22**	**4**
	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. 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)		
10	**Design test cases for MS Word application using an Automation tool.**	**W-22**	**6**

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
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

11 **State any two differences between manual and automated testing.** **S-23 S-24** **2**

Ans

Parameters	Manual Testing	Automation Testing
Definition	In manual testing, the test cases are executed by the human tester.	In automated testing, the test cases are executed by the software tools.
Processing Time	Manual testing is time-	Automation testing is

				testing.		
		Resources requirement	Manual testing takes up human resources.	Automation testing takes up automation tools and trained employees.		
		Exploratory testing	Exploratory testing is possible in manual testing.	Exploratory testing is not possible in automation testing.		
		Framework requirement	Manual testing doesn't use frameworks.	Automation testing uses frameworks like Data Drive, Keyword, etc.		
		Reliability	Manual testing is not reliable due to the possibility of manual errors.	Automated testing is more reliable due to the use of automated tools and scripts.		
		Investment	In manual testing, investment is required for human resources.	In automated testing, investment is required for tools and automated engineers.		
12	**Describe any four limitations of manual testing.**				**S-23 W-23**	**4**
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.					

13	**Describe the criteria's to select testing tools.**	**S-23**	**4**
	Criteria for Selecting Test Tools: The Categories for selecting Test Tools are, 1. Meeting requirements; 2. Technology expectations; 3. Training/skills; 4. Management aspects **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. Given of the plethora of choice available, huge delay is involved in selecting and implementing test tools. **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. A good number of test tools require their libraries to be linked with product binaries. **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. **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		
14	**Define metrics and measurements. Describe three types of metrics.**	**S-23**	**6**
	Metrics & measurement: Metrics is a relative measurement of status of process or product in terms of two or more entities taken together for comparison. Measurements are key element for controlling software engineering processes. Types of Metrics: • Process quality • Product quality		

	• Objective Metrics • Subjective Metrics **Process quality:** Activities related to the production of software, tasks or milestones. 1. Process metrics are collected across all projects and over long periods of time. 2. They are used for making strategic decisions. 3. The intent is to provide a set of process indicators that lead to long-term software process improvement. 4. The only way to know how/where to improve any process is to: • Measure specific attributes of the process. • Develop a set of meaningful metrics based on these attributes. **Product quality:** Explicit result of the software development activity, deliverables, products. 1. Product metrics help software engineers to better understand the attributes of models and assess the quality of the software. 2. They help software engineers to gain insight into the design and construction of the software. 3. Focus on specific attributes of software engineering work products resulting from analysis, design, coding, and testing. 4. Provide a systematic way to assess quality based on a set of clearly defined rules. 5. Provide an "on-the-spot" rather than "after-the-fact" insight into the software development. **Objective Metrics:** 1. They are non-negotiable – that is the way they are defined doesn't change with respect to the niche or the type of endeavour they are being applied to. 2. Actual cost or AC is always the total cost actually incurred in accomplishing a certain activity or a sequence of activities. **Subjective Metrics:** These metrics are a relatively new precept and are more flexible than the rigid framework of the objective metrics. Subjective metrics do deal with performance but the approach is more tailored. For some enterprises the niche in which they function forces project management to change in order to adapt to the demands of the workplace.		
15	**Enlist any four software testing tools.**	**W-23**	**2**

	1. Manual Testing tool 2. Automation testing tool 3. Static Testing tool 4. Dynamic Testing tool		
16	**State & explain any four benefits of automation in testing.**	**W-23**	**4**
	Benefits of automation testing: 1. Speed: Think about how long it would take you to manually try a few thousand test cases for the windows Calculator. You might average a test case every five seconds or so. Automation might be able to run 10, 100 even 1000 times that fast. 2. Efficiency: While you are busy running test cases, you can‟t be doing anything else. If you have a test tool that reduces the time it takes for you to run your tests, you have more time for test planning and thinking up new tests. 3. Accuracy and Precision: After trying a few hundred cases, your attention may reduce and you will start to make mistakes .A test tool will perform the same test and check the result perfectly, each and every time. 4. Resource Reduction: Sometimes it can be physically impossible to perform a certain test case. The number of people or the amount of equipment required to create the test condition could be prohibitive. A test tool can used to simulate the real world and greatly reduce the physical resources necessary to perform the testing. 5. Simulation and Emulation: Test tools are used to replace hardware or software that would normally interface to your product. This “face” device or application can then be used to drive or respond to your software in ways that you choose-and ways that might otherwise be difficult to achieve. 6. Relentlessness: Test tool and automation never tire or give up. It will continuously test the software.		
17	**State the need of Automated testing.**	**S-24**	**2**
	Need for automated testing tools are: • An automated testing tool can playback pre-recorded and predefined actions, compare the results to the expected behaviour 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 rel		
18	**How to select testing tool? Explain in detail.**	**S-24**	**6**
	 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. election 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. • 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.		

