0. 1 2					
2	Ouestion Concept of "Total Quality" was created in	Option 1 Janan	Option 2 US	Option 3 UK	Option 4 South Korea
		Japan Specification	US Market	UK Developer	South Korea Company
3	Quality can be recongnized but not defined. This is called as the view of quality	Transcendental	User	Product	Manufacturing
4	"Quality is fitness for purpose". This is called as the	Product	User	Transcendental	Manufacturing
5	When Quality depends on the amount which the customer is willing to pay, it is called as	User	Manufacturing	Value	Product
	The cost which arises from the efforts to prevent defects is	Appraisal cost	Prevention cost	Failure cost	Miscellaneous cost
6	called as The cost which aries from defects like rework, repair etc. are	Internal Failure Cost	External Failure	Appraisal Cost	Prevention Cost
7	called as		Cost		
8	The cost which aries from tasks like helpline support, warranty etc. are called as	Appraisal Cost	Prevention Cost	Internal Failure Cost	External Failure Cost
9		Cost of Failure of	Appraisal Cost	Cost of Internal	Prevention Cost
10	is the conformity of the software with the actual	Reliability	Performance	Security	Functionality
10	requirements and specifications is the degree to which an application is protected	Reliability	Security	Performance	Functionality
11	against malicious attacks.	-		a	
12	is the ability of the program to adapt to possible changes in its requirements.	Flexibility	Functionality	Security	Performance
	is the ability of an application to consistently	Reliability	Performance	Usability	Security
13		Performance	Reliability	Functionality	Usability
15	"Delivering the right product" is part of view of	Customer	Developer	Supplier	Market
	principle of TQM (Total Quality Management) aims to create a culture where employees feel involved with the	Total Employee involvement	Total Staff involvement	Member involvement	Resource involvement
16	organization				a
	principle of TQM (Total Quality Management) aims to create proper functioning between units of the	Integrated System	Combined System	Department System	Staff System
17	organization	D.C. M.S.		DC M S	D.C. M
18	The cycle of Improvement Sequence is: Communication in TQM (Total Quality Management) can be	Define, Monitor, Strategies	Define, Methods	Define, Monitor, Timelines	Define, Measure, All of the above
19	based on which of the following?	÷			
20	The most successful tool used for Statistical Process Control (SPC) is	Gantt Chart	Control Chart	Line Chart	Bar Chart
	Which among the following are included among the	Process-centered	Customer-	Integrated System	All of the above
21	principles of TQM (Total Quality Management)? Organizational refers to the pattern of shared	Setting	focused Focus	Culture	Morals
22	values, beliefs etc. within an organization.				
23	The way to act and think within an organization is influenced	Organizational Setting	Organizational Culture	Organizational Focus	Organizational Moral
24	Organizational culture can include which of the following?	Setting Written rules	Culture Unwritten rules	Beliefs	All of the above
25	PDCA concept is related to?	Process	Process	Process Selection	None of the above
26		Plan, Do, Change, Cost of Prevention	Plan, Do, Cost of	Plan, Decide, Check, Cost of Failure	Plan, Do, Check, Cost of Control
	With respect to PDCA cycle- " If any deviations are	Plan	Do	Change	Act
28	observed in actual outcomes with respect to planned results the organization may need to decide the actions". This will				
29	Which of the following could be example of metrics for	Number of tests	Number of tests	Number of tests failed	
30 31	Which of these charts is used as a Problem solving Software "Ishikawa diagram" is also known as	Block diagram Fishbone diagram	Flowchart Linechart	Histogram Block diagram	Line chart Flow diagram
	Which of these are the components of a Fishbone diagram?	Head	Backbone	Causes	All of the above
33	Normally, Testing covers how much time of a Software	0 - 10 %	30 - 40 %	5 - 15 %	60 - 70 %
	How is the concept of Productivity described for working	Output / Input	Output x Input	Output + Input	Output - Input
35	Improvement in Quality has a effect on	Positive	Negative	No effect	Worsening
36	Which of the following are the mandatory way of doing characteristic of a Software refers to its ability	Guidelines Functionality	Standards Reliability	Templates Efficiency	Format Maintainability
37	to be used in most effective manner.	-	-		
38	Ability to transform a software from one working platform to another is called as	Security	Portability	Usability	Functionality
	Modifying the software to correct errors is referred to as	Efficiency	Maintainability	Portability	Security
40	Abiliity to upgrade a software for more number of users is referred to as	Scalability	Portability	Security	Efficiency
	Which of the following is the 3rd Tier of Quality	Quality Manual	Quality Policy	Quality Objectives	Quality Process
41					
44	Users can be trained to use the software product during	Deployment	Maintainance	Coding	Testing
43 44		MS-Windows	Word Processor	Linux	Unix
44		Calculator	Linux	Word Processor	Adode Photoshop
46	An Operating system can be considered an example of	Application Severity	Embedded	System Software	Non-system software
47	is defined as the degree of impact a defect has				
	on the development of a Software.	Severny	Impact	Effect	Output
48	defect affects the functionality of the software.	Low	Impact Minor	Major	Small
48 49	defect affects the functionality of the software. Spelling mistake on a website is the example of a	-	Impact	Major Minor	Small
48 49	defect affects the functionality of the software. Spelling mistake on a website is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ?	Low Critical Mission Camouflage Effect	Impact Minor Major	Major Minor Objective Coverage Effect	Small Low Vision Redundant Code
48 49 50 51 52	defect affects the functionality of the software. Spelling mistake on a website is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ? Which of the following is not included in failure costs?	Low Critical Mission Camouflage Effect rework	Impact Minor Major Goal Cascading Effect repair	Major Minor Objective Coverage Effect failure mode analysis	Small Low Vision Redundant Code none of the mentioned
48 49 50 51	defect affects the functionality of the software. Selling mittake on a website is the example of a To become a billion dollar company could be termed as: 'One defect hields another defect' is termed as? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not included in External failure Which of the following is not included in EXTERNAL for SOA?	Low Critical Mission Camouflage Effect rework Testing inter-process	Impact Minor Major Goal Cascading Effect repair help line support maintenance	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning	Small Low Vision Redundant Code none of the mentioned complaint resolution testing
48 49 50 51 52 53 54	defect affects the functionality of the software. Spelling mistake on a website is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not included in External failure	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from	Impact Major Goal Cascading Effect repair help line support maintenance The cost arises	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from	Small Low Vision Redundant Code none of the mentioner complaint resolution testing The cost arises from
48 49 50 51 52 53	defect affects the functionality of the software. Selling mittake on a website is the example of a To become a billion dollar company could be termed as: 'One defect hields another defect' is termed as? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not included in External failure Which of the following is not included in EXTERNAL for SOA?	Low Critical Mission Camouflage Effect rework Testing inter-process	Impact Minor Major Goal Cascading Effect repair help line support maintenance	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from efforts to prevent	Small Low Vision Redundant Code none of the mentioned complaint resolution testing
48 49 50 51 52 53 54	defect affects the functionality of the software. Seelling mistake on a webbie is the example of a To become a billion dollar company could be termed as: 'One defect hides another defect' is termed as ? Which of the following is not included in failure costs? Which of the following is not included in External failure Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? is define as the degree of inpact a defect has on the development of a component application being test.	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality	Impact Minor Major Goal Cascading Effect repair help line support maintenance The cost arises form defects Product	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from efforts to prevent Severity	Small Low Vision Redundant Code none of the mentione complaint resolution testing The cost arises from efforts to implement Process
48 49 50 51 52 53 54 55	defect affects the functionality of the software. Spelling mistake on a website is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? is define as the degree of impact a defect has on the development of a component application heim test.	Low Critical Mission Canouflage Effect Tework Testing inter-process The cost arises from efforts to prevent Quality Total quality	Impact Minor Major Goal Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from efforts to prevent Severity Requirement	Small Low Vision Redundant Code none of the mentione complaint resolution testing The cost arises from efforts to implement Process Project Quality
48 49 50 51 52 53 54 55 55 56 57	defect affects the functionality of the software. Spelling mistake on a website is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not included in failure costs? Which of the following is not appraisal cost in SOA? What is prevention cost? is define as the degree of impact a defect has on the development of a component application being test. In a mark is used to trace the requirement to the test that are needed to verify whether the requirement to To mark is a star of software testing where	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality	Impact Minor Major Goal Cascading Effect repair help line support maintenance The cost arises form defects Product	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from efforts to prevent Severity	Small Low Vision Redundant Code none of the mentione complaint resolution testing The cost arises from efforts to implement Process
48 49 50 51 52 53 54 55 55 56 57	defect affects the functionality of the software. Seelling mistake on a webbit is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? and the event of a component application being test. 	Low Critical Mission Canouflage Effect rework Testing Effect rework The cost arises from efforts to prevent Quality Total quality management Mutation	Impact Minor Major Goal Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement Traceability	Maior Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from efforts to prevent Severity Requirement engineering Big bang	Small Low Vision none of the mentione complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value
48 49 50 51 52 53 54 55 55 56 57	defect affects the functionality of the software. Seelling mistake on a webbit is the example of a To become a billion dollar company could be termed as: "One defect hide another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? and the defect of a component application being test. 	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test	Impact Minor Maior Goal Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement Traceability Decision Table Beta test	Major Minor Objective Coverage Effect failure mode analysis warantw work quality planning The cost arises from efforts to prevent Severity Requirement engineering Big hang Regression test	small Low Vision Redundant Code none of the menione complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing
48 49 50 51 52 53 54 55 56 57 58	defect affects the functionality of the software. Selling mistake on a website is the example of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as? Which of the following is not included in falture costs? Which of the following is not included in falture costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? is define as the degree of impact a defect has on the develomment of a component application becing test. matrix is used to trace the requirement on test that are needed to verify whether the requirement to the sufficient of the following is a type of software testing where we change certain statements in the source code and check if is a verify of the route source of an end of the custo of the complex software to solve the depleced by customer at his or her own site without the developer being is high the input values to be passed to the	Low Critical Mission Canouflage Effect rework Testing Effect rework The cost arises from efforts to prevent Quality Total quality management Mutation	Impact Minor Maior Goal Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement Traceability Decision Table	Maior Minor Objective Coverage Effect failure mode analysis warranty work quality planning The cost arises from efforts to prevent Severity Requirement engineering Big bang	Small Low Vision none of the mentione complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value
48 49 50 51 52 53 54 55 56 57 58 59 60	defect affects the functionality of the software. Seelling mistake on a weakie is the scannels of a To become a billion dollar company could be termed as: "One defects hides another defect" is termed as? Which of the following is not included in falture costs? Which of the following is not included in falture costs? Which of the following is not included in External failure Which of the following is not included in External falture Which of the following is not included in External falture which of the following is not included in External falture the development of a component application being test. In the development of a component application being test. In that are needed to verify whether the reguirement to the that are needed to verify whether the reguirement are "Testing is a type of software testing where is define on the compilete software tested by customer at his or her own site without the developer being in singly the input values to be passed to the system under test. "In the software tested to be compilete software tested by customer at his or her own site without the developer being is singly the input values to be passed to the system under test.	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test	Impact Minor Maior Goal Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement Traceability Decision Table Beta test	Major Minor Objective Coverage Effect failure mode analysis warantw work quality planning The cost arises from efforts to prevent Severity Requirement engineering Big hang Regression test	small Low Vision Redundant Code none of the menione complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing
48 49 50 51 52 53 54 55 56 57 58 59 60 61	defect affects the functionality of the software. Seelling mistake on a webbit is the acumple of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not an appraial cost in SOA? What is prevention cost? What is prevention cost? What is prevention cost? The development of a component application being test. 	Low Critical Mission Canouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan	Impact Minor Major Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement Traceability Decision Table Beta test Test Design Test document	Major Minor Objective Coverage Effect failure mode analysis warnarty work quality clauming The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case	Small Low Vision Redundan Code none of the mentiones complaint resolution testing The cost arises from efforts to implement Project Quality Management Boundary value System Testing Test Cases Test note
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62	defect affects the functionality of the software. Seelling mistake on a website is the example of a To become a billion dollar company could be termed as: "On defect hides another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not an appraial cost in SOA? What is prevention cost? What is prevention cost? What is prevention cost? What is prevention cost? What is prevention cost? The development of a component application being test. matrix is used to trace the requirement to the est that are needed to verify whether the requirement are Testing is a type of software testing where we chance certain statements in the source code and check if customer at his or ber own site without the developer being is a define work products to be testichwork my will tested and test type. cod arises from efforts to defects. defines work products to be	Low Critical Mission Camouflage Effect rework Testing Testing Inter-process The cost arises from efforts to prevent Quality management Mutation Alpha test Test Plan	Impact Minor Major Goal Cascading Effect repair help line support maintenance The cost arises form defects Product Requirement Traccability Decision Table Beta test Test Design	Major Minor Objective Coverage Effect failure mode analysis warrantw work warrantw work mode analysis warrantw work The cost arises from efforts to prevent Severity Requirement engineering Big hang Regression test Test Data	Small Low Vision Redundant Code none of the mentione complaint resolution lesting The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases
48 49 50 51 52 53 54 55 56 57 58 59 60 61	defect affects the functionality of the software. Seelling mistake on a website is the scannel of A. To become a billion dollar company could be termed as: "On defect hides another defect" is termed as ? Which of the following is not included in failure cons? Which of the following is not anaptrainal cost in SOA? What is prevention cost? What is prevention cost? What is prevention cost? In the following is not an appraisal cost in SOA? What is prevention cost? In the define as the degree of impact a defect has on the development of a component application being test. 	Low Critical Mission Camouflage Effect rework Testing envork florts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Test Plan	Impact Minor Major Goal Cascading Effect repair Major Belo Inn support helo Inn support from defects Product Requirement Tracehility Decision Table Beta test Test Design Test document Prevention Test Strategy	Major Minor Objective Coverage Effect Coverage Effect (aulare mode analysis warantw work quality planning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case	small Low Vision Redundan Code men of the mentiones complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64	defect affects the functionality of the software. Seelling mistake on a website is the scannel of A. To become a billion dollar company could be termed as: "On defect hides another defect" is termed as ? Which of the following is not included in failure cons? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? What is prevention cost? is define as the degree of impact a defect has on the development of a component application being test. 	Low Critical Mission Camouflage Effect rework Testing envork florts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity	Impact Minor Major Goal Cascading Effect repair Major Belo Inn support helo Inn support helo Inn support Major Major Product Requirement Tracehility Decision Table Beist test Test Design Test document Prevention Test Strategy Priority	Major Minor Objective Coverage Effect failure mode analysis warranty work quality planning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure	small Low Vision Redundan Code men of the mentione: complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure.
