

ISTQB.ISTQB-CTFL-KR.v2026-01-22.q147

□□□□:	ISTQB-CTFL-KR
□□□□:	ISTQB-Foundation Level Exam (ISTQB-CTFL Korean Version)
□□□:	ISTQB
□□ □□ □□□:	147
□□:	v2026-01-22
# □□ □:	175
# □□ □□□:	1470
https://www.krdump.com/ISTQB.ISTQB-CTFL-KR.v2026-01-22.q147.html	

NEW QUESTION: 1

□□ □□ □□□□(ATM)□□ □□□ □□□□□ □□□ □□ □□□ □□□□□.

- □□□□□ □□□□□

- PIN □□□ □□□□□.

- □□□ □□□ □□ □□□□

□□□□ ATM □□□□ □□□ □ □□ □□□□□ □□□ □□□□.

- □□□□ □□□ □□□□ □□□□. □□□ □□□□□.

- PIN □□□ □□□□ ATM□□ □□ PIN □□□ □□□□

- □□□ □□□ □□□ □□□ □□ □□□□ □□ □□: ATM□□ □□ □□□ □□□□□.

- □□□ □□□ □□□ □□□ □□□□. ATM□□ □□ □□□□□. □□ □□ □□□ □□□ □□ □□□□ □□ □□ □□□ □□ □□□ □□□□ □□□□ □□□?

A. □□ □□ □□ □□□

B. □□□

C. □□□ □□

D. □□ □□□ □□

Answer: B ([LEAVE A REPLY](#))

A decision table is a technique that should be used to cover all possible combinations of input conditions for withdrawing money from an Automated Teller Machine (ATM). A decision table shows combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects). A decision table consists of four quadrants: conditions (inputs), actions (outputs), condition entries (values) and action entries (results).

A decision table can be used to test components that have multiple inputs and outputs that depend on logical combinations of conditions. For example, for testing the ATM, we can identify three input conditions: the bank card is valid, the PIN code is correct, and money is available in the user's account. We can also identify four output actions: the card is rejected, the ATM asks for another PIN code, the ATM asks for another amount, and the ATM dispenses the money. A

decision table can show all possible combinations of these conditions and actions in a systematic way.

Use case based testing is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Use case based testing is a technique that verifies that a software product or system meets its specified requirements or user stories by executing realistic scenarios or workflows. Use case based testing can be used to test components that have complex or dynamic interactions with users or other systems. For example, for testing the ATM, we can identify several use cases, such as withdraw money, check balance, transfer money, etc. Each use case can have one or more scenarios that describe the steps and outcomes of the interaction. However, use case based testing may not cover all possible combinations of input conditions, as some scenarios may be omitted or overlooked.

Boundary value analysis is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Boundary value analysis is a technique that tests boundary values between partitions of equivalent data. Boundary values are values at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge.

Boundary value analysis can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify boundary values for the input amount, such as the minimum and maximum amount allowed by the system or the user's account. However, boundary value analysis may not cover all possible combinations of input conditions, as some conditions may not have boundary values or may not be related to input values.

Equivalence class partitioning is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Equivalence class partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component.

Equivalence class partitioning can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify equivalence partitions for the input amount, such as valid amount (within the range allowed by the system and the user's account) and invalid amount (outside the range allowed by the system or the user's account). However, equivalence class partitioning may not cover all possible combinations of input conditions, as some conditions may not be related to input values or may have more than two partitions. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 4, page 34-46.

NEW QUESTION: 2

3□ □□ □□□ □□□□ □□ □□□□ □□ □□ □□ SD□ □□□□□□. □□ □□□□ □□
(a) -120□□ □□ □□□ □□ □□(m) -180□□ □□ □□□□ □□(b)
- 240□□

- A. 120
- B. 20
- C. 180

D. 197

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 3

□□ □ □□ □□□ □□ □□ □□□□□?

A. □□□□□ □□□□ □□ □□

B. □□□ □□□□ □□ □□□□□□ □□□□□.

C. □□□ □□□□ □□□ □□□

D. □□□□□□ □□□ □□□ □□□□ □□□□.

Answer: ([SHOW ANSWER](#))

Problems in defining the right requirements is not a product risk, but rather a project risk. A product risk is a risk that affects the quality or performance of the software product itself, such as poor usability, failure-prone functionality, security vulnerabilities, compatibility issues, etc. A project risk is a risk that affects the management or delivery of the software project itself, such as unrealistic schedule, insufficient resources, unclear scope, changing requirements, etc. The other options are examples of product risks, as they relate to the software product's characteristics or features. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 12.

NEW QUESTION: 4

□□ □□ □ □□□ □□□ □□□ □□ □□□□□?

I) □□ □ □□ □□ □□

II) □□ □□□□ □□□ □□

III) □□□ □□□ □ □□

IV) □□□ □□□□ □□ □□ □□ □□

A. 1. 4.

B. 1. 3.

C. 1, 3, 4

D. 1, 2, 4

Answer: B ([LEAVE A REPLY](#))

Test planning is a key activity in the testing process that involves defining the objectives, approach, resources, and schedule of intended test activities. Setting the entry and exit criteria (I) and determining the number of resources required (III) are integral parts of test planning. Determining the validity of bug reports (II) is more aligned with test analysis or test management activities post-execution, and determining the expected result for test cases (IV) is part of test design. Therefore, options I and III (B) are the activities that belong to test planning.

NEW QUESTION: 5

□□□□□ □□ □□□ □□□□ □□□ □□ □□□□□ □□ □□□□□□□ □□□□ □□□
□□□ □□□ □□ □□□□. □ □□□ □□□ □□□□ □□□□ □□□□, □□ □□□ □□□

Which of the following is a key benefit of DevOps?
 DevOps enables faster time to market for new products.

- A. DevOps enables faster time to market for new products.
- B. DevOps enables faster time to market for new products.
- C. DevOps enables faster time to market for new products.
- D. DevOps enables faster time to market for new products.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 6

Which of the following is a key benefit of DevOps?
 DevOps enables faster time to market for new products.

- A. DevOps enables faster time to market for new products.
- B. DevOps enables faster time to market for new products.
- C. DevOps enables faster time to market for new products.
- D. DevOps enables faster time to market for new products.

Answer: D ([LEAVE A REPLY](#))

A failure is an event in which a component or system does not perform a required function within specified limits. A failure is observable by the software users or other stakeholders. A failure is caused by one or more defects in the software. In this case, the software not being able to meet the desired response time is an example of a failure, as it deviates from the stated requirement and affects the user experience. It is not a defect, which is a flaw in the software that causes the failure. It is not an error, which is a human action that produces an incorrect result. It is not a non-defect, as it clearly violates a specified requirement. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 1, page 4.

NEW QUESTION: 7

- Which of the following is a key benefit of DevOps?
 DevOps enables faster time to market for new products.
- A. DevOps enables faster time to market for new products.
 - B. DevOps enables faster time to market for new products.
 - C. DevOps enables faster time to market for new products.
 - D. DevOps enables faster time to market for new products.

Answer: ([SHOW ANSWER](#))

A typical test plan includes various elements, such as resource requirements, test completion criteria, and suspension/resumption criteria. However, the list of product risks that have not been fully mitigated is generally not included in the test plan but rather in the risk management documentation.