Jaywant Shikshan Prasarak Mandal's
RAJARSHI SHAHU COLLEGE OF ENGINEERING, POLYTECHNIC, TATHAWADE.
S.No.80, Pune-Mumbai Bypass Highway, Tathawade Campus, Pune.

MSBTE

QUESTION PAPER

WINTER-19 TO SUMMER-24

22518

11920

3 Hours / 70 Marks Seat No. | | | | | | | | |

Instructions – (1) All Questions are *Compulsory.*

(2) Answer each next main Question on a new page.

(3) Illustrate your answers with neat sketches wherever necessary.

(4) Figures to the right indicate full marks.

(5) Assume suitable data, if necessary.

(6) Mobile Phone, Pager and any other Electronic Communication devices are not permissible in Examination Hall.

Marks

1. Attempt any FIVE of the following: **10**

a) Define static and dynamic testing.

b) State any two example of integration testing.

c) Enlist any two activities involved in test planning.

d) Enlist objectives of software testing.

e) Define Defect

f) State any four advantages of using tools.

g) Define Bug, Error, Fault and Failure.

Marks

2. Attempt any THREE of the following: **12**

a) Describe Boundry value analysis with suitable example.

b) Differentiate between drivers and stub (any four points)

c) State the contents of 'Test Summary Reports' used in test reporting

d) State any eight limitations of manual testing

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

a) Describe the use of decision table in black box testing with the help of suitable example

b) Describe standards included in Test Management.

c) Enlist different techniques for finding defects and describe any one technique with an example.

d) Enlist factors considered for selecting a testing tool for test automation.

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

a) Differentiate between alpha and beta testing (any four points)

b) Describe test infracture management

c) Describe the process of preparing summary report in test planning

d) Describe object oriented metrics in testing.

e) State the testing approaches that are considered during client-server testing.

Marks

5. Attempt any TWO of the following: **12**

a) Design test cases for railway reservation system.

b) With respect to GUI testing write the test cases for Amazon login form

c) Elaborate the term metrics and measurement and write the need of software measurement.

6. Attempt any TWO of the following: **12**

a) Design test cases for hostel admission form of your institute.

b) Design a test plan along with the test cases for edit function in notepad.

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

Summer-22

22518

21222

3 Hours / 70 Marks Seat No. | | | | | | | | |

15 minutes extra for each hour

Instructions – (1) All Questions are *Compulsory.*

(2) Answer each next main Question on a new page.

(3) Illustrate your answer with neat sketches wherever necessary.

(4) Figures to the right indicate full marks.

(5) Assume suitable data, if necessary.

(6) Mobile Phone, Pager and any other Electronic Communication devices are not permissible in Examination Hall.

Marks

1. Attempt any FIVE of the following: **10**

a) Define verification and validation with respect to software testing.

b) State the process of performance testing.

c) Enlist different types of test deliverables.

d) Define following terms –

i) Error.

ii) Fault.

iii) Bug.

iv) Failure.

e) State any four defect reporting guidelines.

f) Enlist any four testing tools.

g) Enlist any four skills for software tester.

Marks

2. Attempt any THREE of the following: **12**

a) Describe Code Complexity testing.

b) Differentiate between Smoke Testing and Sanity Testing. (any four points).

c) Describe different types of external standards.

d) Describe different factors for selecting a testing tools.