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63	defect affects the functionality of the software. Seelling misside and na website is the example of a To become a billion dollar company could be termed as: "One defect hieles another defect" is termed as ? Which of the following is not included in future conts? Which of the following is not included in future conts? Which of the following is not an appreadal cost in SOA? What is prevention cost? is define as the degree of impact a defect has on the development of a component application beint test. 	Low Critical Mission Canouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Finding broken code	Impact Minor Major Grain Cascading Effect repair Relation of the surport minimenance The cost arrises from defects Product Requirement Traceability Decision Table Beta test Test Design Test document Prevention Test Strategy Priority A stage of all	Major Minor Objective Objective Coverage Effect failure mode analysis warnativ work ounlity claiming The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internsity Evaluating deliverable	Small Low Vision Redundar Code none of the mentiones complaint resolution testing The cost arises from efforts to implement Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note Complexity process of developing
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64	defect affects the functionality of the software. Seelling mistake another defects example of A. To become a billion dollar company could be termed as: "On defect hides another defect" is termed as ? Which of the following is not included in failure costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? What is prevention cost? is define as the degree of impact a defect has on the development of a component application being test. 	Low Critical Mission Camouflage Effect rework Testing envork florts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity	Impact Minor Major Grafi Cascaling Effect respir Pablic Bissen Product Requirement Traceability Decision Table Beta test Test Design Test document Prevention Test Strategy Priority A stage of all Integration Lessing	Major Minor Objective Coverage Effect failure mode analysis waranty work quality planning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internal failure	small Low Vision Redundan Code men of the mentione: complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure.
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 63 64 65 66 67	defect affects the functionality of the software. Scelling mistake another defects wearangle of A. To become a billion dollar company could be termed as: "On defect hides another defects" is termed as ? Which of the following is not included in future costs? Which of the following is not an appraisal cost in SQA? What is prevention cost? What is prevention cost? What is prevention cost? What is prevention cost? Is define as the degree of impact a defect has on the development of a component application being test. 	Low Critical Mission Camouflage Effect rework, rework, resources for so prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Severity Finding byken code Unit testing errors	Impact Minor Minor Goal Casading Effect repair Maiorenaue Help Insurpro- Help Insurpro- Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Insurpro- Help Insurpro- Help In	Major Minor Objective Coverage Effect failure mode analysis waranty work quality planning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internal failure Test case Internsity Evaluating deliverable System testing bags	Small Low Vision Redundan Code complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note Complexity process of developing Component
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68	defect affects the functionality of the software. Seelling misside anon averbaic is the acumple of a To become a billion dollar company could be termed as: "One defect hieles another defect" is termed as ? "Which of the following is not included in failure costs? Which of the following is not an appreciated one is a soft which of the following is not an appreciated one is SOA? "What is prevention cost?" What is the following is not an appreciated cost in SOA? "What is prevention cost? "In the soft of the following is not an appreciated one is a soft the development of a component application being test, matrix is used to trace the requirement to the extinct after an effect on the soft of the soft of the soft test that are needed to verify whether the requirement are ending the soft of the complete software testing where the charace ended to verify whether the requirement are is a version of the complete software testing where is a version of the complete software testing where is a document is a high level document develope being software under test. 	Low Critical Mission Canouflage Effect rework Testine Inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Finding broken code Unit testing errors maintenance	Impact Minor Grad Cascaling Effect repair Leb line surport minimenance The cost arises from defects Product Requirement Traceability Decision Table Beta test Test Design Test document Prevention Test Strategy Priority A stage of all Integration testing failures design	Major Minor Objective Objective Coverage Effect failure mode analysis warnativ work ounlity alamine The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Intensity Evaluating deliverable System testing bags coding	Small Low Vision Redundant Code nones of the mentiones complaint resolution testing The cost arises from efforts to implement Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note Complexity process of developing Component defects
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69	defect affects the functionality of the software. Selfing missides another defects the example of a To become a billion dollar company could be termed as: To hection their some of the dollar software of the following is not included in failure conts? Which of the following is not included in failure conts? Which of the following is not an apprecial cost in SOA? Which of the following is not apprecial cost in SOA? Which of the following is not apprecial cost in SOA? Which of the following is not apprecial cost in SOA? What is prevention cost?	Low Critical Mission Camouflage Effect rework, rework, resources for so prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Severity Finding byken code Unit testing errors	Impact Minor Minor Goal Casading Effect repair Maiorenaue Help Insurpro- Help Insurpro- Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Help Insurpro- Insurpro- Help Insurpro- Help In	Major Minor Objective Coverage Effect failure mode analysis waranty work quality planning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internal failure Test case Internsity Evaluating deliverable System testing bags	Small Low Vision Redundan Code complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note Complexity process of developing Component
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68	defect affects the functionality of the software. Selling mistake another defects the example of A To become a billion dollar company could be termed as: To hedren their somber defects' its termed as ? Which of the following is not included in future costs? Which of the following is not an appreciable of Externed as ? Which of the following is not an appreciable of Externed as ? Which of the following is not an appreciable of Externed as ? Which of the following is not an appreciable of Externed as ? Which of the following is not an appreciable of Externed as ? Which of the following is not an appreciable of Externed as ? If the externed as the degree of impact a defect has on the development of a component application being test.	Low Critical Mission Camouflage Effect rework, resource from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Finding by eken code Unit testing errors maintenance always possible QA leader	Impact Minor Minor Goal Cascading Effect Cascading Effect Exacading Effect Bible Innesupport Help Innesupport Minor Minor Minore Innesu Product Requirement Product Beta test Test Design Test document Prevention Test Strategy Priority A stage of all Integration testing Integration testing Integration testing Integration	Major Minor Objective Coverage Effect failure mode analysis warnaty work quality blanning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internsity Evaluating deliverable System testing bugs coding bugs coding test engineer	Small Low Vision Redmanar Code more of the mentiones: complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note Complexity process of developing Complexity defects documentation impractical and Program manager
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69	defect affects the functionality of the software. Seelling mistake another defects a example of A To become a billion dollar company could be termed as: "To defect hides another defects" its termed as ? Which of the following is not included in future costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? The term of the following is not an appraisal cost in SOA? What is prevention cost? The term of the following is not an appraisal cost in SOA? What is prevention cost? The term of the following is not an appraisal cost in SOA? What is prevention cost? The term of the component application before the term matrix is used to trace the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that the test of the requires the source code and check if is a version of the comput values to be passed to be system under test. 	Low Critical Mission Camouflage Effect rework resources enter-process Inter-oreal and the efforts to prevent Quality Total quality Mutation Alpha test Test Plan Test Plan Severity Finding by exercise Severity Finding by exercise entors entors entors	Impact Minor Minor Goal Casading Effect repair Major Belo Insurpro- holo Insurpro- holo Insurpro- holo Insurpro- maintenance The cost arises form defects Product Requirement Traceability Decision Table Beta test Test document Prevention Test Strategy Priority A stage of all Integration testing failures: design practically	Major Minor Objective Coverage Effect failure mode analysis warnaty work quality blanning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internsity Evaluating deliverable System testing bugs coding bugs coding	Small Low Vision Redundan Code complaint resolution testing from of the mentiones complaint resolution testing from testing from testing Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note Complexity process of developing Component defects documentation impractical and
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71	defect affects the functionality of the software. Selling missides another defects' tax example of a To become a billion dollar company could be termed as: To hection the some of the second seco	Low Critical Mission Camouflage Effect rework, resource from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Finding by eken code Unit testing errors maintenance always possible QA leader	Impact Minor Minor Goal Cascading Effect Cascading Effect Exacading Effect Bible Innesupport Hole loss and Hole Innesupport Minor Market Product Requirement Product Requirement Product Beta test Test Design Test document Prevention Test Strategy Priority A stage of all Integration testing Integration testing Integration testing Integration	Major Minor Objective Coverage Effect failure mode analysis warnaty work quality blanning. The cost arises from efforts to prevent Severity Requirement engineering Big bang Regression test Test Data Test case Internal failure Test case Internsity Evaluating deliverable System testing bugs coding bugs coding test engineer	Small Low Vision Redundan Code mone of the mentione: complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note Complexity process of developing Component defects documentation impractical and Program manager
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	defect affects the functionality of the software. Seelling mistake another defects' to example of A To become a billion dollar company could be termed as: "On defect hides another defects' its termed as ? Which of the following is not included in future costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? What is prevention cost? What is prevention cost? The soft of a component application being test. matrix is used to trace the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the development of a component application being test. matrix is used to trace the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that the test of the requires the sources to be applied to the requires the sources to be software is a a document is a high level document defines work products to be tested. Ow they will tested and test type. Order testing approach to achieve testing objective. Which do the following testing testing is to defects. Which defects need to be fixed? The which defects are object? Software mistake during coding are known as Effective testing will reduce cost. Effective testing will reduce cost. Effective testing will reduce develoase the outcome in test testing. The reportish for planning and execution af the project and to censure the success of	Low Critical Mission Camouflage Effect rework, rework, envorting for the cost arises from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test Analysts Test engineer	Impact Minor Minor Goal Cascading Effect Exacading Effect help Imsupport help Imsupport Imsupport help Imsupport help Imsupport hel	Major Minor Objective Coverage Effect failure mode analysis waranty work quality hanning. The cost arises from efforts to prevent Severity Requirement engineering Big hang Regression test Test Data Test case Internal failure Test case Internal failure Test case Internsity Evaluating deliverable System testing bugs coding bugs coding test engineer Program member	Small Low Vision Redundan Code mone of the mentione: complaint resolution testing The cost arises from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note Complexity process of developing Component defects documentation impractical and Program manager Test engineer QA leader
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73	defect affects the functionality of the software. Seeling mistake another defects' to example of A To become a billion dollar company could be termed as: To hectome a billion dollar company could be termed as: "On defect hides another defects' its termed as ? Which of the following is not included in failure coulds? Which of the following is not an appraisal cost in SOA? What is prevention cost? To a defect hides as the degree of impact a defect has on the development of a component application being test. matrix is used to trace the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that the needed to verify the software test and by customer at his or her own site without the developer being is a document defines work products to be tested. Now they will tested and test type. Out of the following testing testing below they will be test to the derive. Which of the following testing testing below: Which defects are of the software testing and testafts during out and are testing below: Heat testing is In test testing will reduce cost. Effective testing will reduce cost. Effective testing will reduce develow the sources of a In test testing is in test testes of a diverse the derivets. Cost of relative control- intentent Panteure to defects. Cost of relative control- intenten	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality Mutation Alpha test Test Plan Test Plan Appraisal Test Plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test Analysts Test engineer Appriasal cost Appriasal cost	Impact Minor Minor Gaal Cascading Effect Acasading Effect Acasading Effect Biol Imsupport Minor Mi	Major Minor Minor Objective Coverage Effect failure mode analysis warnaty work quality thanitos The cost arises from efforts to prevent Sevenity Requirement cogineering Big hang Regression test Test Data Test case Internal failure Test case Internal failure Evaluating deliverable System testing bugs coding Usystem testing bugs coding test engineer Program member program member program member cost case cost Storenal failure cost	Small Low Vision Redundan Code more of the mentione: complaint resolution testing of the mentione: complaint resolution testing from the second stress Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note External failure. Test note Complexity process of developinn defects documentation impractical and Program manager Test engineer QA leader cost of control
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73	defect affects the functionality of the software. Seelling missides another defect? userupte 1 abs example of a To become a billion dollar company could be termed as: "On defect hides another defect?" is termed as ? Which of the following is not included in failure coult? "Which of the following is not included in failure coult?" Which of the following is not included in failure coult? "Which of the following is not an appraial cost in SOA? What is prevention cost? appraint of the software company. "What is prevention cost?" with a that are needed to verify whicher the requirement to the test that are needed to verify whicher the requirement to the test that are needed to verify whicher the requirement to the software tested by "counter at his not ero on site which the development and count is a type of volfware testing where is a version of the complete software tested by "counter at his not ero on site which the developer being is a document defines work products to be testion where the counter is a high evelop document defines work products to be testion which defects need to be first so defects.	Low Critical Mission Canouflage Effect rework Testine Inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Test Plan Severity Finding broken code Unit testing Ends possible QA leader Test Analysts Test angineer Apprisal cost	Impact Minor Minor Grain Cascading Effect repair Reduction of the surport minimenance The cost arcses from defects Product Requirement Traceability Decision Table Beta test Test Design Test document Prevention Test Strategy Priority A stage of all Integration testing failures docisin Test analyst Internal failures	Major Major Amjor	Small Law Vision Redundant Code none of the mentiones complaint resolution testing The cost arises from efforts to implement Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note External failure. Complexity process of developing Component defects decumentation impractical and Program manager Test engineer QA leader cost of control