* The test plan focuses on planning and executing tests, including resource allocation and defining criteria for test suspension and resumption.

* While risk management is crucial, unmitigated risks are typically documented in risk logs or separate risk management plans.

Reference: ISTQB CTFL Syllabus V4.0, Chapter 5.1.1, Test Planning.

NEW QUESTION: 8

□□ □ □□ □□(QA)□ □□ □□(QC)□ □□ □□□□ □□□□□?

A. A□ □□□□□ □□□□ □□ □□□ □□□ □□, QC□ □□ □□ □□□□□.

B. A□ □□□□□, QC□ □□□□□□.

C. A□ □□□ □□□ □□ □□, QC□ □□□ □□□ □□□.

D. A□ □□□□ □□□ □□ □□□□, QC□ □□ □□□ □□ □□□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 9

□□ □□□ □□□ □□□□□□□ □□□□ □□□□□ □□□□ □□□□ □□□□ □□□□
□□ □□□□□ □□□□ □□□□.

□□□□□ □□ □□□ □□□ □□□□ □□□□ □ □□ □□□ □□ □□□ □□□□□?

A. □□□□

B. □□□ □□

C. □□ □□

D. □□ □□

Answer: C ([LEAVE A REPLY](#))

A technical review is a type of formal review that involves a team of technical experts who evaluate a software product against a set of predefined quality criteria. A technical review is suitable for ensuring high quality and technical correctness of complex or critical software components, such as algorithms, architectures or designs. A technical review is not a walkthrough, which is an informal review led by the author of the work product. A technical review is not an informal review, which is a review that does not follow a defined process and has no formal entry or exit criteria. A technical review is not a management review, which is a type of formal review that focuses on business aspects and project progress. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 3, page 29-30.

NEW QUESTION: 10

□□□ □□□□ □□ □□ □□ □ □□ □□ □□□□□?

I. □□ □□□□ □□□□ □□□ □□□ □□□.

II. □□□ □□□□ □□ □□□□□ □□ □□□ □□□ □□ □□□ □□□□ □□□.

□□ □□□□ □□ □□ □□ □□□ □□ □□□□ □□□.

IV. □□□ □□□□ □□□, □□□□ □ □□□ □□□ □□□ □□□.

V. □□□ □□□□□ □□ □□ □□□ □□ □□□ □□□□□ □□□.

A. 2, 3, 5

B. 1, 2, 4

C. 1, 3, 5

D. 2, 3, 4

Answer: ([SHOW ANSWER](#))

Statements II, III and V are true about test reports. Test reports are documents that provide information on the results and status of testing activities for a given period or phase. Test reports should give stakeholders information as basis for decisions, such as whether to release the software product, whether to continue testing, whether to change the scope or priorities of testing, etc. Test reports should summarize what happened through a period of testing, such as what test cases were executed, what defects were found, what risks were identified, what issues were encountered, what achievements were made, etc. Test reports should include information about remaining risks, such as what defects are still open, what test cases are still pending, what functionalities are still untested, what uncertainties are still unresolved, etc. Statements I and IV are not true about test reports. Test reports do not need to be approved by the test team, the development team, or the customer, unless it is specified by the test policy or the test plan. Test reports only need to be reviewed and verified by the test leader or the test manager before being distributed to the intended recipients. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 141.

NEW QUESTION: 11

□□ □ □□□ □□ □□□□ □□□□□ □□ □□□ □□□□□?

I. □□ □□ □□

II. □□ □□□ □□ □□□ □ □

III □□ □□□ □□

IV. □□ □□

V. □□□ □□□, □□□ □□□, □□□ □□□□ □

A. III.V

B. II, III

C. I. II. IV

D. II, III.V

Answer: D ([LEAVE A REPLY](#))

The following should be included in a test status report: total number of open and closed defects, actual effort spent, and number of executed, failed, and blocked tests. A test status report is a document that provides information on the results and status of testing activities for a given period or phase. A test status report should include information that is relevant, accurate, and timely for the intended audience and purpose. Some of the information that should be included in a test status report are: total number of open and closed defects, which can indicate the defect trend and defect density of the software product; actual effort spent, which can indicate the productivity and efficiency of the testing process; number of executed, failed, and blocked tests, which can indicate the test progress and test coverage of the software product. The following should not be included in a test status report: estimation details, defect reports, and impact analysis. Estimation details are not part of a test status report, but rather part of a test plan or a test estimation

document. Estimation details provide information on the expected time, resources, and costs for testing activities, not on the actual results or status of testing activities. Defect reports are not part of a test status report, but rather separate documents that provide detailed information on individual defects found during testing. Defect reports include information such as defect description, defect severity, defect priority, defect status, defect resolution, etc.

Defect reports can be referenced or summarized in a test status report, but not included in full. Impact analysis is not part of a test status report, but rather part of a risk assessment or prioritization process. Impact analysis provides information on the potential effects or consequences of a change or a defect on the software product or project. Impact analysis can be used to evaluate the amount or scope of testing to be performed, but not to report the results or status of testing activities. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 141.

NEW QUESTION: 12

□□□ □□□□ □□□ □□ □□□ □ □□ □□□ □□□ □□□□.

- A. □□□ □□□ □□□□ □□ □ □□□□.
- B. □ □□□ □□□□ □□ □□□□ □□□ □ □□□□.
- C. □□ □□□ □□ □□□□ □□□□ □□ □ □□□□.
- D. □ □□□ □□□□ □□□□ □□ □□□□ □□□□ □□□□□.

Answer: ([SHOW ANSWER](#))

One possible risk of introducing test automation is that the selected tool may not be fit-for-purpose. This means that the tool might not meet the specific needs and requirements of the project, leading to inefficiencies and possibly failing to provide the expected benefits. It is crucial to evaluate and select the appropriate tool based on the project's context and objectives. The ISTQB CTFL syllabus highlights the importance of careful tool evaluation and selection to ensure it aligns with the testing goals and the development environment.

References: ISTQB CTFL Syllabus, Section 6.2, "Potential Benefits and Risks of Test Automation."

NEW QUESTION: 13

□□ □□□ □□ □ □□ □□□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□□?

- A. □□□ □□ □□
- B. □□□□ □□
- C. □□□□ □□
- D. □□□ □□

Answer: ([SHOW ANSWER](#))

Test exit criteria are the conditions that must be fulfilled before concluding a particular testing phase. These criteria act as a checkpoint to assess whether we have achieved the testing objectives and are done with testing¹. Test exit criteria are typically defined in the test plan document, which is one of the outputs of the test planning phase. The test plan document describes the scope, approach, resources, and schedule of the testing activities. It also identifies

the test items, the features to be tested, the testing tasks, the risks, and the test deliverables². According to the ISTQB Certified Tester Foundation Level Syllabus v4.0, the test plan document should include the following information related to the test exit criteria³:

- * The criteria for evaluating test completion, such as the percentage of test cases executed, the percentage of test coverage achieved, the number and severity of defects found and fixed, the quality and reliability of the software product, and the stakeholder satisfaction.
- * The criteria for evaluating test process improvement, such as the adherence to the test strategy, the efficiency and effectiveness of the testing activities, the lessons learned and best practices identified, and the recommendations for future improvements.

Therefore, the test plan document is the most appropriate test document to find the test exit criteria described.