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

a) Differentiate between Quality Assurance and Quality Control.

b) Explain test management with Test Infrastructure Management and Test People Management.

c) Draw defect prevention process cycle. State working of each phase.

d) State any eight limitations of Manual Testing.

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

a) Describe Unit Testing.

b) Explain test reporting with executing test cases and preparing test summary report.

c) Explain test case specification.

d) Describe need for Automated Testing tools.

e) Explain following concepts related to security testing –

i) Authorization.

ii) Access Control.

Marks

5. Attempt any TWO of the following: **12**

a) Design test cases for Online Mobile Recharge
(Data filed are mobile number, state, email-id, recharge amount.)

b) With respect to client-server testing design test cases for Online Payment Transfer at banking system.

c) Elaborate the concept of Software Metrics? Describe Base Metrics and calculated matrics with suitable example.

6. Attempt any TWO of the following: **12**

a) With respect to GUI testing, write test cases for Flipkart login form.

b) Prepare a test plan along with test cases for creating a saving account at bank. (Test cases should be atleast six).

c) Design any four test cases for withdraw an amount from ATM and prepare defeat report of it.

22518

12223

3 Hours / 70 Marks Seat No. | | | | | | | | |

Instructions – (1) All Questions are *Compulsory.*

(2) Illustrate your answer with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

(5) Mobile Phone, Pager and any other Electronic Communication devices are not permissible in Examination Hall.

Marks

1. Attempt any FIVE of the following: **10**

a) Define the testing terminology –

i) Error

ii) Fault

iii) Defect

iv) Bug

b) List the levels of testing.

c) State any four needs to prepare a test plan.

d) Give the defect classification and its meaning.

e) Compare verification and validation (any two points).

f) State the need of automated testing tools.

g) Give the objectives of software testing.

Marks

2. Attempt any THREE of the following: **12**

a) State the Entry and Exit criterias for the software testing.

b) State and describe top-down approach of integration testing with diagram.

c) Describe the 'Test Infrastructure' components with diagram.

d) State the limitations of manual testing.

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

a) Differentiate between white box testing and black box testing (any four points).

b) State the contents of "Test Summary Reports" used in test reporting.

c) Prepare defect report after executing test cases for any login form.

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

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

a) Describe graphical user interface (GUI) testing and its important traits.

b) Describe test deliverables in details.

c) Describe load testing and stress testing with suitable example.

d) State the advantages and disadvantages of using tools.

e) Write the test cases for Notepad application. (any eight test case)

5. Attempt any TWO of the following: **12**

a) Design test cases for simple calculator application. (Black box testing.) (Any six points.)

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

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

Marks

6. Attempt any TWO of the following: **12**

a) Write program for calculating even numbers from 1 to 20 And design test cases for the same.

b) Prepare test plan for 'Cam Scanner' which is installed on mobile.

c) Prepare defect report after executing test cases for withdrawn of amount from ATM machine.

Summer- 23

22518

22223

3 Hours / 70 Marks Seat No. | | | | | | | | |

Instructions – (1) All Questions are *Compulsory.*

(2) Illustrate your answers with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

(5) Mobile Phone, Pager and any other Electronic Communication devices are not permissible in Examination Hall.

(6) Write the answers in sequential order.

Marks

1. Attempt any FIVE of the following: **10**

a) Write two differences between quality assurance and quality control.

b) Define test plan with two advantages.

c) Enlist any two advantages of acceptance testing.

d) Write steps to prepare test test cases.

e) State any four attributes of defect.

f) State any two difference between manual and automated testing.

g) Define following terms

i) Defect

ii) Bug

Marks

2. Attempt any THREE of the following: **12**

a) Describe any four skills of software tester.

b) Describe the concept of driver and stub with suitable example.

c) Describe the factors considered to decide test approach.

d) Describe any four limitations of manual testing.

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

a) List types of white box testing. Describe any two types of white box testing with neat diagram.

b) Describe the 'Test Infrastructure' components with neat diagram.

c) Describe detect management process with neat diagram.

d) Differentiate between static and dynamic testing tools. (any four points).