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73	defect affects the functionality of the software. Seeling mistake another defects' to scramele of A To become a billion dollar company could be termed as: To defect hides another defects' to stermed as ? Which of the following is not included in future costs? Which of the following is not anaptraisal cost in SOA? Which of the following is not anaptraisal cost in SOA? Which of the following is not anaptraisal cost in SOA? Which of the following is not anaptraisal cost in SOA? What is prevention cost? is define as the degree of impact a defect has on the development of a component application being test.	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality Mutation Alpha test Test Plan Test Plan Appraisal Test Plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test Analysts Test engineer Appriasal cost Appriasal cost	Impact Minor Minor Gradi Cascading Effect repair Relp line surport minimenance The cost arcses from defects Product Requirement Taccability Decision Table Beta test Test Design Test document Prevention Test Strategy Priority A stage of all Integration testing Fast Market Priority A stage of all Integration testing failures design practically Test analyst Internal failure Integral failures Integral failures Integ	Major Minor Minor Objective Coverage Effect failure mode analysis warnaty work quality thanitos The cost arises from efforts to prevent Sevenity Requirement cogineering Big hang Regression test Test Data Test case Internal failure Test case Internal failure Evaluating deliverable System testing bugs coding Usystem testing bugs coding test engineer Program member program member program member cost case cost setternal failure cost	Small Low Vision Redundan Code more of the mentione: complaint resolution testing of the mentione: complaint resolution testing from the second stress Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note External failure. Test note Complexity process of developinn defects documentation impractical and Program manager Test engineer QA leader cost of control
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 70 71 71 72 73 74 75 57 6	defect affects the functionality of the software. Seeling missike another defects' to example of A To become a billion dollar company could be termed as: To become a billion dollar company could be termed as: "On defect hides another defects' its termed as ? Which of the following is not included in failure costs? Which of the following is not an appraisal cost in SOA? What is prevention cost? The software issue of the software testing the software issue of the following is not an appraisal cost in SOA? What is prevention cost? The software testing is a type of forware testing where we change certain statements in the source code and check if is a version of the company taken the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that are needed to verify whether the requirement to the test that one her own site without the developer being is insignly the input values to be passed to system under test. — cost arises from efforts to defects. A	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality Mutation Alpha test Test Plan Test Plan Test Plan Appraisal Test Plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test Analysts Test engineer Appriasal cost Appriasal cost Appriasal cost Appriasal cost Test engineer Severity Test engineer Appriasal cost Appriasal cost Test policy Verified Version	Impact Minor Minor Goal Cascading Effect Acascading Effect Cascading Effect Product Requirement Traceability Decision Table Beta test Test document Test document Test document Test document Prevention Test Strategy Priority A stage of all Integration testing faitures design practicallys Test analyst Cast analys	Major Major Alian	Small Low Vision Redundan Code more of the mentione: complaint resolution testing The counsists from efforts to implement Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note External failure. Complexity process of developinn defects documentation impractical and Program manager Test engineer QA leader cost of control cost of control cost of control test data
48 49 50 51 52 53 54 55 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 71 72 73 74	defect affects the functionality of the software. Selfing mistake another defects' tax example of a To become a billion dollar company could be termed as: To hection the some of the second termed as: To hection the some of the second termed as: To hection the some of the second termed as: To hection the some of the second termed as: Which of the following is not an appraival cost in SOA? What sing the following is not an appraival cost in SOA? What is prevention cost? To the second term of the second term of the following is not an appraival cost in SOA? What is prevention cost? if the second term of the second term	Low Critical Mission Canouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality management Mutation Alpha test Test Plan Test Plan Test Plan Test Plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test Analysts Test engineer Apprisal cost Apprisal cost Apprisal cost Apprisal cost Apprisal cost Test plan test plan test plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test analysts Test engineer Apprisal cost Apprisal cost Apprisal cost Apprisal cost Severity Verified Version Ouick Design	Impact Minor Minor Gaal Cascading Effect Performance Major Cascading Effect Performance International International Requirement Requirement Interactivity Product Test Design Test document Prevention Test Strategy Priority A stage of all Integration testing failures design practically Test analyst Internal failure Test plan Integration test analyst Internal failure Internal failure Test plan Version	Major Major Amjor	Small Low Vision Redundant Code none of the mentiones complaint resolution testing The cost arises from efforts to implement Project Quality Management Boundary value System Testing Test Cases Test noite External failure. Test noite External failure. Test noite Complexity process of developins Component defects documentation impractical and Program manager Test engineer QA leader cost of control test data Version Verification
48 49 50 51 52 53 54 55 57 58 59 60 61 62 63 64 65 66 66 67 70 71 72 73 74 75 57 76 76 77 77	defect affects the functionality of the software. Seelling missides another defect? seamele of a To become a billion dollar company could be termed as: "One defect hides another defect" is termed as ? Which of the following is not included in failure costs? "Defect hides another defect" is termed as ? Which of the following is not an appraial cost in SOA? "Which of the following is not an appraial cost in SOA? What is prevention cost?	Low Critical Mission Camouflage Effect rework Testing inter-process The cost arises from efforts to prevent Quality Total quality Mutation Alpha test Test Plan Test Plan Test Plan Appraisal Test Plan Severity Finding broken code Unit testing errors maintenance always possible QA leader Test Analysts Test engineer Appriasal cost Appriasal cost Appriasal cost Appriasal cost Test engineer Severity Test engineer Appriasal cost Appriasal cost Test policy Verified Version	Impact Minor Minor Goal Cascading Effect Acascading Effect Cascading Effect Product Requirement Traceability Decision Table Beta test Test document Test document Test document Test document Prevention Test Strategy Priority A stage of all Integration testing faitures design practicallys Test analyst Cast analys	Major Major Alian	Small Low Vision Redmant Code more of the mentiones: complaint resolution testing the cost arises from The cost arises from The cost arises from Process Project Quality Management Boundary value System Testing Test Cases Test note External failure. Test note External failure. Test note External failure. Test note Complexity process of developing component in defects documentation impractical and Program manager Test engineer QA leader cost of control cost of control test data

81					
82	Which two models doesn't allow defining requirements early Agile Software Development is based on	Waterfall & RAD Incremental	Prototyping & Iterative	Prototyping & RAD Linear Development	Waterfall & Spiral Both Incremental and
83		Uses incremental Different testing	Only essential Testing is	Eliminate the use of Testing and	It is based on iterative None of the mentioned
	characteristics?	techniques are	conducted by the	debugging are	
85 86	Test cases should uncover errors like In which testing level the focus is on customer usage?	Nonexistent loop Alpha Testing	Comparison of Beta Testing	Incorrect logical Validation Testing	All of the mentioned both alpha and beta
87	What is the goal of the requirements analysis and specifications phase of software develoment life cycle?	understanding the customer	Analyzing the cost of	Determine scope of the software	designing model
	Software tester focuses more on complex part of the	Testing shows	Pesticide	Testing is context	Defect Clustering
88 89		presence of defects Retesting	paradox Sanity testing	dependent Breadth test and depth	Confirmation testing
	Following are fundamental test processes arranged randomly. What will be the logical sequential flow of these	5,4,2,1,3	5,2,3,4,1	5,4,2,3,1	5,2,4,3,1
	activities.1. Test Closure activity				
	2. Implementation and execution 3. Evaluating exit criteria and Reporting				
	4. Analysis and Design				
	5. Planning and Control Exit criteria is determined during	Planning and	Implementation	Evaluating exit	Analysis and Design
	What is pesticide paradox?	If the same tests are repeated over and	A small number of modules	Testing can show that defects are present,	Finding and fixing defects does not help i
92		over again ,	contain most of	but can not prove that	the system built is
93 94		During testing Black box	During White Box	During review yellow box	Throughout life-cycle Green box
	Test cases are designed during which of the following In the spiral model 'risk analysis' is performed	Test recording In the first loop	Test In every loop	Test planning Before using the	Test specification in first and second loo
97	Identify the disadvantage of the Spiral Model.	Doesn't work well	High amount of	Strong approval and	Earlier involvement of
98	The Prototyping model of software development is	A reasonable approach when a	A useful approach when a	The best approach to use for project with	Arisky model that rarely produces a
	What are the four framework activities found in the Extreme	analysis, design	planning, analysis	planning, analysis, codi	planning, design, coding
99 100		coding, testing Good Programmer	design,coding Reliable	ng,testing Attention to details	testing Being diplomatic
101	What will be the next-date value for mm-dd-yy format for 07	08-31-2020 Decision Table	08/01/2020 Baundana Vialua	07-32-2020	07/08/2021
102			Boundary Value Analysis	Equivalence Partition	Decision Coverage
	Which of the following is/are White box technique? Boundary value analysis belong to?	Statement Testing White Box Testing	Boundary Value Black Box	Error Guessing Grey Box Testing	Equivalence Red Box Testing
	A city field in software accepts 3 to 25 alpha characters	2, 3, 25, 26	1, 3, 25, 27	2, 3, 25, 27	1, 3, 25, 27
105 106		Normal BVT	Robust BVT	Normal-Robust BVT	Worst-case BVT
	The independent considerations that apply to input domain	validity of inputs	Validity of	Number of faults	Errors
108		(a,b]	[a,b]	(a,b)	[a,b)
109	Which of the following is a fourth generation strongly typed The basic idea to test normal boundary values are to check	Ada minimum boundary	C nominal	Cobol maximum boundary	Fortran max+100
	the given options except				
		temperature Weak -Robust	pressure Strong-Robust	air Weak Normal	commision problem Normal-Normal
	What will be the next-date value for mm-dd-yy format for 07-	08-31-2020	08/01/2020	07-32-2020	07/08/2021
114	In a decision-table for triangles there areways of creating an isoseles triangle	one	two	three	four
	Cause and effect graphs deploy the following discrete Which second generation language uses statement labels to	AND FortranII	OR C	NOT Ada	Nor Pascal
116	refer to target paths for d-d graph				
117	It is acceptable to create DD-Path graph for source code In a weakly -coupled triplet	50 lines one affects second	100 lines One affects both	200 lines One does not affect	500 lines All affect each other
	AND conditions are coupled OR conditions	Strongly, strongly	strongly, weakly	weakly, weakly	weakly,strongly
119	are coupled The mathematical notion of a "basis" has attractive	functional	non-functional	structural	non-structural
	possibilities for testing McCabe Complexity can be calculated from the formula:	v-n+2	v-n+3	v-n+4	v-n+5
	A graph has 2 IF-conditions; what is itsMcCabe complexity?	1	2	3	4
123	A program has 4 slices in Slice Testing, the of all gives the complete program	Intersection	Union	difference	SUM
	The number of input variables to be tested in a Normal	7	8	9	19
124	Boundary Value system is Boundary Value testing on NextDate program does not give	bad	good	poor	excellent
125		Black Box	White box	green box	yellow box
120	Equivalence partitioning is:	A black box testing	A black box	A black box testing	A white box testing
127	Code Coverage is	technique used only WhiteBox	testing technique BlackBox	technique appropriate GreyBox	technique appropriate Green Box
129	A graph with McCabe complexity=3 has how many basis	one	two	three	four
130					
130	Structured Testing includes the following except	Path Coverage sequencing	Call Coverage alteration	Loop Coverage iteration	Exit multiple exits from
131	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a		Call Coverage alteration 5,5 ⁿ		Exit
131	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a total of test cases will be generated. Robust Boundary Value Testing forces attention on	sequencing 7, 7 ⁿ Exception Handling	alteration 5,5 ⁿ Error	iteration 7,7 Input	Exit multiple exits from 5,5 Output
131 132 133	Structured Testing includes the following except Structured programming codes includes all except In Robust Work-Case Testing, there are states and a total oftest cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the	sequencing 7, 7 ⁿ	alteration 5,5 ⁿ	iteration 7,7	Exit multiple exits from 5,5
131 132 133 134	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tatal of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the strument on which states that errors will result in a testing is a hybrid of boundary value analysis and	sequencing 7, 7 ⁿ Exception Handling	alteration 5,5 ⁿ Error multiple Robust	iteration 7,7 Input double Robust Boundary	Exit multiple exits from 5,5 Output
131 132 133	Structured Texting includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a otal of test cases will be generated. Robust Boundary Value Testing forces attention on Storug Normal Equivalence Class testing is based on the setting is a bybrid of boundary value analysis and equivalence class testing.	sequencing 7, 7 ⁿ Exception Handling single	alteration 5,5 ⁿ Error multiple	iteration 7,7 Input double Robust Boundary Value	Exit multiple exits from 5,5 Output random
131 132 133 134	Structured Texting includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a path of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and equivalence Class testing. A Decision tables in which all the conditions are binary are called Entry Decision Tables.	sequencing 7, 7 ⁿ Exception Handling single Edge Extended	alteration 5,5 ⁿ Error multiple Robust Equivalence Limited	iteration 7.7 Input double Robust Boundary Value Single	Exit multiple exits from 5,5 Output random Special Value Several
131 132 133 134 135 135	Structured Texting includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a notal of test cases will be generated. Robust Boundary Value Texting forces attermion on Strong Normal Equivalence Class testing is based on the samption which states that errors will result in a Testing is a hybrid of boundary value analysis and equivalence class testing. A Decision tables in which all the conditions are binary are called trup Decision Tables is a systematic approach where the different input combinations and their corresponding system Delaviour	sequencing 7, 7 ⁿ Exception Handling single Edge Extended	alteration 5,5 ⁿ Error multiple Robust Equivalence	iteration 7,7 Input double Robust Boundary Value	Exit multiple exits from 5.5 Output random Special Value
131 132 133 134 135 136	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tail of test case will be generated. Robust Boundary Value Testing in testing in a sumption which states that errors will result in a testing in the states that errors will result in a testing in the states that errors will result in a testing in the states that errors will result in a testing in the states that errors will result in a testing in the states that errors will result in a testing in the conditions are binary are called in a systematic approach where the different input combinations and their corresponding system behaviour (control include) in a tabilart form.	sequencing 7, 7 ⁿ Exception Handling single Edge Extended Cause Effect Graph	alteration 5,5 ⁶ Error multiple Robust Equivalence Limited Program Graph	iteration 7,7 Input double Robust Boundary Value Single Equivalence Class	Exit multiple exits from 5,5 Output random Special Value Several Decision Table
131 132 133 134 135 136	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tail of test case will be generated. Robust Boundary Value Testing increases that in the structure of	sequencing 7. 7° Exception Handling single Edge Extended Cause Effect Graph Edge	Alteration 5.5° Error multiple Robust Equivalence Limited Program Graph Chain	iteration 7.7 Input double Robust Boundary Value Single Equivalence Class Program	Exit multiple exits from 5.5 Output random Special Value Several Decision Table
131 132 133 134 135 136 136	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a lutal of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is hased on the testing is a hybrid of boundary value analysis and testing is a hybrid of boundary value analysis and casivalence Class testing is have a binary are A Decision tables in which all the conditions are binary are called trup Decision Tables. testing is a systemic thetwicour (output) are captured in a tablear form, coverage is when executed on the program, every node in the program Graph is traversed.	sequencing 7, 7 ⁿ Exception Handling single Edge Extended Cause Effect Graph	alteration 5,5 ⁶ Error multiple Robust Equivalence Limited Program Graph	iteration 7,7 Input double Robust Boundary Value Single Equivalence Class	Exit multiple exits from 5,5 Output random Special Value Several Decision Table