The other options, such as test design specification, project plan, and requirements specification, are not directly related to the test exit criteria. The test design specification describes the test cases and test procedures for a specific test level or test type³. The project plan describes the overall objectives, scope, assumptions, risks, and deliverables of the software project⁴. The requirements specification describes the functional and non-functional requirements of the software product⁵. None of these documents specify the conditions for ending the testing process or evaluating the testing outcomes. References = ISTQB Certified Tester Foundation Level Syllabus v4.0, Entry and Exit Criteria in Software Testing | Baeldung on Computer Science, Entry And Exit Criteria In Software Testing - Rishabh Software, Entry and Exit Criteria in Software Testing Life Cycle - STLC [2022 Updated] - Testsigma Blog, ISTQB releases Certified Tester Foundation Level v4.0 (CTFL).

NEW QUESTION: 14

□□ □□ □□ □□□ □□ □□□□ □□□□ □□□ □□□□□ □□ □□□ □□□□ □ □□ □□□ □□□□□?

- A. □□□□□ □□
- B. □□ □□ □□.
- C. □□ □□ □□.
- D. □□□ □□□ □□ □□

Answer: C ([LEAVE A REPLY](#))

A static analysis tool is best suited for determining source code compliance with coding standards. These tools analyze the code without executing it and can check for adherence to coding standards, syntax errors, and other static properties of the code. The ISTQB CTFL syllabus emphasizes the role of static analysis tools in verifying that code meets predefined standards and guidelines.

References: ISTQB CTFL Syllabus, Section on static testing and tools.

NEW QUESTION: 15

□□ □ MOT□ □□□□ □□□ □□□ □□□□ □□ □□□□□?

- A. □□□ 100% □□ □□□□□ □□□□□ □□□□□□.

- B. □□□ □□□ □□□ □□□ □□ □□□□□□ □□□ □□□□ □□□ □□□□□□.
- C. □□□□ □ □□ □□□□ □□ □□□□□ □□□□ □□□ □□ □□□□□□.
- D. □□□ □□□ □□□ □□□ □□□ □□□□ □□ □□□□.

Answer: B (LEAVE A REPLY)

Testing is necessary for various reasons, such as:

- * To detect defects and failures that may affect the quality, performance, reliability or security of a software product or system
 - * To verify that a software product or system meets its specified requirements, expectations and standards
 - * To validate that a software product or system fulfills its intended purpose and satisfies its stakeholders' needs
 - * To provide information and feedback about the status and risks of a software product or system
 - * To comply with regulations or contractual obligations that mandate testing for certain types of software products or systems
- The following statements describe some reasons why testing is necessary:
- * A) The customer decided that 100% branch coverage shall be achieved. This is a reason why testing is necessary, as it reflects a contractual obligation or a quality standard that requires testing to measure and achieve a certain level of code coverage.
 - * C) For avionics and pharmaceutical systems software testing is mandated by standards. This is a reason why testing is necessary, as it reflects a regulation or a compliance requirement that mandates testing for certain types of software products or systems that have high safety or security risks.
 - * D) The risks associated with delivering the system are far higher than the cost of testing. This is a reason why testing is necessary, as it reflects a risk-based approach that considers testing as an investment to reduce the probability and impact of potential failures or defects. The following statement does not describe a reason why testing is necessary:
 - * B) The acquisition of test automation tools was based on the assumption that it will be used in all projects. This is not a reason why testing is necessary, as it reflects a business decision or a resource allocation that does not justify the need or purpose of testing. Test automation tools are not always suitable or beneficial for all projects, and testing can be performed with or without test automation tools. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 1, page 5-6.

NEW QUESTION: 16

□□□ □□□□ □□□ □□ □□:

- A. "Given/When/Then"□□□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□ □□□□.
- B. □□ □□□□ □□□□ □□ □□ □□□□ □□□□□ □□□ □□□□. "[□□]□□ [□□]□□ □□□□ [□□]□ □□ □ □□□□."
- C. □□□ □□□□ □□□ □ □□□, □□ "3C"□ □□□□□ □□ □□□ □□□□ □□□ □□ □□□.
- D. □□ □ □□ □□ □ □□□ □□□□ □□□: □□□□ □□ □□ □□ □□

Answer: C (LEAVE A REPLY)

The acceptance criteria associated with a user story are the conditions that must be met for the user story to be considered done and to deliver the expected value to the user. They are often written in different formats, such as rule-oriented, scenario-oriented, or table-oriented, depending on the nature and complexity of the user story. They represent an aspect of a user story referred to as confirmation, which is one of the so called "3 C" s of user stories. The other two aspects are card and conversation. Card refers to the concise and informal description of the user story, usually following the template: "As a [role], I want [feature], so that I can [benefit]". Conversation refers to the ongoing dialogue between the stakeholders and the team members to clarify and refine the user story and its acceptance criteria. Therefore, option C is the correct answer.

ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 3.2.2, page 35-36; ISTQB Glossary v4.02, page 37.

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□. □□□□ □□□ □□□□ □□□□ □□□□ □□□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 17

[illegible]

- A.** □□□□ □□□□ □□□ □□□□□□□□ □□ □□□□ □□ □□□□ □□□ □□□□□
□□□□, □□□□ □□□ □□□□ □□□ □□□□.
- B.** □□□□ □□□ □□□ □□□ □□□ □□□□ □□□ □□□ □□ □□□□ □□□ □ □□
□□.
- C.** □□□ □□ □□ □□□ □□□ □□□ □□□□ □□□ □□□□□ □□□ □□ □□□ □□
□□□ □□□□□.
- D.** □□ □□ □□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□□ □□□ □□ □□ □□
□ □□□.

Answer: ([SHOW ANSWER](#))

Exploratory testing is an experience-based test technique where testers actively engage with the software, learning about its behavior while simultaneously designing and executing tests. According to the ISTQB CTFL syllabus, exploratory testing can be structured into sessions guided by test charters, which outline the test objectives and provide direction for the testers' exploration. This method is particularly useful in situations where test documentation is limited or where rapid feedback is needed. Thus, option B correctly describes how exploratory testing can be organized.

NEW QUESTION: 18

□□ □□□ □□□ □ □□□ □□□ □□□□□ □ □□□ □□□ □□□□ □□□□□□. □ □
□ □□□ □□□ □□□ □, □□□□ □□□ □□□ □□□ □□ □ □□□□ □□□□□□□□.
□□ □□□ □□□ □□ □□□ □□□ □□ □□ □□□ □□□ □□□ □□ □□□□□?

- A. □□□□ □□
- B. □□ □□□
- C. □□□ □□ □□
- D. □□ □□□□□

Answer: D ([LEAVE A REPLY](#))

Defect clustering is a testing principle that states that a small number of modules contain most of the defects detected during pre-release testing, or are responsible for most of the operational failures. Defect clustering can be explained by Pareto's principle (also known as the 80-20 rule), which states that approximately 80% of the problems are found in 20% of the modules. Defect clustering suggests a change to the current test strategy for the next test cycle, as it implies that more effort should be allocated to test the modules that have shown high defect density or criticality. Pesticide paradox is another testing principle that states that if the same tests are repeated over and over again, eventually they will no longer find any new defects. Pesticide paradox suggests a change to the current test strategy for the next test cycle, but not based on defect clustering, but rather on test diversity and coverage. Early testing is a testing principle that states that testing activities should start as early as possible in the software development life cycle and should be focused on defined objectives.