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

a) Describe the types of integration testing with neat diagram.

b) Explain the testing approaches that are considered during web application testing.

c) Describe test management with test people management.

d) Describe the criterias to select testing tools.

e) Describe the regression testing. State entry and exit criteria for regression testing.

5. Attempt any TWO of the following: **12**

a) Design test cases for hostel admission form.

b) Write test cases for facebook login form with respect to graphical user interlac (GUI) testing.

c) Define metrics and measurements. Describe three types of metrics.

Marks

6. Attempt any TWO of the following: **12**

a) Design test cases for online railway reservation applications.

b) Prepare a test plan along with the test cases for the MS Excel option "Save AS".

c) Design test cases for online shopping website flipkart and prepare defect report of it.

Winter-23

22518

23124

3 Hours / 70 Marks Seat No. | | | | | | | | |

Instructions : (1) All Questions are *compulsory.*

(2) Illustrate your answers with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

Marks

1. Attempt any FIVE of the following : **10**

(a) Compare verification and validation (any 2 point).

(b) Define failure, error, fault, bug.

(c) List the objectives of software testing. (any four)

(d) Define Driver and stub.

(e) What is GUI testing ? Give one example.

(f) Write any two root causes of defect.

(g) Enlist any four software testing tools.

2. Attempt any THREE of the following : **12**

(a) State the Entry and Exit criteria's for the software testing.

(b) Illustrate process of bi-directional integration testing. State its two advantages & disadvantages.

(c) Enlist any four attributes of defect. Describe them with suitable example.

(d) Describe any four factors for selecting a testing tools.

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

(a) Differentiate between alpha testing and beta testing. (any four points)

(b) Prepare test plan for Notepad application. (Windows based)

(c) Explain defect management process with suitable diagram.

(d) State & explain any four benefits of automation in testing.

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

(a) What is boundary value analysis ? Explain with suitable example.

(b) Explain the Regression testing. State when the Regression testing shall be done.

(c) What is test plan ? What is its need ? List test planning activities.

(d) Prepare defect report for login field of email application.

(e) State any four limitations of manual testing.

5. Attempt any TWO of the following : **12**

(a) Describe V-model with labelled diagram.

(b) Describe with one example each :

(i) Load testing

(ii) Stress testing

(c) Prepare six test cases for marketing site www.flipkart.com.

6. Attempt any TWO of the following : **12**

(a) Explain client-server testing with suitable diagram.

(b) Write important six test cases for the 'Login Form' of the Facebook website.

(c) Describe defect life cycle with neat diagram.

23242

3 Hours / 70 Marks

Seat No. | | | 2 | 7 | 1 | 6 | 7 | 8 |

Instructions : (1) All Questions are *compulsory.*

Summer-24

(2) Illustrate your answers with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

Marks

1. Attempt any FIVE of the following : **10**

(a) Enlist objectives of software testing.

(b) Compare Alpha testing and Beta testing. (Any two differences)

(c) Define a test plan.

(d) List any four skills of software tester.

(e) State the classification of defects.

(f) State the need of automated testing tool. (Any two)

(g) Define the terms error and defect in relation with software testing.

2. Attempt any THREE of the following : **12**

(a) Write any four differences between quality assurance and quality control.

(b) Explain GUI testing with example.

(c) Describe the contents of "Test Summary Report' used in test reporting.

(d) Differentiate between static and dynamic testing tools. (any four points)

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

(a) Distinguish between white box testing and black box testing. (any four points)

(b) Explain the need of test deliverables & test plan for test planning.

(c) Explain defect management process with suitable diagram.

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

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

(a) State the process of performance testing.

(b) Explain people management in test planning.

(c) Explain test infrastructure management with its component.

(d) Describe any four limitations of manual testing.

(e) Describe acceptance testing with its advantages.

5. Attempt any TWO of the following : **12**

(a) Prepare and write six test cases for Library Management System of college.

(b) With the help of diagram, describe client-server testing.

(c) How to select a testing tool ? Explain in detail.

6. Attempt any TWO of the following : **12**

(a) Explain the need of stubs and drivers with diagram and its importance in software testing.

(b) Explain in detail, how to prepare a test plan with suitable example.

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

ALL THE BEST