131 132 133 134 135 136 137 138 139	Structured Testing includes the following except. Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a tudal of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is hased on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and canivalence class testing. A Decision tables in which all the conditions are binary are allold furty Decision Tables. coverage is when executed on the program, every inde in the program Graph is traversed. Given a set of test cases for a program constitute if, when executed on the program, every solves the program constitute if, when executed on the program, every solves the solves of the program, every solves the solves of the program, every solves the program constitute if, when executed on the program, every solves the solves of the program, every solves the program constitute if, when executed on the program, every solves the solves of the program, every solves the program constitute the integram constitute if when executed on the program, every solves the program constitute the integram constitute if when executed on the program, every solves the program constitute the solves that the top the program, that the program is that the top the program is the top the program is that the top the program is the top the p	sequencing 7. 7° Exception Handling single Edge Extended Cause Effect Graph Edge	Alteration 5.5° Error multiple Robust Equivalence Limited Program Graph Chain	iteration 7.7 Input double Robust Boundary Value Single Equivalence Class Program	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Node
131 132 133 134 135 136 136 137 138 139	Structured Textine includes the following except. Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a tudal of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is hased on the testing is a hybrid of boundary value analysis and centivalence class testing. A Decision tables in which all the conditions are binary are allold term y Decision Tables. coverage approach where the different input combinations and their corresponding system behaviour (instut) are captured in a tablaal form. coverage is when executed on the program, every nde in the program Graph is traversed. Given a sof of test cases for a program constitute is discussed on the program, every sites S(x), arefer to statement fragments that contribute to the value of vat statement n. are DD-put graph is twolf-test.	sequencing 7, 7° Exception Handling single Edge Extended Cause Effect Graph Edge Edge Edge Coverage	Alteration 5.5° Error multiple Robust Equivalence Limited Program Graph Chain Path Coverage	iteration 7,7 Input double Robust Boundary Value Single Equivalence Class Program Chain Coverage	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Node Node Coverage
131 132 133 134 135 136 137 138 139 140	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a tail ad test cases will be generated. Robust Worst-Case Testing forces attention on Storeg Normal Equivalence Cass testing is based on thestates in a bary of the Monther value analysis and equivalence class testing. A Decision tables in which all the conditions are binary are calledin a systematic approach where the different input combinations and their corresponding system behaviour (contequit) are captured in a tablad form.	sequencing 7, 7 ^s Exception Handling single Edge Extended Cause Effect Graph Edge Edge Forward	alteration 5.5° Error multiple Robust <u>Equivalence</u> Limited Program Graph Chain Path Coverage Backward	iteration 7,7 Input Input Arobast Boundary Value Single Equivalence Class Program Chain Coverage Upward	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Node Node Coverage Downward
131 132 133 134 135 136 137 138 139 140 141	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tail ad test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is in which all the conditions are binary are called Entry Decision Tables.	sequencing 7, 7 ⁴ Exception Handling single Edge Extended Cause Effect Graph Edge Coverage Forward path.nodes graph coverage	alteration 5,5 st Error multiple Robust Robust Robust Robust Robust Limited Chain Path Coverage Backward nodes,edges Edge coverage	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward edges.graph Node coverage	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage
131 132 133 134 135 136 136 137 138 139 140 141 142	Sinucured Testinia includes the following except. Sinucured programming codes includes all except In Robust Worst-Case Testing, there are states and a tital of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and eunivalence class testing testing the conditions are binary are called Entry Decision Tables coverage is not tabular form coverage is when executed on the program, every nde in the program Graph is traversed. Green as of the cases for a program constitute the directed graph in which are DD- paths on its program graph, ad represent control flow between accessor DD-paths. When all the nodes are travesered in program them is successfor all program then such that her nores. When all the nodes are travesered in program then such that be such as the direct agraph is may such as the directed graph in which are DD- paths of its program graph, ad represent control More has the force coverage for accessing a set of niles such	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Edge Coverage Forward path,nodes	alteration 5.5° Erore multiple Robust Equivalence Limited Program Graph Path Coverage Backward nodes,edges Edge coverage condition	iteration 7.7 Annot double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward edges.graph	Exit multiple exits from multiple exits from 5.5 Output random Several Value Several Decision Table Node Node Coverage Downward nodes,graph
131 132 133 134 135 136 137 138 139 140 141 142 143 144	Structured Togramming codes includes all except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a tudal of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and enuivalence class testing. bestion tables in which all the conditions are binary are called <u>Entry Decision Tables</u> . testing the induction of the program, every inde in the program Graph is traversed. Given a set of est cases for a program constitute the directed graph in which are DD- paths of its program graph, ad represent control flow between accessor DD-paging not be directed program, the error When all the nodes are travesered in program program that When all the nodes are travesered in program program, then the When all the nodes are travesered in program graph, then its When all the nodes are travesered in program graph then its When all the nodes are travesered in program graph then its sub- that each condition is evaluated to both true and false? Statement and Branch coverage on testing is a spirated for the sub- that case condition is evaluated to both true and false?	sequencing 7, 7 ^{*7} Exception Handling single Edge Estended Cause Effect Graph Edge Edge Coverage Forward path.nodes graph coverage Decession coverage Analysis Model	alteration 5,5% Eror multiple Robust Limited Program Graph Chain Path Coverage Backward nodes,edges Edge coverage condition coverage	iteration 7.7 Annut double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward edges.graph Node coverage Statement coverage	Exit multiple exits from multiple exits from 5.5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes.graph path coverage Source Code
131 132 133 134 135 136 137 138 139 140 141 142 143 144	Structured Togramming codes includes all except Structured programming codes includes all except In Robust Worst-Case Testing, there are states and a tudal of test cases will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and enuivalence class testing. bestion tables in which all the conditions are binary are called <u>Entry Decision Tables</u> . testing the induction of the program, every inde in the program Graph is traversed. Given a set of est cases for a program constitute the directed graph in which are DD- paths of its program graph, ad represent control flow between accessor DD-paging not be directed program, the error When all the nodes are travesered in program program that When all the nodes are travesered in program program, then the When all the nodes are travesered in program graph, then its When all the nodes are travesered in program graph then its When all the nodes are travesered in program graph then its sub- that each condition is evaluated to both true and false? Statement and Branch coverage on testing is a spirated for the sub- that case condition is evaluated to both true and false?	sequencing 7, 7 ⁴ Exception Handling single Edge Extended Cause Effect Graph Edge Coverage Forward path.nodes <u>graph coverage</u> Decesion coverage	alteration 5,5% Eror multiple Robust Limited Program Graph Chain Path Coverage Backward nodes.edges Edge coverage condition coverage Design Model Path coverage	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward edges.graph Node coverage Statement coverage	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage path coverage
131 132 133 134 135 136 137 137 138 139 140 141 142 142 144 145	Structured Testing includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing in a hybrid of boundary value analysis and equivalence class testing. is a sybrid of boundary value analysis hocision Tables in which all the conditions are binary are called interposed where the different input combinations and their corresponding system behaviour (output) are captured in a tablat form. coverage is when executed on the program, every sites S(va) refer to statement fragments that contribute to the value of value attament for, represent control How between successor DD-paths. When all the nodes are travestered in program query Statement coverage is alknown as represent control How between successor DD-paths. When all the nodes are travestered in program reprofiles? Statement and branch coverage metrics are part of Statement or streament is a banknown as Major benefit of output and the securits on the executed.	sequencing 7, 7 ⁴ Exception Handling single Edge Extended Cause Effect Graph Edge Coverage Forward path.nodes emph coverage Decesion coverage Analysis Model Lane coverage parameter coverage	alteration 5.5% Error multiple Robust Eurivalence Eurivalence Eurivalence Eurivalence Chain Program Graph Chain Path Coverage Backward Backward Backward Backward Backward Backward Backward Backward Backward Design Adodel Pesign Adodel Pesign Adodel	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward odges.graph Node coverage Statement coverage Testing Code coverage	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Decision Table Node Coverage Downward nodes,graph path coverage path coverage Source Code Decesion coverage Loop coverage Loop coverage Loop coverage Loop coverage Loop coverage Loop coverage Loop coverage Loop coverage Loop coverage Loo
131 132 133 134 135 135 136 137 138 139 140 141 142 143 144 145 144 145	Structured Testing includes the following except. Structured programming codes includes all except. In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is in which all the conditions are binary are called Entry Decision Tables is a systematic approach where the different input combinations and their corresponding system behaviour (output) are captured in a tablatf form coverage is when executed on the program, every sites S(va) refer to statement fingments that contribute to the value of v at statement in, routput if, when executed on the program, every sites S(va) refer to statement fingments that contribute to the value of v at statement fingments that contribute to the value of v at statement fingments that Contribute to the value of v at statement fingments that contribute to the value of v at statement fingments that contribute to the value of v at statement fingments that contribute to the value of v at statement fingments that contribute to the value of v at statement fingments that contribute to the value of value value of value if, when all the nodes are travescred in program raph then it is Which in type of coverage for exercising a set of rules such When all the nodes are travescred in program raph then it is Which therefit of counce_ment is a balancom as	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Analysis Model Lune coverage	alteration 5.5% Error multiple Robust Eurivalence Drogram Graph Program Graph Chain Program Graph Chain Path Coverage Edge coverage Condition Coverage Design Model Path coverage statement coverage 2	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward Odge.graph Node coverage Statement coverage Testing Code coverage Decision coverage Decision testing 0	Exit multiple exits from 5,5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage path coverage path coverage Source Code Decesion coverage Loop coverage Functional testing 3
131 132 133 134 134 135 136 137 138 138 139 140 141 142 144 144 144 144 144 144 144 144	Structured Togramming codes includes all except Structured rogramming codes includes all except In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and enuivalence class testing. is a hybrid of boundary value analysis and testing the conditions are binary are called testing the conditions are binary are contrained is their access of a program constitute coverage is to thatbular form. coverage is the access of a program constitute testing the structured on the program, every mode in the program Graph is traversed. Dry and parts the directed graph in which are DD- paths of its program graph, and represent control flow between access of Dtesting a set of rales such that each condition is evaluated to both true and false? Statement and branch coverage for exercising a set of rales such that profile condition is evaluated to both true and false? Statement and branch coverage for exercising a set of rales such that profile condition is evaluated to both true and false? Statement and branch coverage for exercising a set of rales such that profile condition is evaluated to both true and false? Statement and branch coverage for exercising a set of rales such that profile condition is evaluated to a both true and false? Statement and branch coverage for exercising a set of rales such the portion of code, which could not be executed. Cause-offect graphing is ione form of of. The degree of an isolated mode is alwavs	sequencing 7, 7 ⁴ Exception Handling single Edge Edge Cause Effect Graph Edge Edge Coverage Forward path,nodes graph coverage Decesion coverage Analysis Model Line coverage Structural testing 1	alteration 5,5% Eror multiple Robust Entry Entry Program Graph Chain Path Coverage Backward nodes,edges Edge coverage Condition coverage Design Model Path coverage Statement coverage 2	iteration 7,7 7 7 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9	Exit multiple exits from multiple exits from multiple exits from 5.5 Output random Special Value Several Decision Table Decision Table Node Node Coverage Downward nodes.graph path coverage path coverage Source Code Decession coverage Functional testing 3 outleg(n)
131 132 133 134 135 136 136 137 137 138 139 140 141 142 143 144 145 146 147 148 148 148 149 150	Structured Togramming codes includes all except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tital of test cases will be generated. Robust Boundary Value: Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing in a hybrid of Doundary value analysis and enuivalence: class testing: a hybrid of Doundary value analysis and and their convergence of the testing is based a hybrid of Doundary value analysis and enuivalence: class testing: a hybrid of Doundary value analysis a hybrid of Doundary value analysis and their convergence there the different input combinitions and their convergence functions are binary are coverage is when caccuted on the program, every mode in the program Graph and represent control flow between assees of Dy pogram constitute the directed graph in which are DD- paths of its program graph, and represent control flow between assees of Dy Dogram graph, then direct Statement and herance coverage of tescersing a set of rules such that each condition is evaluated to both true and false? Statement and true coverage in the courting metrics are part of Statement coverage is also and the secured. Cause-effect graphing is none from of: The degreer of an isolated nood is alwavas How the direct of and hybrid in directed graph? The Decision to Decision (DD) path graph is an extension of Thes of Use which is key for class cause of offers.	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Coverage Forward path.nodes erraph coverage Decesion coverage Analysis Model Line coverage parameter coverage 1	alteration 5.5% Error multiple Robust Eurivalence Drogram Graph Program Graph Chain Program Graph Chain Path Coverage Edge coverage Condition Coverage Design Model Path coverage statement coverage 2	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward Odge.graph Node coverage Statement coverage Testing Code coverage Decision coverage Decision testing 0	Exit multiple exits from 5,5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage path coverage path coverage Source Code Decesion coverage Loop coverage Functional testing 3
131 132 133 134 135 136 137 138 139 140 141 142 144 145 144 145 146 147 148 148 149 150	Structured Testing includes the following except. Structured programming codes includes all except. In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing in a hybrid of boundary value analysis and equivalence class testing is a systematic approach where the different input combinations and their corresponding system behaviour (output) are captured in a tabular form coverage is when executed on the program, every likes S(va) refer to statement fingments that contribute to the value of values attament in, respect control How between auccessor DD-paths. When all the nodes are travestered in program apply and of rules control How between auccessor for exercising a set of rules (Satement or fingments that contribute to the value of values are travestered. Statement coverage is a block norm as respect control How between auccessor DD-paths. When all the nodes are travestered in program respit Statement coverage is a block norm as	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Decesion coverage Janalysis Model Line coverage parameter coverage Structural testing index(n) program graph Verification	alteration 5.5% Error multiple Robust Eurivalence Drogram Graph Chain Program Graph Chain Path Coverage Backward Dath Coverage Backward Rackward Backward Backward Chain	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward Chain Coverage Upward edges,draph Nade coverage Statement coverage Testing Code coverage Decision coverage Regression testing ndes(m ²) ² graph Regression	Exit multiple exits from 5,5 Output 7,5,5 Output 7,100 Special Value Several Decision Table Node Node Node Coverage Downward nodes,graph path coverage path coverage Loop coverage Loop coverage Docesion coverage Source Code Decesion Coverage Source Code Decesion Coverage Source Code Decesion Coverage Source Code Decesion Coverage Source Code Code Code Source Code Code Code Code Code Code Code Code