Early testing does not suggest a change to the current test strategy for the next test cycle, but rather a proactive approach to prevent defects from occurring or propagating. Absence-of-errors fallacy is a testing principle that states that finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations. Absence-of-errors fallacy does not suggest a change to the current test strategy for the next test cycle, but rather a focus on quality attributes and user requirements. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 1, page 9-10.

NEW QUESTION: 19

- □ □□ □□ □□□ □□ □□ □ □□ □□ □□□□□?
- A. Shift-Left □□□ □□□ □□□□□ □□ □□□□□.
 - B. □□□□□□ □□ □□ □□□ DevOps □□□ □□□□□.
 - C. □□□□ □□□□ □□ □□□ □□ □□□□ □□□ □ □□□□.
 - D. □□□□ □□ □□□ □□ □□ □□□ □□□ □ □□□□.

Answer: A ([LEAVE A REPLY](#))

In a formal review process, the recorder's role is typically responsible for documenting the findings of the review team, including action items, decisions, and recommendations. This ensures that there is an accurate record of what was discussed and agreed upon, facilitating follow-up and continuous improvement. Therefore, statement C is correct as per the ISTQB CTFL syllabus.

NEW QUESTION: 20

□□ □□ □ □□ □□ □□□ □□ □□ □□□ □□ □ □□ □□ □□ □ □□□□□?

- A. □□□□ □□ □□□ □□ □□ □□ □□□ □□□ □□ □□ □□□ □□ □□□ □□ □□□.
- B. □□□ □□ □ □□ □□ □□ □□ □□ □□ □□ □□ □□ □□□ □□ □.
- C. □□□ □□ □□ □□ □□□ □□□ □□ □□ □□□ □□ □□ □□ □ □□ □□ □ □□□.
- D. □□ □□ □□ □□□ □□ □□ □□ □□ □□□ □□□ □□ □□ □ □□ □□□ □□.

Answer: D ([LEAVE A REPLY](#))

The statement that refers to good testing practice to be applied regardless of the chosen software development model is option D, which says that involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle. Work product reviews are static testing techniques, in which the work products of the software development process, such as the requirements, the design, the code, the test cases, etc., are examined by one or more reviewers, with or without the author, to identify defects, violations, or improvements. Involvement of testers in work product reviews can provide various benefits for the testing process, such as improving the test quality, the test efficiency, and the test communication. The early testing principle states that testing activities should start as early as possible in the software development lifecycle, and should be performed iteratively and continuously throughout the lifecycle. Applying the early testing principle can help to prevent, detect, and remove defects at an early stage, when they are easier, cheaper, and faster to fix, as well as to reduce the risk, the cost, and the time of the testing process. The other options are not good testing practices to be applied regardless of the chosen software development model, but rather specific testing practices that may or may not be applicable or beneficial for testing, depending on the context and the objectives of the testing activities, such as:

* Tests should be written in executable format before the code is written and should act as executable specifications that drive coding: This is a specific testing practice that is associated with test-driven development, which is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design, and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a good testing practice to be applied regardless of the chosen software development model, as it may not be feasible, suitable, or effective for testing in some contexts or situations, such as when the requirements are unclear, unstable, or complex, when the test automation tools or skills are not available or adequate, when the testing objectives or levels are not aligned with the unit testing, etc.

* Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level: This is a specific testing practice that is associated with sequential software development models, such as the waterfall model, the V-model, or the W-model, in which the software development and testing activities are performed in a linear and sequential order, with well-defined phases, deliverables, and dependencies. Test levels are the stages of testing that correspond to the levels of integration of the software system, such as component testing, integration testing, system testing, and acceptance testing. Test levels should have clear and measurable entry criteria and exit criteria, which are the conditions that must be met before starting or finishing a test level. In sequential software development models, the exit criteria of one test level are usually part of the entry criteria for the next test level, to ensure that the software system is ready and stable for the next level of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be relevant, flexible, or efficient for testing in some contexts or situations, such as when the software development and testing activities are performed in an iterative and incremental order, with frequent changes, feedback, and adaptations, as in agile software development models, such as Scrum, Kanban, or XP, when the test levels are not clearly defined or distinguished, or when the test levels are performed in parallel or concurrently, etc.

* Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly: This is a specific testing practice that is associated with uniform software development models, such as the spiral model, the incremental model, or the prototyping model, in which the software development and testing activities are performed in a cyclical and repetitive manner, with similar phases, deliverables, and processes. Test objectives are the goals or the purposes of testing, which can vary depending on the test level, the test type, the test technique, the test environment, the test stakeholder, etc. Test objectives can be defined in terms of the test basis, the test coverage, the test quality, the test risk, the test cost, the test time, etc. Test objectives should be specific, measurable, achievable, relevant, and time-bound, and they should be aligned with the project objectives and the quality characteristics. In uniform software development models, the test objectives may be the same for all test levels, as the testing process is repeated for each cycle or iteration, with similar focus, scope, and perspective of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be appropriate, realistic, or effective for testing in some contexts or situations, such as when the software development and testing activities are performed in a hierarchical and modular manner, with different phases, deliverables, and dependencies, as in sequential software development models, such as the waterfall model, the V-model, or the W-model, when the test objectives vary according to the test levels, such as component testing, integration testing, system testing, and acceptance testing, or when the test objectives change according to the feedback, the learning, or the adaptation of the testing process, as in agile software development models, such as Scrum, Kanban, or XP, etc.

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle¹

- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.2, Test Monitoring and Control1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.4, Test Implementation1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.5, Test Execution1
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.6, Test Closure1
- * ISTQB Glossary of Testing Terms v4.0, Work Product Review, Static Testing, Early Testing, Test- driven Development, Test Level, Entry Criterion, Exit Criterion, Test Objective, Test Basis, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time2

NEW QUESTION: 21

□□□ □□□ □□□ □□ □□□ □□□ □□ □□□ □□ □□□□ □□□□□?

- A. □□ □□□ □□□ □□□□, □□ □□□ □□□ □□□□□.
- B. □□ □□□ □□□□□ □□□ □□□□ □□ □□, □□ □□□ □□□□ □□□ □□□ □ □□□.
- C. □□ □□□ □□□ □□ □□□ □□ □ □□□ □□ □□□ □□ □ □□□□.
- D. □□ □□□ □□□ □□□□, □□ □□□ □□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

The key difference between a peer review of code and static analysis of code using a tool lies in their approaches and scope. A peer review is a manual inspection of the code by peers or colleagues, focusing not only on the technical aspects of the code but also on other elements such as design, compliance with standards, and maintainability. Peer reviews can identify defects, suggest improvements, and ensure that the code adheres to best practices and team standards.

On the other hand, static analysis is an automated process performed by tools designed to analyze the code without executing it. These tools can detect potential issues such as syntax errors, vulnerabilities, and code smells based on predefined rules and patterns. While static analysis is technically focused, it lacks the broader perspective that human reviewers can provide, such as evaluating the code's maintainability or adherence to project-specific standards. Therefore, static analysis targets the code technically, whereas peer review encompasses a wider range of aspects, making option B the correct answer.

NEW QUESTION: 22

□□ □ □□ □ □□ □□□ □□□□ □□ □□□ □□□□ □□□ □□ □□□□□?

- A. □□ □□□ □□□ □□□□ □□□ □ □□□□.
- B. □□ □□□□ □□□□ □□ □□.
- C. □□□ □□□ □□ □□□ □□□□ □□ □□.
- D. □ □□□ □□□□ □ □.

Answer: A ([LEAVE A REPLY](#))

The whole team approach involves collaboration among all team members, including testers, developers, and business representatives, to achieve quality goals. However, this approach may not be suitable in situations where a high level of test independence is required. Test independence is essential in cases where unbiased testing is critical, such as in regulated environments or where high-risk systems are involved. This is because team members might unintentionally influence each other's work, leading to potential bias in testing outcomes.

NEW QUESTION: 23

- □□ □□□ □□□□ □□□ □□ □□ □□ □ □□□□□?
- A. □□ □□□ □□ □□ □□ □□□ □□□ □□ □□□□ □□□□□ □□ □ □□□□ □□□ □□.
 - B. ISO/IEC/IEEE 29119-3 □□□□ □□□□ □□ □□ □□ □□□ □ □□ □□□ □□ □□□ □□□□ □□□□.
 - C. □□ □□□□ □□□ □ □□□ □□□□□ □□ □□ □□□□ □□□□□.
 - D. □□ □□□ □□ □□ □ □□□ □□ □□□□ □□ □□□□□□ □□□ □□□ □□□ □□ □□ □□□ □□□□ □□□□.

Answer: C ([LEAVE A REPLY](#))

During the review planning stage, key decisions are made, including the selection of standards and procedures to be followed during the review. This planning phase ensures that the review process is structured and adheres to agreed-upon standards, which can come from industry standards such as ISO/IEC/IEEE 29119-3.

The ISTQB CTFL Syllabus v4.0 emphasizes the importance of review planning in establishing the framework and guidelines for the review process.

NEW QUESTION: 24

- □□ □□□□ □□ □□ □□ □ □□ □□ □□□□□?
- A. □□ □□□ □□□□ □□□ □□□□ □□□ □□ □□ □□□□□□□.
 - B. □□□ □□□□ □□ □□□□ □□ □□ □□□□ □□□□□ □□□□□□□.
 - C. □□□ □□□□ □□ □□□□ □□ □□□□ □□□□□.
 - D. □□□ □□□ □□□□ □□□□ □□□ □□ □□ □□□□□□□.

Answer: D ([LEAVE A REPLY](#))

Use case testing is a technique that helps identify test cases that exercise the whole system on a transaction by transaction basis from start to finish. Use cases are descriptions of how users interact with the system to achieve a specific goal. Use case testing is not focused on data flow, but rather on process flow. Use case testing can be performed by professional testers, customers or end users, depending on the context. Use case testing does not require the test cases to be

designed by customers or end users, but rather by anyone who has access to the use case specifications. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 4, page 36.

NEW QUESTION: 25

□□ □□ □ □□ □□ □□(BDD)□ □□ □ □□□□ □□ □□□□□?

- A. □□ □□□□ □□□□ □□ □□□□ □□□ □□□ □□□ □□□□□□□□ □□□ □□□ □□□□□. □□□□□ Given/When/Then □□□ □□□□□. □□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□ □□□□□.
- B. □□ □□□ □□□□□□ □□□□ □□□ □□□ □□□□□ □□□ □□□□ □□□□□.
- C. □□□ □□□□ □□□ □□□□ □□ □□□ □□□ □□□ □□ □□□ □□□.
- D. □□□□□ □□□ □□□□ □□ □□□ □□□ □ □□□□ □□□□ □□□.

Answer: A (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation:

BDD emphasizes collaboration between developers, testers, and business stakeholders to define system behavior in a readable format (A). It typically uses the Given-When-Then syntax. Unlike unit testing (B), BDD is at a higher level of abstraction. It does not focus solely on non-functional testing (C) and encourages early testing rather than post-development validation (D).

Reference: ISTQB CTFL v4.0 Syllabus, Section 2.1.5 - Behavior-Driven Development (BDD)

NEW QUESTION: 26

□□ □ □□□□□ □□□□□ □□ □□□ □□□□ □□□ □□□ □□ □ □□□□ □□ □□ □□□?

- A. □□ □□□ □□□□ □□, □□, □□□□ □□□ □□□ □□□ □□, □□□□ □□□ □□ □□□ □□□ □□ □ □□□ □□□ □□□ □□□□.
- B. □□ □□□ □□ □□□ □□ □□, □□□□ □□□ □□ □□□ □□□ □□□ □□□ □□, □□□□ □□□ □□□□□ □□, □□, □□□□ □□□ □□□ □□□□.
- C. □□ □□□ □□□□ □□□ □□□□□ □□□□ □□ □□ □□□ □ □□□□.
- D. □□ □□□ □□ □□□ □□□□ □□□□ □□□ □□ □□□ □□□ □□□ □□□ □□□ □□□, □□□□ □□□ □□□ □□□□ □□ □□□ □□□ □□□ □□□□.

Answer: B (LEAVE A REPLY)

Product risk involves the potential issues that can affect the quality and functionality of the software product, such as defects, performance problems, and usability issues. Project risk, on the other hand, relates to the risks that can impact the project's schedule, budget, and resources, such as delays, cost overruns, and resource constraints. Understanding both types of risks is crucial for managing and mitigating potential problems in software projects.

References: ISTQB CTFL Syllabus, Section 5.2.1, "Risk Management in Testing."

NEW QUESTION: 27

□□□□□ □□□□ □□ □□ □□ □ □□□ □□ □□□□□?

- A. □□ □□□□ □□ □□ □□ □□ □□□□ □□□ □□□ □□□□ □□□ □□ □ □ □□□.
- B. □□□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□ □□ □□□ □ □□□□.
- C. □□ □□ □□ □□□□ □□ □□ □□ □□□ □□□□ □□ □□ □□ □□ □□□□ □□□ □□□□□□□ □□□ □ □□□□.
- D. □□□□ □□□□ □□ □□ □□□□□ □□□ □□□ □□□□ □□□ □□ □□□□ □□ □ □ □□□ □ □ □□□□.

Answer: C (LEAVE A REPLY)

Achieving full code coverage does not guarantee that the component or system is fully tested or free of defects. Code coverage metrics indicate the extent to which the source code has been tested, but they do not account for the quality of the tests or whether all possible scenarios have been considered. Other types of testing, including functional, performance, and security testing, are necessary to ensure comprehensive testing. The ISTQB CTFL Syllabus v4.0 highlights that while high code coverage is beneficial, it does not equate to complete testing.

NEW QUESTION: 28

□□ □□ □ □□□ □□□□□□ □□□ □□□ □□□ □□□□ □□□ □□□□ □□ □□□ □□□□ □□□?