131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 144 145 146 147 148 149 150 151	Structured Testing includes the following except. Structured programming codes includes all except. In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing in a hybrid of bundary value analysis and equivalence class testing is a systematic approach where the different input combinations and their corresponding system behaviour (output) are captured in a tabular form coverage is when executed on the program, every then executed on the program, every sites S(va) refer to statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of value of value value discusses for D-paths. When all the nodes are travescered. Cause-effect graphing is one form of: Statement coverage is ablo known as	sequencing 7, 7 ⁴ Exception Handling single Edge Edge Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Decesion coverage parameter coverage parameter coverage Structural testing index(n) program graph Verification	alteration 5,5% Error multiple Robust Eurivalence Drogram Graph Chain Program Graph Chain Path Coverage Backward Dath Coverage Backward Backward Backward Backward Backward Backward Backward Design Andel Path coverage Statement coverage Path coverage Jach coverage Jach coverage Jackward Design Andel Path coverage Statement coverage Validation	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward Chain Coverage Upward cdge.graph Nade coverage Statement coverage Testing Code coverage Ection coverage Regression testing Ogen testing Regression	Exit multiple exits from 5.5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes.graph path coverage path coverage path coverage Docesion coverage Loop coverage Docesion coverage Loop coverage Jone coverage Source Code Decesion coverage Loop coverage Jone coverage
131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 144 145 146 147 148 149 150 151	Structured Testing includes the following except. Structured programming codes includes all except. In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing in a hybrid of bundary value analysis and equivalence class testing is a systematic approach where the different input combinations and their corresponding system behaviour (output) are captured in a tabular form coverage is when executed on the program, every then executed on the program, every sites S(va) refer to statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that Contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of v at statement fragments that contribute to the value of value of value value discusses for D-paths. When all the nodes are travescered. Cause-effect graphing is one form of: Statement coverage is ablo known as	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Decesion coverage Janalysis Model Line coverage parameter coverage Structural testing index(n) program graph Verification	alteration 5.5% Error multiple Robust Eurivalence Drogram Graph Chain Program Graph Chain Path Coverage Backward Dath Coverage Backward Rackward Backward Backward Chain	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward Chain Coverage Upward edges,draph Nade coverage Statement coverage Testing Code coverage Decision coverage Regression testing ndes(m ²) ² graph Regression	Exit multiple exits from 5,5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes.graph path coverage path coverage path coverage Loop coverage Loop coverage Jone coverage Source Code Decesion coverage Loop coverage Jone cov
131 132 133 134 135 136 137 138 139 140 141 142 144 142 144 143 144 145 146 147 151 151 151 155 154	Structured Togramming codes includes all except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tital of test cases will be generated. Robust Boundary Value: Testing forces attention on Strong Normal Equivalence: Class testing is housed on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and enuivalence class testing. as a hybrid of boundary value analysis as a hybrid of boundary value analysis and enuivalence class testing. as a systematic approach where the different input combinations and their corresponding system behaviour fortunt) are captured in a tabular form. Coverage is when executed on the program, every is systematic approach where the different input contributes to the value of v at statement in genesis that contribute to the value of v at statement in a most of Given a set of est cases for a program constitute 	sequencing 7, 7 ⁴ Exception Handling single Edge Edge Cause Effect Graph Edge Edge Coverage Forward path,nodes Enge Coverage Decesion coverage Analysis Model Line coverage Suructural testing 1 Indes(n) program graph Verification Verification	alteration 5,5% Eror multiple Robust Enotype Eron Another Program Graph Chain Path Coverage Backward nodes.edges Backward nodes.edges Backward nodes.edges Backward Nationation Design Model Path coverage Statement coverage Design Model Path coverage Statement Coverage Statement Coverage Statement Coverage Mainteence Path coverage Statement Coverage Mainteence Statement Coverage Statement Statemen	iteration 7,7 7 7 7 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9	Exit multiple exits from multiple exits from multiple exits from multiple exits from 5.5 or a constraint of the exits of t
131 132 133 134 135 136 137 138 139 140 141 142 144 144 144 145 146 147 148 149 151 151 155	Structured Togramming codes includes all except Structured rogramming codes includes all except In Robust Worst-Case Testing, there are tates and a tital of test cases will be generated. Robust Boundary Value: Testing forces attention on Strong Normal Equivalence: Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and enuivalence: class testing. as hybrid of boundary value analysis as hybrid of boundary value analysis and enuivalence: class testing. as hybrid of boundary value analysis as hybrid of boundary value analysis 	sequencing 7, 7 ⁴ Exception Handling single Edge Edge Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward Edge Coverage Forward Decesion coverage Analysis Model Line coverage Structural testing Decesion coverage Structural testing Unification Verification	alteration 5.5% Error multiple Robust Eurivalence Limited Program Graph Chain Path Coverage Backward Dath Coverage Backward Backward Backward Backward Backward Backward Backward Backward Design Andel Path Coverage Condition Coverage Design Model Path coverage Statement Coverage Validation Validation Process rework Peor Review	iteration 7.7 7.7 Notes Boundary Value Single Equivalence Class Program Chain Coverage Upward Chain Coverage Upward edges,graph Node coverage Statement coverage Testing Code coverage Decision coverage Regression testing Regression Regression Regression	Exit multiple exits from multiple exits from multiple exits from from 5.5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes.graph path coverage path coverage Source Code Decesion coverage Functional testing 3 outleg(n) path Coverage Confirmation Confirmation Validation Process
131 132 133 133 134 135 135 136 137 138 139 140 141 142 142 142 144 144 144 144 147 155	Structured Technic includes the following except. Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tial of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing in a hybrid of boundary value analysis and equivalence class testing is a hybrid of boundary value analysis and equivalence class testing is a hybrid of boundary value analysis and equivalence class testing is a hybrid of boundary value analysis and equivalence class testing is a hybrid of boundary value analysis and equivalence class testing is a hybrid of boundary value analysis and equivalence class testing is a hybrid of boundary value analysis is a systematic approach where the different input combinations and their corresponding system behaviour (context) are captured in a tabular form coverage is when executed on the program, every slices S(va) refer to statement fngments that contribute to the value of v at statement fn. DD-path of its program graph, and represent control flow between successor DD-paths. When all the nodes are travesered in program result the portion of code, which could not be executed. Cause-effect graphing is one form cit. The degree of node is calculated in directed graph' The Decision DD paths. How the degree of node is alvavas. How the degree of node is alvavas. How the degree of node is alvavas. Conformance to requirement which is developer view of anality can also be termed as an of	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Decesion coverage Janalysis Model Line coverage parameter coverage Structural testing index(n) program graph Verification Verification Verification Verification Verification Self Review Predelivery Audit	alteration 5.5% Error multiple Robust Eurivalence Limited Program Graph Chain Path Coverage Backward Dath Coverage Backward Backward Backward Backward Backward Backward Backward Design Andel Path Coverage Condition Coverage Design Model Path coverage Statement Coverage Validation Validation Process rework Peor Review Phase End Audit	iteration 7,7 Inout Inout Activation Comparison Inou Inou Inou Inou Inou Inou Inou In	Exit multiple exits from 5,5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage path coverage path coverage Source Code Decession coverage Loop coverage Loop coverage Punctional testing 3 outdeg(n) path Confirmation Confirmation Confirmation Putaktrough Walkthrough Product Andit
131 132 133 133 134 135 135 136 137 138 139 140 141 142 142 142 144 144 144 144 147 155	Structured Technic includes the following except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tail of test case will be generated. Robust Boundary Value Testing forces attention on Strong Normal Equivalence Class testing is based on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and equivalence class testing. is a sybrid of boundary value analysis and equivalence class testing. is a sybrid of boundary value analysis A Decision tables in which all the conditions are binary are called into the conditions are binary are called into the conditions are binary are conditions and their corresponding system behaviour (control and their corresponding system behaviour (control and their corresponding system behaviour (control and control in a tablat form. overage is when executed on the program, every sites S(va) refer to statement fingments that contribute to the valued of vat statement fingments that control control to the strong mergin. Major brenft of is that it is greatly able to isolate the portion of cocles, which could not be executed. Cause-effect graphing is one form of: The degree of node is calculated in directed graph? How the degree of node is calculated in directed graph? How the degree of node is calculated in directed graph? The Decision DDJ paths. How the degree of node is calculated in directed graph? The beckering to the cody for the strong ord Finess of Use which is defined as the customing of Finess of Use which is defined as the customent of Finess of Use which is developer view of cality can able termed as an official type of review in most of the software verification processes is	sequencing 7, 7 ⁴ Exception Handling single Edge Edge Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Decesion coverage Analysis Model Line coverage parameter coverage Structural testing index(n) program graph Verification Verification Verification Verification Verification Self Review Predelivery Audit Audit	alteration 5.5% Error multiple Robust Limited Program Graph Chuin Path Coverage Backward Dath Coverage Backward Backward Backward Backward Backward Backward Backward Design Andel Path Coverage Condition Coverage Design Model Path coverage Usign Model Path coverage Usign Model Path coverage Usign Model Path coverage Coverage Design Model Path coverage Coverage Design Model Path coverage Coverage Design Model Path coverage Path coverage Coverage Path coverage Path coverage Path coverage Path coverage Path coverage Path coverage Path coverage Coverage Path coverage Path co	iteration 7,7 Inout double Robust Boundary Value Single Equivalence Class Program Chain Coverage Upward Chain Coverage Upward cdges.graph Node coverage Statement coverage Testing Code coverage Eccision coverage Decision coverage Decision coverage Regression testing Od no '2' graph Regression Standards Inspection Inspection Periodic Audit Inspection	Exit multiple exits from 5,5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage path coverage path coverage Docession coverage Loop coverage Loop coverage Functional testing 3 outdeg(n) path Confirmation Confirmation Confirmation Validation Process Walkthrough Walkthrough Smarte Code Decessing Surger Code Decessing Surger Code Decessing Surger Code Decessing Surger Surger
131 132 133 134 135 136 137 137 138 139 140 141 142 142 144 144 144 144 145 155 156 156	Structured Togramming codes includes all except Structured programming codes includes all except In Robust Worst-Case Testing, there are tates and a tital of test cases will be generated. Robust Boundary Value Testing Incress attention on Strong Normal Equivalence Class testing is hosted on the assumption which states that errors will result in a Testing is a hybrid of boundary value analysis and enuivalence class testing. a hybrid of boundary value analysis and assumption which states that errors will result in a assumption which states that errors will result in a assumption where the different input combinations and their corresponding system behaviour forum a set of test cases for a program constitute Given a set of test cases for a program constitute 	sequencing 7, 7 ⁴ Exception Handling single Edge Estended Cause Effect Graph Edge Cause Effect Graph Edge Coverage Forward path.nodes graph coverage Decesion coverage Decesion coverage Janalysis Model Line coverage parameter coverage Structural testing index(n) program graph Verification Verification Verification Verification Verification Self Review Predelivery Audit	alteration 5.5% Error multiple Robust Eurivalence Limited Program Graph Chain Path Coverage Backward Dath Coverage Backward Backward Backward Backward Backward Backward Backward Design Andel Path Coverage Condition Coverage Design Model Path coverage Statement Coverage Validation Validation Process rework Peor Review Phase End Audit	iteration 7,7 Inout Inout Activation Comparison Inou Inou Inou Inou Inou Inou Inou In	Exit multiple exits from 5,5 Output random Special Value Several Decision Table Node Node Coverage Downward nodes,graph path coverage path coverage path coverage Source Code Decession coverage Loop coverage Loop coverage Punctional testing 3 outdeg(n) path Confirmation Confirmation Confirmation Putaktrough Walkthrough Product Andit

160	decides the execution of the inspection, defines the schedules, allocates time and defines objectives of	Manager	Moderator	Author	Reviewer
	Name the step that is precedor to the step 'follow up'	Decision on	Inspection	Planning for	Individual prepartion
	name the audit that checks whether the phase defined in the SDLC model achieves it outcome or not	Predelivery Audit	Phase End Audit	Periodic Audit	Product Audit
	This is the person who prepares the artifact for inspection One of the following is not a characteristic of nice domain	Scribe orthogonal	Moderator complete	Author linear	Reviewer inconsistent
165	testing invoves testing of software with software environmental factors like database , operating system, where the application is supported to work	Interface testing	Integration testing	System Testing	Unit Testing
	system , where the application is supposed to work. Name the testing that involves testing of many units by	Interface testing	Integration	System Testing	Unit Testing
166	combing them together to form a module or sub module. matrix starts with the requirements as stated in the	Traceability	testing Testing	Specification	Execution
167	requirement specification and goes forward upto test results. This si a testing program based on specification like	Feature Coverage	Specification	Functionality	Integration Testing
	requirement specification, design specification, user manual		Based Testing	Coverage	
170	One of the folowing is not included in levels of validation. testing involves stubs and drivers in the	Review Review	Unit testing Unit testing	Integration Testing Integration Testing	Acceptance testing Acceptance testing
171	This testing is termed as dynamic testing. Name the technique used to find heavily used path from other path present the application where the control goes	valication testing Path Sensitizing	verification Path Profiling	requirement testing Path Testing	stress testing Path reading
173	In this stratergy we find a defect or a bug that go through the the testing done to find whher the application is alive or not	Path Sensitizing Interface testing	Path Profiling Integration testing	Path Testing System Testing	Path reading Smoke Testing
175	testing is done when the developemnt	Updation testing	Installation	Pre-requisite testing	Unstallation testing
176	organisation wishes to check that the uninstallation is clean testing is an onsite acceptance test.	Beta	testing Gamma	Alpha	Unit
177	testing is also known as field testing. is an offsite acceptance test.	Beta Alpha	Gamma Gamma	Alpha Beta	Unit Unit
179	Testing is carried out at clients site by the end users or the stake holders of the product.	Beta	Gamma	Alpha	Unit
	Testing is carried out at the end of software	Beta	Gamma	Alpha	Unit
180 181	deveopment process and before handing over the software to Find which of the following are the types of acceptance	Alpha Testing	Beta Testing	Both Alpha and Beta	Unit testing
182	Acceptance testing is also referred as testing.	Blue Box Verification	Red Box VV	Grey Box Waterfall	White Box validation
183	associated with different phases of software development. Program level designs are associated with	Unit			
184	At the code level to validate individual units	Unit	Componant Componant	Integration Integration	Acceptance Acceptance
186	Design phase testing is associated with testing which covers design specification as well as structural	Unit	interface	integration	acceptance.