- A. □□□ □□
- B. □□□ □□
- C. □□□ □□
- D. □□□ □□

Answer: B (LEAVE A REPLY)

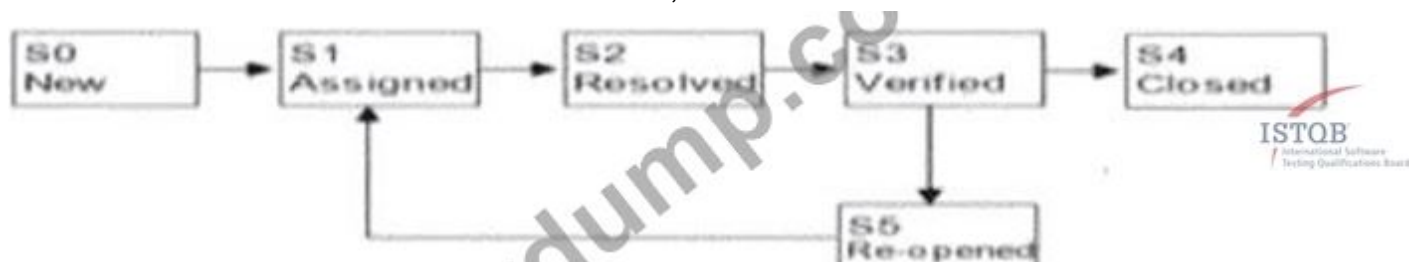
Comprehensive and Detailed In-Depth Explanation:

Test analysis involves identifying the features to be tested and deriving test conditions. It is the phase where testers analyze the test basis (e.g., requirements, specifications) to identify testable aspects of the system. Test design (A) focuses on creating test cases, test implementation (C) involves preparing the test environment, and test execution (D) runs the tests.

Reference: ISTQB CTFL v4.0 Syllabus, Section 1.4.1 - Test Analysis

NEW QUESTION: 29

□□ □□□ □□□ □□ □□□ □□□ □□□, □□ □□□□ □□□ □□□ □□ □□?



- A. S0->S1->S2->S3->S5->S1
- B. S0->S1->S2->S3->S5->S1->S2->S3

- C. S0->S1->S2~>S3->S4
 D. S0->S1 ->S2->S3->S5->S3->S4

Answer: D ([LEAVE A REPLY](#))

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0, the life cycle of a defect typically follows a sequence from its discovery to its closure. In the provided figure, it starts with S0 (New), moves to S1 (Assigned), then to S2 (Resolved), followed by S3 (Verified). If the defect is not fixed, it can be Re- opened (S5) and goes back for verification (S3). Once verified, it is Closed (S4). References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.4.3, Page 17.

NEW QUESTION: 30

□□ □ □□ □□□ □□□ □□ □□□□ □□□□ □ □□□□□ □□□□□?

- A. □□□ □□□ □□□ □□□ □□□□ □□□ □□□□□.
 B. □□□□ □□□□ □□ □□□ □□□ □ □□□ □□ □□□□ □□□□□.
 C. □□□□ □ □□ □□□ □□□ □□□ □□□ □□□□□□□.
 D. □□□ □□□ □□ □□□□ □□□□□ □□ □□ □□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

Comprehensive and Detailed In-Depth Explanation:

Exploratory testing is an unscripted approach where testers actively design and execute tests based on their intuition, experience, and knowledge of the system. A test charter (A) provides a guideline on what areas to explore, defining test objectives but leaving room for dynamic adaptation.

* (B), (C), and (D) contradict exploratory testing principles, which focus on freedom rather than rigid documentation.

* Exploratory testing is especially useful in agile environments, usability testing, and uncovering unexpected defects.

Reference: ISTQB CTFL v4.0 Syllabus, Section 4.4.2 - Exploratory Testing

NEW QUESTION: 31

□□□□ □□□ □□□ □□□□ □□□ □□□ □□ □□□□ □□□□ □□□□. □□ □□□
 □□ □□□ □□□ □□ □□(A, X □ Z)□ □□□ □□□□ □□ □□ □□□□□□. □□ "□□
 □□□" □□□□□ □□□ □□ □□□ □ □□□ □□□□□.

□ □□□ □□□□ □□ □□□□ □□ □□ □ □□□ □ □□ □□ □□□□□?

- A. □□ □□
 B. □□□ □□□ □□□
 C. □□□ □□ □□ □□
 D. □□ □□

Answer: B ([LEAVE A REPLY](#))

An incident report is a document that records the details of an incident. An incident report typically contains the following items:

* Identifier: A unique identifier for the incident report

- * **Summary:** A concise summary of the incident
 - * **Description:** A detailed description of the incident, including the steps to reproduce it, the expected and actual results, and any relevant screenshots or logs
 - * **Severity:** The degree of impact that the incident has on the system
 - * **Priority:** The level of urgency for resolving the incident
 - * **Status:** The current state of the incident, such as new, open, resolved, closed, etc.
 - * **Resolution:** The action taken to resolve the incident, such as fix, workaround, reject, etc. Based on the information given in the question, the tester would be able to write down all of these items except for the test case identifier. A test case identifier is a unique identifier for a test case that is used to link it to other test artifacts, such as test plans, test scripts, test results or incident reports. However, since the tester is performing exploratory testing, there is no predefined test case that can be associated with the incident. Exploratory testing is an approach to testing that emphasizes learning, test design and test execution at the same time. Exploratory testing relies on the tester's skills, creativity and intuition to explore the software under test and discover defects. Exploratory testing does not use formal test cases or scripts, but rather uses test charters or missions that guide the tester's actions and objectives.
- Verified References:** A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33; Chapter 5, page 47-48.

NEW QUESTION: 33

□□□ □ □□□□□□ □□□ □□ □□□ □□ □□□□□□.

□□ □□□ □□□ □□ □□□ □□□□□ □□□□□□?

□□ □□□□: 15 8.2010

□□ □□□: Joe Smith

□□□ □□ □□□ □□□

□□□ □□□: Area 5/TC 98

□□ □□: 2011-16.2

□□ □□ □□ 1□ □□ □□□ □□ □□□ □ ENTER□ □□□□ □□ 2□ □□□□□. □□ □
□ □□□□ □□, □□□ □□□ □□ □□□□.

A. □□ □ □□□□□ □□□□□□□□ □□ □□□ □□□□□.

B. □□□ □□□□ □□ □□□□ □□ □□□ □□□□□.

C. □□□ □□□ □□□□□□ □□□□□.

D. □□ □□ □□

Answer: A ([LEAVE A REPLY](#))

The most important report improvement for the given incident report would be to add information about which web browser was used when the defect was detected. This information is relevant for reproducing and debugging the defect, as different web browsers may have different behaviors or compatibility issues with the web application. The other options are less important or irrelevant for the incident report. The developer who should fix the bug can be assigned by the project manager or the defect tracking system, not by the tester who reports the defect. The time stamp when the incident happened is not very useful, as it does not indicate the cause or the frequency of the defect. The impact analysis is not part of the incident report, but rather of the risk assessment or prioritization process. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 140.

NEW QUESTION: 34

□□□□ □□ □□□ □□ □□ □□ □ □□ □□ □□□□□?

□□□ □□□ □□□ □□□□□ □□□□ □□□.

II □□ □□□□ □□ □□□ □□□ □□ □□□□ □□□.

III. □□□□ □ □□ □□□□ □□ □□□ □□□□ □□□ □ □□□□.

IV □□ □□□□□ □□□, □□□□, □□□ □□□□ □□□ □ □□□□.

V. □□□ □□□ □□□□□ □□□□ □□□.

A. 1, 3, 4

B. II, v

C. □, □

D. II. IV, V

Answer: ([SHOW ANSWER](#))

The following statements about re-testing and regression testing are true:

* I) Re-testing should be performed after a defect is fixed. Re-testing is a type of testing that verifies that a defect has been successfully resolved by executing a test case that previously failed due to that defect.

Re-testing should be performed after a defect is fixed and delivered to ensure that it does not cause any new failures or side effects.

* III) Re-testing and regression testing may be performed at any test level. Re-testing and regression testing are not limited to a specific test level, but can be applied at any level depending on the context and objectives. For example, re-testing and regression testing can be performed at unit level, integration level, system level or acceptance level.

* IV) Regression testing may include functional, non-functional and structural testing. Regression testing is a type of testing that verifies that previously tested software still performs correctly after changes.

Regression testing may include various types of testing depending on the scope and purpose of the changes. For example, regression testing may include functional testing to check if the software meets its requirements, non-functional testing to check if the software meets its quality attributes, or structural testing to check if the software meets its design or code standards. The following statement about re- testing and regression testing is false:

* II) Regression testing should always be performed after a defect is fixed. Regression testing is not always necessary after a defect is fixed, as some defects may have a low impact or low likelihood of affecting other parts of the software. Regression testing should be performed after a defect is fixed only if there is a risk of introducing new defects or causing existing defects due to the changes made to fix the defect. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 2, page 19; Chapter 4, page 45.

NEW QUESTION: 35

□□ □□□□ □□ □□□ □□□ □ □□□□ □□□ □□□ □□ □□ □□□□□?

A. □□□ □□□ □□□ □□

B. □□ □□ □□

C. □□ □□

D. □□ □□

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 36

5+6□ □□□ □□□□□ □□□ □□□□□□ □□□□□.

□□□□ □□□ □□□ 6□□ □□□□□□.

□□ □□□ □□□□.

A. □□

B. □□

C. □□

D. □□

Answer: D ([LEAVE A REPLY](#))

According to the ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 18, a failure is "an event in which a component or system does not perform a required function within specified limits". In this case, the calculator software does not perform the required function of calculating

The other options are incorrect because:

* A fault is "a defect in a component or system that can cause the component or system to fail to perform its required function" (page 16). A fault is not an event, but a defect, and it may or may not cause a failure. For example, a fault could be a logical error in the code, a missing specification in the design, or a contradiction in the requirement.

References = ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 15-18, 25; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 96, page 34.

[illegible]

- ☐☐☐ ☐☐☐☐ ☐☐☐, ☐☐ ☐☐☐☐ ☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐☐ ☐☐
- ☐?

- Answer: B (LEAVE A REPLY)**

When writing a defect report, the tester can specify the expected result and the actual result based on the observation. The expected result is what the requirements specify, and the actual result is what was observed during testing. These elements are crucial for clearly communicating the nature of the defect to developers and other stakeholders. The other items such as test

environment, test level, and root cause may not be clear or necessary at this stage of defect reporting.

References:ISTQB CTFL Syllabus, Section on defect management and reporting.

NEW QUESTION: 38

□□ □ □□□ □□□□ □□ □□□ □□□□ □□ □□□ □□□□□?

- A. □□□ □□□ □□□ □□
- B. □□□□ □ □□□ □□□□ □□
- C. □□ □□ □□ □□□ □□□ □□ □□
- D. □□ □□ □□ □□□□ □□□ □□ □□

Answer: C ([LEAVE A REPLY](#))

This answer is correct because identifying potential anomalies in the work product under review is one of the tasks the Author is responsible for, as part of a typical formal review. The Author is the person who creates the work product to be reviewed, such as a requirement specification, a design document, or a test case. The Author's tasks include preparing the work product for the review, identifying potential anomalies in the work product, and fixing the anomalies found in the work product after the review. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.4.2.1

NEW QUESTION: 39

□□□□ □□□ □□□ □□□□□ □□□□ □□ □ □□□□□?

- A. OTS(□□ □□ □□□)□□ □□□ □□□ □□ □□□ □ □□□ □□ □□
- B. □□□ □□ □□(S)□ □□□□ □□□□ □□□□ □□□ □□ □□(S) □□□ □□ □□□□ □□□□□ □□□□ □ □□□ □□ □□
- C. □□ □□□□ □□ □□ □□□□ □□ □□□□ □□ □□□ □□□□□ □□□ □ □□□ □ □□□□□.
- D. □□ □□(S) □□□ □□□□ □□ □□□□□□ □□□ □□□□ □□

Answer: B ([LEAVE A REPLY](#))

The purpose of a review is to find defects and improve the quality of the work product, not to criticize or blame the author. Reporting defects objectively means describing them factually and constructively, without using negative or emotional language that could offend the author or damage their motivation. This way, the author can take the feedback positively as an effort at improving the product and not as a personal assault.

Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 138.

NEW QUESTION: 40

□□ □□ □ □□ □□ □□ □□ □□□ □□ □□□ □□□□ □□ □□ □□□ □□□□ □□□ □□□ □□□□□?

- A. □□ □□□, □□ □ □□□
- B. □□□□, □□□□ □ □□□

C. ☐☐☐☐ , ☐☐ ☐ ☐☐

D. ☐☐☐☐ , ☐☐ ☐ ☐ ☐☐

Answer: ([SHOW ANSWER](#))

Comprehensive and Detailed In-Depth Explanation:

Business, development, and testing (C) are the three key perspectives in agile user story refinement:

* Business (Product Owner, Stakeholders) - Ensures the story aligns with user needs and business goals.

* Development (Developers) - Provides insights on feasibility and technical constraints.

* Testing (Testers, QA) - Ensures testability, acceptance criteria, and risk identification.

* (A) is incorrect because the product owner is part of business stakeholders.

* (B) is incorrect because architecture is part of development, but not the primary driver of user stories.

* (D) is incorrect because acceptance testing is a process, not a stakeholder group.

A cross-functional team collaboration ensures well-defined, testable, and feasible user stories.

Reference: ISTQB CTFL v4.0 Syllabus, Section 2.1.1 - Agile Team Collaboration in Testing

NEW QUESTION: 41

☐☐ ☐ ☐☐ ☐☐:

A. ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐.

B. ☐☐ ☐☐ ☐☐ ☐ ☐ ☐ ☐☐ ☐ ☐ ☐ ☐☐.

C. ☐☐ ☐☐ ☐ ☐☐ ☐☐ ☐ ☐☐ ☐☐ ☐☐.

D. ☐☐ ☐☐ ☐ ☐☐ ☐☐ ☐☐ ☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐.

Answer: C ([LEAVE A REPLY](#))

This answer is correct because the whole-team approach is a way of working in agile projects where all team members share the responsibility for the quality of the product, and collaborate on delivering value to the customer. The whole-team approach involves testers, developers, business analysts, product owners, and other stakeholders in planning, designing, developing, testing, and delivering the product. The whole-team approach fosters communication, feedback, learning, and continuous improvement within the team. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.1

NEW QUESTION: 42

☐☐☐ ☐☐☐☐ 4~7 ☐☐☐ ☐☐ , ☐☐ 1 ☐☐ ☐☐ , ☐☐☐ 1 ☐☐ , ☐☐☐ ☐☐ 1 ☐☐ ☐☐ 1 ☐☐ ☐☐☐.