187	Which of the following in testing is not the characteristic of Good Requirement.	Adequate	Clear	Verifiable	Complex
	should be consistent with the application	Test Case	Test Scenario	Test Plan	Test Document
	development methodology,schedules and deliverables.	Test Lead	Project Manager	Test Manager	Moderator
189 190	strategies,define schedules and methods of testing. is responsible for defining test strategies in	Test Lead	Project Manager	Test Manager	Moderator
191	is the most vital stage in software development where product is actually built.	Designing	coding	testing	Maintainance
	describes verification and validation activities	RAD Model	Incremetal	VV Model	Validation Model
	The project team along with architests and designers may walk through the design to find the completeness and give	Design Validation	Design Verification	Design Implementation	Prioritorisation
193	comments, if any this process is called as is carried out through the inspection of	requiement	Design	Design Validation	Requirement
194	requirement specification. review helps in identification of errors with	Verification Requirement	Verification Code Review	Design Review	Validation. Team Review
195	respect to indenting, commenting and coding standards.	Review		Customers	Testers
196	is the final user group or people who are actually sponsoring the project.	Developers	Managers		
197	testers as per the scope of testing.	Development	Analysis	Testing	Maintainance
198	is a technique used for constructing the program structure while at the same time carrying out tests	Integration Testing	Interface Testing	System Testing	Unit Testing
				Senter Testing	
199	The testing that is done to verify the interface functionality is called	Integration Testing	Interface Testing	System Testing	Unit Testing
199 200	called is not involved in System Testing.	Developers	Designer	User	Testers
200	called is not involved in System Testing. is conducted on whole integrated system to estimate the systems compliance with its specified set of	Developers Integration Testing	Designer Interface Testing	User System Testing	Testers Unit Testing
200	calledis not involved in System Testing. is conducted on whole integrated system to estimate the systems compliance with its specified set of is largely a white box oriented. Hybrid Integration Testing is also known as	Developers	Designer Interface Testing Integration Sandwich	User	Testers
200 201 202 203	called is not involved in System Testing, is conducted on whole integrated system to estimate the systems compliance with its specified set of is largely a while box oriented. Hybrid Integration Testing is also known as Which testing is concerned with behavior of whole product	Developers Integration Testing Module testing	Designer Interface Testing Integration Sandwich Component	User System Testing Design testing	Testers Unit Testing Proposal testing
200 201 202	called is not involved in System Testing, is conducted on whole integrated system to estimate the systems compliance with its specified set of sis largely a white box oriented. Hybrid Integration Testing is also known as Which testing is concerned with behavior of whole product as per specified requirements? System architecture is determined during which phase?	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement	Designer Interface Testing Integration Sandwich Component testing Implementation	User System Testing Design testing Bottom-Up System testing Development	Testers Unit Testing Proposal testing Integration Testing Integration testing Design
200 201 202 203 204 204 205 206	called is not involved in System Testing. is conducted on whole integrated system to situate the systems compliance with its specified set of is largely a white box oriented. Hybrid Integration Testing is also known as which testing is concerned with behavior of whole product as per specified requirement? System architecture is determined during which phase? Verifying that whether software components are functioning corrective and identifying the defects in them is obsective of	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement Integration testing	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing	User System Testing Design testing Bottom-Up System testing Development Unit testing	Testers Unit Testing Proposal testing Integration Testing Integration testing Design System Testing
200 201 202 203 204 205 206 207	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement Integration testing Alpha Testing To verify that	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system	Testers Unit Testing Proposal testing Integration Testing Integration testing Design System Testing Unit Testing To verify that
200 201 202 203 204 204 205 206 207 208 209	called is not involved in System Testing. is conducted on whole integrated system to is conducted on whole integrated system to is largent a white box oriented. Hybrid Integration Testing is also known as. Which testing is concerned with behavior of whole product as per specified requirement? System architecture is determined during which phase? Verifying that whether software components are functioning correctly and identifying the defects in them is obsective of "Stubs" and "Drivers" are used is which type of testing? What is the objective of integration testing?	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement Integration testing Alpha Testing To verify that system is Developers	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that system meets Business	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing	Testers Unit Testing Proposal testing Integration Testing Integration testing Design System Testing Unit Testing
200 201 202 203 204 205 206 207 208	calledis on timoloed in System Testing. is conducted on whole integrated system to simate the system compliance with its specified set of is integrated with behavior of whole product as per specified requirements? System architecture is determined during which phase? Verifying that whether software components are functioning correctly and identifying the defects in them is objective of "Stubs" and Thriver" are used is which type of testing? What is the objective of integration testing? are the Testers of System Testing?	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement Integration testing Alpha Testing To verify that system is Developers Design	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that system meets	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system separately testable Independent Testers Planning	Testers Unit Testing Proposal testing Integration Testing Integration testing Design System Testing Unit Testing To verify that interfaces between Customers Release
200 201 202 203 204 205 206 207 208 209 209 210 211 212	calledis on timoloed in System Testing. is conducted on whole integrated system to simate the system compliance with its specified set of is integrated with behavior of whole product as per specified requirements? System architecture is determined during which phase? Verifying that whether software components are functioning correctly and identifying the defects in them is objective of "Stubs" and Thrivers" are used is which type of testing? What is the objective of integration testing? 	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement Integration testing Alpha Testing To verify that system is Developers Design Avoids hung coding Functionality	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that system meets Business Execution Reusability Recoverability	User System Tessing Design tessing Bottom-Up System tessing Development Unit tessing Integration Tessing To verify that system separately testable Independent Testers Planning testing becomes easier	Testers Unit Testing Proposal testing Integration Testing Integration Testing Design System Testing Unit Testing To verify that Interfaces between Customers Release None of the above
200 201 202 203 204 205 206 207 208 209 209 210 211	called	Developers Integration Testing Module testing Big-Jang Acceptance testing Requirement Integration testing Alpha Testing To verify that system is Design Avoids hurge coding Functionality Usability Testing Stress Testing	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that system meets Business Execution Recoverability Scalability Volume Testing	User System Tessing Design tessing Bottom-Up System tessing Development Unit tessing Integration Tessing To verify that system separately testable Independent Testers Planning testing becomes easier	Testers Unit Testing Proposal testing Integration Testing Integration testing Oesian System Testing Unit Testing Unit Testing To verify that interfaces between Customers Reclassifi Recoverability Testing Recoverability Testing
200 201 202 203 204 205 206 207 208 209 210 211 212 213	called is not involved in System Testing. is conducted on whole integrated system to sis conducted on whole integrated system to is largely a while box oriented. Hydral Integration Testing is also known as Which testing is concerned with behavior of whole product as per specific productions of the system. System architecture is determined during which shave? Verifying that whether software components are functioning removely and identifying the defects in them is achiective of Stubs' and "Drivers" are used is which type of testing? What is the objective of integration testing? are the Testers of System Testing? Fauls are found most cost-effectively in which test activity? What is the objective of integration studied of the studied of the system. Testing makes sure that the functionality of Testing makes sure that the functionality is Testing arms to test the limits of the System. principle of Security Testing, mans that data	Developers Integration Testing Module testing Big-Bang Acceptance testing Requirement Integration testing To verify that youth to Developers Devision Developers Design Avoids huge coding Functionality Usability Testing	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that system meets Business Execution Reusability Scalability Scalability Volume Testing Volume Testing	User System Testing Design testing Bottom-Up System testing Development Unit testing To verify that system separately testable Independent Testing Planning testing becomes easier Performance Testing GUI Testing GUI Testing Indeprive	Testers Unit Testing Integration Testing Integration Testing Integration testing Design To verify that Interfaces ketween Customers Release Release Recoverability Testing Installation Testing Installation Testing
200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217	calledis on involved in System Testingis conducted on whole integrated system to sis conducted on whole integrated system to sistance the systems compliance with its specified set of sistance the systems compliance with the specified set of setting is concerned with helawior of whole product as ner specified routinements? Which testing is concerned with helawior of whole product an ery specified routinements? Which testing is concerned with helawior testing? System architectures is determined during which theme? What is the objective of integration testing? Paulus are found most cost-effectively in which test activity? which of this is an disadvantages of small code? Testing makes sure that the functionality of Testing makes sure that the functionality of Testing makes sure that the functionality is a testing technique for writing test cases. principle of Scenity? Testing, mants that data is a testing technique for writing test cases.	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Alpha Testing To verify that yastem is Developers Design Vavids hung coding Functionality Usability Testing Stress Testing Vulnerability Installation Testing System Recovery	Designer Interface Testing Mandreich Sandwich Component testing Implementation Beta Testing Beta Testing Scalability Volume Testing Authorization Stress Testing System Design	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Independent Testers Planning Regressive Testing GUI Testing Indegrity Performance Testing GUI Testing Indegrity Decision Table System Coding	Testers Unit Testing Integration Testing Integration Testing Integration testing Design System Testing Unit Testing Unit Testing Unit Testing Release Release Release Release Release Reversability Testing Installation Testing Installation Testing Installation Testing Installation Testing System Miniainence
200 201 202 203 204 205 206 207 208 209 210 211 211 212 213 214 215 216 217 218	calledis on involved in System Testingis conducted on whole integrated system to sis conducted on whole integrated system to is itarcely a white box oriented. Hyrdi Integration Testing is also known as Which testing is concerned with behavior of whole product as ner specified routinements? Urifying flat whether software components are functioning correctly and identifying the defects in them is objective of "Stubs" and Drivers' are used is which type of testing? Fulls are found most cost effectively in which test cost are the objective of integration testing? Fulls are found most cost effectively in which test citizity? Testing makes sure that by spents' anality of Testing makes sure that by spents' astaldation is a testing technique for writing test cases. Recovery Testing comprises of testing techsif if by system is capable of handling Testing comprises of Testing comprises of the system. Recovery Testing, we give same in tegration to full for the system's capable of handling Testing comprises of testing techs if the system is capable of handling Testing integration testings in the system is capable of handling Testing integration testing test	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing To verify that system is Developers Design Avoids hung coding Functionality Testing Stress Testing Vulnerability Installation Testing	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Beta Testing To verify that systemses Execution Recoverability Recoverability Recoverability Recoverability Authorization Stress Testing	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Segaratiely testable Independent Testers Planning Regressive Testing GUI Testing Cull Testing Decision Table	Testers Unit Testing Proposal lesting Integration Testing Integration Testing Design System Testing Unit Testing To verify that interfaces between Cutstomers Reclassifi Recoverability Testing Bealability Testing Denial of Service Denial of Service
200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Alpha Testing To verify that system is Developers Design Avoids hung coding Functionality Testing Stress Testing Vulnerability Testing System Recovery Unit Testing	Designer Interface Testing Interration Sandwich Component testing Implementation Acceptance testing Bawiness Saiden neets Execution Recoretability Recoretability Recoretability Recoretability Recoretability Steam Festing System Design Integration Parallel Testing Common	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Segaratiely testable Independent Testers Planning testing becomes cashing Regressive Testing GUI Testing Decision Table System Coding	Testers Unit Testing Proposal testing Integration Testing Integration Testing Desian System Testing Unit Testing To verify that interfaces between Customers Recoverbility Testing Recoverbility Testing Denial of Service Integration testing System Maintainence
200 201 202 203 204 206 206 207 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221	calledis on timoloed in System Testing. is conducted on whole integrated system to sistmate the systems compliance with its specified set of is such as the system is specified set of is specified set of the system is specified set of is specified set of the system is specified set of is specified set of the system is specified set of is specified set of the system is specified set of the system is specified set of the system is specified set of is specified set of the system is specified set of is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system is specified set of the system 	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Integration testing Developers Design Avoids hune coding Functionality Testing System Georgers Design Coding System Recovery Unabrability Testing Operations Testing Operations Testing Operations Testing Operations Testing Operations Testing Capability Maturity Fire	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Business Execution Business Execution Business Execution Recusability Scalability Scalability Scalability Stress Testing System Design Integration Parallel Testing Common Two	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system separately vestable Independent Testers Planning testing becomes cashing Regressive Testing GUI Testing Old UT esting System Coding System Coding Sanoke Testing Sanoke Testing Capability. Three	Testers Unit Testing Proposal testing Integration Testing Integration Testing Desian System Testing Unit Testing To verify that interfaces between Customers Recoverbility Testing Recoverbility Testing Denial of Service Denial of Service Integration testing System Maintainence Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing
200 201 202 203 203 204 205 207 200 200 200 200 200 200 200 200 200	calledis on timoloed in System Testing. is conducted on whole integrated system to simate the systems compliance with its specified set of is some compliance with its specified set of is some complexed with behavior of whole product as per specified requirements? System architecture is determined during which phase? Verifying that whether software components are functioning correctly and identifying the defects in them is objective of is some components are functioning correctly and identifying the defects in them is objective of are used is which type of testing? What is the objective of integration testing? 	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Integration testing Developers Design Developers Design Avoids hung coding Functionality Testing System Recovery Unistrability Testing Operations Testing Operations Testing Operations Testing Operations Testing Operations Testing Operations Testing Capability Maturity Fire Adhoc	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Business Execution Business Execution Recusability Scalability Scalability Scalability Scalability Scalability Scalability Scalability Stress Testing Parallel Testing Common Two Managed RDBMS	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system segnarately restable Independent Testers Planning testing becomes castle Regressive Testing GUI Testing Columnation System Coding System Coding Sanoke Testing Sanoke Testing Capability Three Defined Assembly Language	Testers Unit Testing Proposal lesting Integration Testing Integration testing Design System Testing Unit Testing To verify that interfaces between Customers Recoverbally Testing Recoverbally Testing Recoverbally Testing Installation Testing
200 201 202 202 203 203 203 204 205 206 207 208 209 200 210 211 212 213 213 214 214 215 212 213 214 214 215 216 202 202 202 202 202 202 202 202 203 203	called is on timoloed in System Testing	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Alpha Testing To verify that system is Developers Design Avoids hung coding Functionality Testing System Recovery Unbrathility Testing Operations Testing System Recovery Unit Testing Operations Testing Operations Testing Operations Testing Capability Maturity Five Adhoc Conventional Increases	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Results Business Execution Business Execution Business Execution Resultivity Scalability Scalability Scalability Scalability Scalability Stress Testing System Design Integration Parallel Testing Common Two Managed RDBMS Decreases Interstate	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system segnarately testable Independent Testers Planning Regressive Testing GUI Testing GUI Testing Call Testing System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language	Testers Unit Testing Proposal testing Integration Testing Integration Testing Design System Testing Unit Testing To verify that interfaces between Customers Recoverability Testing Installation Testing Denial of Service Denial of Service Installation Testing Installation Common Modification Four Permature Machine Language
200 2011 2022 203 205 205 207 207 208 209 200 210 211 213 214 215 215 215 215 215 216 212 212 212 213 214 214 214 214 214 214 214 214 214 214	calledis on tinvolved in System Testingis conducted on whole integrated system to estimate the systems compliance with its specified set ofis integrate within the specified set ofis the start of the system is specified set ofis the start of the system is specified set ofis the system of the system is specified set ofis the system and the system of whole product as ner specified requirements? Which testing is concerned with behavior of whole product as ner specified requirements? Which testing is concerned with behavior of whole product and identifying the defects in them is objective of System anchitecture's are used is whole type of setsing? Publis are found most cost effectively in which test activity? which of this is an disadvantages of small code? Testing makes sure that the functionality of Testing makes sure that the functionality of Testing these test the limits of the System . refricipte of Security Testing, means that data Testing comprises of Testing toechaige for writing test cases. Recovery Testing, we give same inputs in two different versions of the software application. CMM stands for Means and they find the system is capable of handling How many process maturity levels are there? This is not a valid level in CMM The sout of infind level in CMM The sout of infind level in CMM The sout of the advection there inputs in two different const. (Testing is performed without any proper planning. COT's stands for	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Acceptance testing August Alpha Testing To verify that system is Developers Design Avoids hurge coding Enactionality Lusabity Testing Stress Testing Vulnerability Installation Testing Operations Testing Capability Maturity Five Audoo Conventional	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance Restance Business Execution Business Execution Business Execution Business Execution Resultivity Scalability Scalability System Design Integration Parallel Testing Common Two Managed RDBMS Decreases Interstate	User System Testing Design testing Bottom-Up System testing Development Unit testing Internation Testing To verify that system Independent Testers Planning Regressive Testing Regressive Testing Reformance Testing Reformance Testing Reformance Testing Reformance Testing Regressive Testing System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language	Testers Unit Testing Integration Testing Integration Testing Integration Testing Design System Testing Unit Testing To verify that Interfaces Detween Customers Recover and the above Release None of the above Release None of the above Release Recoverability Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Resources and the above Common Modification Four Premature Machine Language
200 2011 2022 203 205 205 207 207 208 209 200 210 211 213 214 215 215 215 215 215 216 212 212 212 213 214 214 214 214 214 214 214 214 214 214	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Alpha Testing To verify that system is Developers Design Avoids hune coding Functionality Testing System Recovery Unabrability Testing System Recovery Unit Testing Operations Testing Operations Testing Operations Testing Operations Testing Conventional Increases Ad Hoc Testing Common Object SQL injection	Designer Interface Testing Integration Sandwich Component Jesting Implementation Acceptance Issilter Heart Testing Business Business Execution Business Execution Business Execution Recoverability Scalability Volume Testing Authorization Stress Testing System Design Integration Parallel Testing Common Two Common Two Managed RDBMS Decreases Interstate RDBMS	User System Testing Design testing Bottom-Up System testing Development Unit testing Indegradue To verify that system Separately testing Regressive Testing Regressive Testing Regressive Testing Reformance Testing Reformance Testing Reformance Testing Reformance Testing System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Parallel Testing Panallel Testing	Testers Unit Testing Integration Testing Integration Testing Integration Testing Integration Testing Operation System Testing Unit Testing To verify that Interfaces Detween Customers Recoverplay the Automation Recoverplay the Automation
200 201 202 202 202 205 206 207 208 209 209 209 209 209 200 210 212 213 213 214 214 215 215 217 212 212 212 212 212 213 214 214 214 214 214 214 214 214 214 214	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Integration testing Developers Design Developers Design Avoids hune coding Functionality Usabitity Testing System Recovery Unitr Testing Operations Testing Operations Testing Operations Testing Operations Testing Operations Testing Operations Testing Common Object SQL injection Performance of system Big-Bang Testing	Designer Interface Testing Integration Sandwich Component seiting Implementation Acceptance Business Business Execution Business Execution Reusability Scalability Scalability Scalability Scalability Scalability Scalability System Design Integration Parallel Testing Common Two Managed RDBMS Decreases Interstate Commencial Off Database injection	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Sagnately testable Independent Testers Planning testing becomes cashier performance Testers Planning Regressive Testing GUI Testing Constant Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Remain Same Parallel Testing Call Online Training Brute Force Attack Online help Bottom-up Testing	Testers Unit Testing Proposal lesting Integration Testing Integration Testing Desian System Testing Unit Testing To verify that interfaces between Customers Release None of the above Relability Testing Recoverability Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language Depide on Software Capability Off the Scripting Attack Shopping Cart Sandwich Testing
200 201 202 203 203 204 205 207 208 209 209 209 209 209 209 209 209 201 2111 212 212 213 214 215 212 212 213 214 215 215 206 6 6 207 207 208 209 209 209 209 209 209 209 209 209 209	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Alpha Testing To verify that system is Developers Design Avoids hung coding Functionality Usabitity Testing System Recovery Unit Testing Operations Testing System Recovery Unit Testing Operations Testing Operations Testing Operations Testing Common Object SQL injection Performance of system Big-Bang Testing Alboc Testing	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance Business Execution Business Execution Business Execution Business Execution Resusability Scalability Scalability Scalability Stress Testing Common Parallel Testing Volume Testing System Design Integration Parallel Testing Common Two Munaged RDBMS Decreases Interstate Common Too Database injection User interfaces Storess Scalability Designess Interstate Common Too-Down Repeatable Hardware	User System Testing Design testing bottom-Up System testing Development Unit testing Integration Testing To verify that system segnately testable Independent Testers Planning Regressive Testing GUI Testing Regressive Testing GUI Testing Call Deling System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Regressive as a second State of the testing Call Online Training Brute Force Attack Online help Bottom-up Testing Change managementi	Testers Unit Testing Proposal testing Integration Testing Integration Testing Operation Testing Unit Testing To verify that interfaces between Customers Release None of the above Reliability Testing Recoverability Testing Installation Testing Derial of Service Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language Contrigues Capability Off the Scripting Attack Shopping Cart Sandwich Testing Following Fixed Jan
200 201 202 203 204 205 206 207 208 209 200 207 208 209 209 201 201 203 204 205 207 208 209 209 200 201 203 205 205 205 205 205 205 205 205	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Developers Design Developers Design Avoids hung coding Environality Testing Stress Testing Vulnerability Testing Stress Testing Operations Testing Capability Maturity Five Adhoc Testing Operations Testing Conventional Increases Ad Hoc Testing Operations Testing Adhoc Testing Operations Testing Adhoc Testing Operations Testing SQL increases Ad Hoc Testing Operations Testing SQL injection Performance of System Receives	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Result of the second Business Execution Business Execution Result of the second System Design Authorization System Design Authorization System Design Authorization System Design Common Union Two Managed Common Two Managed Commercial Octower RDBMS Decreases Interstate Commercial Octower Business Common Top-Down Repeatable	User System Testing Design testing Bottom-Up System testing Development Unit testing Independent Testing To verify that system scorately testing Independent Testers Planning testing becomes cashing Regressive Testing GUI Testing Decision Table System Coding Error Handhing Smoke Testing Capability Decision Table System Coding Error Handhing Smoke Testing Capability Three Defined Assembly Language Remains same Parallel Testing Call Online Training Brute Force Attack Online help Bottom-up Testing	Testers Unit Testing Proposal testing Integration Testing Integration testing Unit Testing Operating System Testing Unit Testing To verify that interfaces between Customers Release None of the above Relability Testing Denial of Service Installation Testing Denial of Service Installation Testing Denial of Service Installation Testing Installation Testing Common Modification Four Premature Machine Language Denial Software Execution Testing Common Modification Four Premature Machine Language Stopping Cart Sandwich Testing Sandwich Testing
200 2011 2022 203 204 205 206 207 208 209 200 207 208 209 207 208 209 207 208 209 201 203 203 203 204 205 205 207 208 209 209 200 207 208 209 209 200 201 201 205 207 208 209 209 209 200 201 201 201 201 205 207 208 209 209 200 201 201 201 201 201 201 201	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Developers Design Avoids hung coding Functionality Usabitity Testing System Recovery Unit Testing Operations Testing System Recovery Unit Testing Operations Testing Operations Testing Operations Testing Adhoc Conventional Increases Ad Hoc Testing Common Object SQL injection Performance of system Big-Bang Testing Adhoc recising Common Object System Esting Adhoc resing Common Object System Sig-Bang Testing Adhoc resing Common Object System	Designer Interface Testing Integration Sandwich Component seiting Implementation Acceptance Business Execution Business Execution Reusability Scalability Scalability Scalability Scalability Scalability Scalability Scalability System Design Imgration Parallel Testing Otomono Parallel Testing Common Parallel Testing Common Parallel Testing Destrass Parallel Testing Common Two Database injection User interface Common Database injection User interface Common Database Top-Down Repeatable Handware Monkey Testing	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system segnately testable Independent Testers Planning testing becomes cashing Regressive Testing GUI Testing Constraints System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Remains same Parallel Testing Call Online Training Brute Force Attack Online help Bottom-up Testing Change management interpreter Agile Testing	Testers Unit Testing Proposal lesting Integration Testing Integration Testing Optimized Content System Testing Unit Testing To verify that interfaces between Customers Release None of the above Reliability Testing Installation Testing Denial of Service Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language control Scripting Attack Shopping Cart Sandwich Testing Following Fixed Plan Language controller Baseline Testing Commot Fixed Testing
200 201 202 202 203 203 205 206 207 208 209 200 201 202 201 202 201 202 201 202 201 202 201 202 203 203 203 203 203 203 203 203 203	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Alpha Testing To verify that system is Developers Design Avoids hune coding Functionality Usabitity Testing System Recovery Unit Testing Operations Testing System Recovery Unit Testing Operations Testing Operations Testing Operations Testing Capability Maturity Five Adhoc Conventional Increases Ad Hoc Testing Common Object SQL injection Performance of system Big-Bang Testing Adhoc receising Common Object System Esting Common Object System Esting Adhoc Testing Common Object System Big-Bang Testing Adhoc receising Common Object System Big-Bang Testing Adhoc receising Common Object System Big-Bang Testing Adhoc receising Common Object System Big-Bang Testing Adhoc receising Common Object System Big-Bang Testing Adhoc receising Clarity	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance Business Execution Business Execution Business Execution Business Execution Reusability Scalability Scalability Scalability Stress Testing Authorization System Design Integration Parallel Testing Oromono Two Managed RyDBMS Decreases Interstate Common Two Managed RyDBMS Decreases Interstate Common Two Managed RyDBMS Decreases Interstate Common Two Managed RyDBMS Decreases Interstate Common Two Managed RyDBMS Decreases Interstate Common Two Managed RyDBMS Decreases Interstate Common Two Managed RyDBMS Decreases Interstate Common Two Notabase Interstate Common Repeatable Hardware Monkey Testing	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system segnately testable Independent Testers Planning testing becomes cashing Regressive Testing GUI Testing Constraints System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Remains same Parallel Testing Call Online Training Brute Force Attack Online help Bottom-up Testing Change management interpreter Agile Testing Change management Interpreter Agile Testing Stating Change management Interpreter Agile Testing Change management Interpreter Agile Testing Maintainable Implementable	Testers Unit Testing Proposal testing Integration Testing Integration Testing System Testing Unit Testing To verify that interfaces between Customers Recovership Testing Recovership Testing Installation Testing Denial of Service Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language Cart Sorpping Cart Sandwich Testing Following Fixed Pan Language Controller Baseline Testing Composed Testing Following Fixed Pan Baseline Testing Composed Testing States State
200 201 202 203 205 205 205 207 207 207 207 207 207 207 207 207 207	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Integration testing Abpla Testing To verify that system is Developers System Recovery Unit Testing Operations Testing Conventional Increases Adhoc rotesting Common Object SQL injection Performance of system Big-Bang Testing Adhoc processes Software interface Gontial Testing Clarity Clarity Clarity Request for Perpetrators	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance Issuing Business Execution Business Execution Reusability Scalability Scalability Volume Testing System Design Integration Parallel Testing Common User Stress Testing System Design Integration Parallel Testing Common User Stress Testing Common User Stress Testing Common User Interfaces Tuo- Database Interstate Commercial Off Database User interfaces Tuo-Down Repeatable Hardware Monkey Testing	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Segnately testable Independent Testens Planning Regressive Testing GUI Testing Regressive Testing GUI Testing Deckino Table System Coding Error Handling Sonde Testing Capability Dericha Table System Coding Error Handling Sonde Testing Capability Defined Defined Language Remains same Parallel Testing Capability Defined Language Remains same Parallel Testing Capability Bottom-up Testing Change management interpreter Aglie Testing Maintainable Imagenetable Request for Potential	Testers Unit Testing Proposal testing Integration Testing Integration Testing System Testing Visite Testing Unit Testing To verify that Interfaces Detween Customers Reclassifue Testing Recovernability Testing Recovernability Testing Recovernability Testing Denial of Service Installation Testing Common Modification Four Premature Denial of Service Installation Testing Common Modification Four Premature Denial of Sorvice Composed and Software Execution Testing Common Modification Four Premature Depends on Software Execution Testing Composed Composed Composed Statements of the Software Statement Composed Composed Statements of the Software Execution Testing Composed Composed Composed Composed Composed Composed Composed Composed Composed Composed Composed Composed Composed Request for prediction Request for prediction
200 201 202 203 205 205 206 207 207 208 209 209 209 209 209 209 209 209 209 209	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Integration testing Acceptance testing Integration testing Abpla Testing To verify that system is Developers System Recovery Unit Testing Operations Testing Conventional Increases Adhoc Testing Common Object SQL ingreases Software Interface Gonila Testing Clarity Clarity Clarity Clarity Developers Deve	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Research Business Execution Business Execution Reusability Scalability Scalability Volume Testing Parallel Testing Parallel Testing System Design Imegration Parallel Testing System Design Imegration Parallel Testing Common Parallel Testing Common Parallel Testing Common Teo Commercial Off Database Interstate Commercial Off Database Interstate Common Robast Commercial Off Database Interstate Common Robast Common Parallel Testing Common Parallel Testing Common	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system segnately testable Independent Testers Planning testing becomes cashing Regressive Testing GUI Testing Constraints System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Remains same Parallel Testing Call Online Training Brute Force Attack Online help Bottom-up Testing Change management interpreter Agile Testing Change management Interpreter Agile Testing Stating Change management Interpreter Agile Testing Change management Interpreter Agile Testing Maintainable Implementable	Testers Unit Testing Proposal testing Integration Testing Integration Testing System Testing Unit Testing To verify that interfaces between Customers Recovership Testing Recovership Testing Installation Testing Denial of Service Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language Cart Sorpping Cart Sandwich Testing Following Fixed Pan Language Controller Baseline Testing Composed Testing Following Fixed Pan Baseline Testing Composed Testing States State
200 201 202 203 202 205 205 206 207 208 209 209 209 209 209 209 209 209 209 209	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Developers Design Avoids huge coding Functionality Usabitity Testing System Recovery Unitr Testing Operations Testing System Recovery Unitr Testing Operations Testing System Recovery Unitr Testing Operations Testing Capability Maturity Five Adhoc Conventional Increases Ad Hoc Testing Common Object SQL injection Performance of system Big-Bang Testing Adhoc recising Common Object Software interface Gorila Testing Clarity Clarity Clarity Engugest for Perpetrators	Designer Interface Testing Integration Sandwich Component esting Implementation Acceptance Business Execution Business Execution Resushilive Execution Resushilive Execution Resushilive Execution Resushilive Scalability System Design Integration Parallel Testing Ourmono Two Managed RDBMS Destrasses Interstate Common Two Managed RDBMS Destrasses Interstate Common Two Managed RDBMS Destrasses Interstate Common Two Nathase Common Two Managed RDBMS Destrasses Interstate Common Two Managed RDBMS Destrasses Interstate Common Two Nathase Top-Down Repeatable Hardware Monkey Testing Traceable Traceable Traceable Traceable Traceable	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system separately testisble Independent Testers Planning testing becomes easier Performance Testers Planning Regressive Testing GUI Testing Capability There Decision Table System Coding Error Handling Smoke Testing Capability There Decision Table System Coding Error Handling Smoke Testing Capability There Defined Assembly Language Remains same Panalle Online training Brute Force Attack Online help Bottom-on Testing Change management Interpreter Agile Testing Agile Testing Agile Testing Capability There Denage management Interpreter Agile Testing Penetration Searching Scenario-based	Testers Unit Testing Proposal lesting Integration Testing Integration Testing Operation testing Unit Testing Unit Testing To verify that interfaces between Customers Release None of the above Reliability Testing Recoverability Testing Installation Testing Denial of Service Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language controller Baseline Testing Capability Off the Scripting Attack Shopping Cart Sandwich Testing Complete Complete Complete Complete Complete Complete Complete Request for prediction
200 201 202 203 204 205 206 207 208 209 209 209 209 209 209 209 209 209 209	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Developers Design Avoids hunge coding Functionality Usabitity Testing System Recovery Usabitity Testing Operations Testing System Recovery Unit Testing Operations Testing Operations Testing System Recovery Unit Testing Operations Testing Capability Maturity Five Adhoc Conventional Increases Ad Hoc Testing Common Object SQL injection Performance of system Big-Bang Testing Adhoc receising Common Object Software interface Gorilla Testing Clarity Clarity Clarity Perpetrators Perpertators	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance Issuing Business Execution Business Execution Reusability Scalability Scalability Volume Testing Execution Reusability Volume Testing System Design Interstate Common Robation System Design Interstate Common Robation Robation Robation Robation Robation Common C	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Samarately testable Independent Testers Planning testing becomes casiler Performance Testers Planning Regressive Testing GUI Testing Capability Decision Table System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Remain Same Parallel Testing Call Online Training Brute Force Attack Online help Bottom-up Testing Change management interpreter Agile Testing Change management Interpreter Agile Testing Change management Interpreter Agile Testing Change management Interpreter Agile Testing Change management Interpreter Agile Testing Change management Request for Potential Threats Penetration Penetration Scanning	Testers Unit Testing Proposal testing Integration Testing Integration Testing Unit Testing System Testing Unit Testing To verify that interfaces between Customers Recoverbility Testing Recoverbility Testing Installation Testing Denial of Service Integration testing System Maintainence Denial of Service Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Lanesuase Depende on Software Execution Testing Four Premature Machine Lanesuase Depende on Software Execution Testing Following fixed Jan Stopping Cart Sandwich Testing Composed Testing Complete Request for prediction Prenetation Invest seministication Treat seministication Threat seministication
200 201 202 203 202 205 205 206 207 207 208 209 209 209 209 209 209 209 209 209 209	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Integration testing Alpha Testing To verify that system is Developers Design Avoids huge coding Functionality Usabitity Testing Stress Testing Valuerability Installation Testing Operations Testing Operations Testing Operations Testing Conventional Increases Ad Hoc Testing Comment Object SQL ingretion Performance of system Big-Bang Testing Content Interface Gontil Testing Control Dijection Performance of System Interface Gontil Testing Carity Clarity Clarity Clarity Carity Clarity Seaming Random Testing E-Business Testing Commercial-Off-the-	Designer Interface Testing Integration Sandwich Component testing Implementation Acceptance testing Research Business Execution Business Execution Reusability Scalability Scalability Volume Testing Execution Reusability Volume Testing System Design Ingeration Parallel Testing Common Parallel Testing Common Parallel Testing Common Parallel Testing Common Design Interstate Common Interface Common Common Common Interface Common RoBMS Datersases Interface Common RoBMS Datersases Interface Common Parallel Testing Common Parallel Testing Common Parallel Testing Common Parallel Testing Common Interface Common Parallel Testing Common Parallel Testing Common C	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system separately testsble Independent Testers Planning testing becomes easier Performance Testers Planning Regressive Testing GUI Testing Compatibility Direction Table System Coding Error Handling Smoke Testing Capability Three Defined Defined Defined Capability Three Defined Capability Three Defined Capability Three Defined Capability Three Defined Capability Rate Force Attack Online help Bottom-on Testing Change management Interpreter Agile Testing Return Testing Compatibility Return Testing Compatibility Science To Potential Threats Scientific Potential Threats Scientific Potential Threats Scientific Potential Threats Scientific Control Threats Scientific Control Threats Scientific Control Threats Scientific Control Threats Commercial Off-the-	Testers Unit Testing Proposal testing Integration Testing Integration Testing Unit Testing Unit Testing Unit Testing Unit Testing To verify that inferfaces between Customers Reclassify Testing Recoverability Testing Recoverability Testing Recoverability Testing Recoverability Testing Installation Testing Denial of Service Installation Testing Installation Testing Installation Testing Installation Testing Denial of Service Denial of Service Denial of Service Denial of Service Denial of Service Installation Testing Common Medification Four Premature Denial of Service Depends on Software Execution Testing Common Medification Four Premature Pendure Common Medification Four Premature Scripting Attack Shopping Cart Sandwich Testing Following fixed Ina Language cosmoller Baseline Testing Complete Complete Complete Complete Complete Complete Conservation Ineglementation Interest sensing Class Testing E-Learning Testing
200 201 202 202 203 202 205 206 207 207 208 209 209 209 209 209 209 2101 211 212 213 213 214 213 214 213 214 214 222 223 224 225 226 229 229 220 220 220 220 220 220 220 220	called	Developers Integration Testing Module testing Big-Bang Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Acceptance testing Developers Design Avoids huse coding Functionality Usability Testing System Recovery Unit Testing Operations Testing System Recovery Unit Testing Operations Testing System Recovery Unit Testing Operations Testing Capability Maturity Five Adhoc Conventional Increases Ad Hoc Testing Common Object SQL injection Performance of system Big-Bang Testing Adhoc processes Software interface Gorilla Testing Clarity Clarity Clarity Perpetrators Perpetrators Perpetrators Random Testing E-Basingess Testing	Designer Interface Testing Integration Sandwich Component esting Implementation Acceptance Business Execution Business Execution Resultive Scalability Scalability Scalability Scalability Commore Parallel Testing Parallel Testing Otomer Scalability Parallel Testing Common Parallel Testing Common Parallel Testing Common Parallel Testing Common Parallel Testing Destrates Parallel Testing Common Parallel Testing Common Parallel Testing Common Parallel Testing Common Parallel Testing Destrates Parallel Testing Common Parallel Testing Destrates Common Parallel Testing Common Parallel Testing Parallel Testing Parallel Testing Common Com	User System Testing Design testing Bottom-Up System testing Development Unit testing Integration Testing To verify that system Samparately testable Integratedent Testers Planning testing becomes casiler Performance Testing GUI Testing Capability Decision Table System Coding Error Handling Smoke Testing Capability Three Defined Assembly Language Remains Same Parallel Testing Call Online Training Brotter Orce Attack Online help Bottom-up Testing Change management interpreter Agile Testing Change management Intergreter Agile Testing Change management Intergreter Agile Testing Change management Intergreter Agile Testing Change management Intergreter Agile Testing Change management Penetration Penetration Scenario-based Testing Econtent Testing	Testers Unit Testing Proposal testing Integration Testing Integration Testing Unit Testing Operation Testing Unit Testing To verify that interfaces thetween Customers Release None of the above Reliability Testing Installation Testing Common Modification Four Premature Machine Language Total Installation Testing Installation Testing Installation Testing Installation Testing Installation Testing Common Modification Four Premature Machine Language Control Four Scripting Attack Shopping Cart Sandwich Testing Complete Complete Complete Complete Complete Caracting Test String Class Testing ELarring Testing

	W		le.	T.	T
	How many types of performance testing techniques are	one	Four	Two	Three
245	Which techniqe is applied for usability testing?	White-box testing	Black-box	Gray-box testing	Combined to all testing
	is the process of executing the code and	Test Execution	Test analyst	Test process	Test expert
246	comparing the expected and actual results.				
	is a testing technique in which the same	Parallel testing	complexity-	Stress-testing	volume-testing
247	inputs are entered in two different versions of the application		testing		
	Agile testing is unstructured as compared to the	SDLC	Agile	Parallel approach	waterfall approach
248	and there is minimal planning.				
249	is minimal and the test process less	Test process	Test planning	Test case	test data
250	Acceptance testing is conducted at theof each	Start	Middle	end	every step