☐☐ ☐☐ ☐ ☐ ☐ ☐ ☐ ☐☐ ☐☐ ☐☐ 2 ☐☐ ☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐? (☐☐: ☐☐ ☐☐ ☐☐☐☐ ☐☐)

A. 1xA;aB11;Pq1ZZab;7iDD0a1x

B. aB11;99rSp:5NnN10;7iDD0a1x

C. 1xB: aB11: 99rSp: 5NnN10; 4NnN10T; 44ghWn19

D. 1RhT;rSp53;3N3e10;8sBdbby

Answer: D ([LEAVE A REPLY](#))

The correct outcome of a two-value boundary value analysis applied to the password length is the set of test cases represented by option D. Boundary value analysis is a test design technique that focuses on the values at the boundaries of an equivalence partition, such as the minimum and maximum values, or the values just above and below the boundaries. A two-value boundary value analysis uses two values for each boundary, one representing the valid value and one representing the invalid value. For example, if the valid range of values is from 4 to 7, then the two values for the lower boundary are 3 and 4, and the two values for the upper boundary are 7 and 8. The test cases in option D use these values for the password length, while also satisfying the other requirements of the password, such as containing at least one numeric character, one capital letter, and one lowercase letter. The test cases in option D are:

* 1RhT: a 4-character password that is valid

* rSp53: a 5-character password that is valid

* 3N3e10: a 6-character password that is valid

* 8sBdbby: an 8-character password that is invalid The test cases in the other options are incorrect, because they either use values that are not at the boundaries of the password length, or they do not meet the other requirements of the password. For example, the test cases in option A are:

* 1xA: a 3-character password that is invalid, but it does not contain a capital letter

* aB11: a 4-character password that is valid

* Pq1ZZab: a 7-character password that is valid

* 7iDD0a1x: an 8-character password that is invalid

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.1, Black-box Test Design Techniques¹

* ISTQB Glossary of Testing Terms v4.0, Boundary Value Analysis, Equivalence Partition²

NEW QUESTION: 43

□□ □ □□□ □□□ □□ □ □ □□□□?

A. □□□□□ □ □

B. □ □

C. □□ □□ □ □ □□□□ □□

D. □□ □□ □ □ □

Answer: C ([LEAVE A REPLY](#))

Debugging the software to find the reason for defects is not a common objective of testing, but rather a task of development or maintenance. Debugging is a process of locating and fixing errors in the software code, while testing is a process of finding and reporting defects in the software behavior or quality. Testing does not aim to fix defects, but rather to provide information on their existence and impact. The other options are common objectives of testing. Finding defects in the

software is one of the main objectives of testing, as it helps to improve the quality and reliability of the software. Preventing defects is another objective of testing, as it helps to avoid rework and reduce costs and risks. Providing information on the status of the system is another objective of testing, as it helps to support decision making and risk management. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 3.

NEW QUESTION: 44

□□□ □□□□□ □□□□□□□□ □□□□ □□□□□ □□ □□ □□ □□□ □□ □□□ □□ □□□ □□□ □□□□. □□□□ □□□ □□(□□ □□ □□ □□□□ □□ □□ □□□ □□ □□ □□)□ □□□□□. □□ □□ □□ □□□ □□□ □□□□□ □□ □□□ □□□□ □□□□□. □□□ □□□ □□, □□□ □ □□□ □□ □□ □□ □□□ □□□ □□ □□□□. □ □□ □ □□ □□□ □□□ □□ □□□□ □□□ □□□□ □□□□□ □□□□□□ □□□□ □□□□ □□□□ □□□□.

□□ □ □ □□□□□ □□ □ □□□□ □□ □□□□□?

- A. □ □□□□□ □ □□ □□□ □□ □□ □□ □□□ □□□□□.
- B. □ □□□□□□□ □□□□ □□ □□□ □□□□□ □□ □□ □□□ □□□□□.
- C. □ □□□□□□□ □□□□ □□ □□□ □□ □□ □□ □□□ □□□□□.
- D. □ □□□□□ □ □□ □□□ □□□□□ □□ □□ □□□ □□□□ □□□□.

Answer: ([SHOW ANSWER](#))

This scenario is using a walkthrough review type and a perspective-based review technique. In a walkthrough, the author of the document leads the meeting and it typically includes a meeting leader and a scribe, as described. This type of review is informal, focuses on discussion, and often involves scenario-based reading of the document to understand different user perspectives (ISTQB Main Web).

References:

ISTQB Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION: 45

□□□ □□ □ □□ □□□ □□□□.

1. □□□ □□□ □□ □□□□□ □□□□□□□.
 2. □□□ □□□ □□□ □□□□ □□□□ □□□□□□□.
 3. □□□ □□□ □□ □□ □□□ □□ □□ □□□□.
 4. □□ □□□ □□□ □□□□□□ □□ □□□□□.
 5. □□□ □□□□□ □□□ □□ □□□ □□□□ □□□□.
 6. □□□ □□□ □□□□□ □□□□ □□□□ □ □□ □□□□ □□ □□ □□□ □□□□□.
- □ □□ □□ □□ □ □□ □□□□ □□ □ □□□□□□?

- A. □□□□ - 2, 4, 5 □□□□ -1, 3, 4
- B. □□□□ - 2, 4 □□□□ - 2, 4, 5, 6
- C. □□ □□ -1,3,6 □□ □□ - 2, 5. 6
- D. □□ □□ -1,3,56 □□ □□ -1,3

Answer: A ([LEAVE A REPLY](#))

Entry and exit criteria are used to determine when to start and stop testing, respectively.

* Entry Criteria:

* These are conditions that should be met before testing begins.

* Examples:

* 2. The test automation tool has been installed and properly configured

* 4. The performance test environment has been set-up and is available

* 5. The user stories have proper acceptance criteria defined

* Exit Criteria:

* These are conditions that should be met before testing concludes.

* Examples:

* 1. A defined level of code coverage has been achieved

* 3. The number of unresolved defects is within the predefined limit

* 6. The testing budget (ISTQB not-for-profit association) spent and the project sponsor bears the risk of not testing any further According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, these criteria help in effectively managing the testing process#6source#.

References:

Certified Tester Foundation Level v4.0

ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 46

□□ □ □□□ □□□ □□ □□ □□□□□?

A. □□□ □□□□□ □□□□□□ □□

B. □□ □□□ □ □□ □□

C. □□ □□

D. □□□ □□□ □□□□□ □□ □□□□ □□□□ □□ □□□□□ □□□□□.

Answer: C (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation:

The primary objectives of testing, as outlined in the ISTQB CTFL v4.0 syllabus, include verifying whether specified requirements are met (A), detecting failures and defects (B), and validating that the test object functions as expected (D). However, "finding errors" (C) is not a direct objective. Errors result from human mistakes, but testing primarily identifies defects, which are flaws in the system that cause failures. Testing aims to reveal defects rather than directly identify errors in the code.

Reference: ISTQB CTFL v4.0 Syllabus, Section 1.1.1 - Test Objectives

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □

D. □□□□□ □□□ □□□ □□□□ □□□ □□□ □□□□□ □ □□□□□ □□□ □□□□
□□□□.

□□ □ □□□ □□□ □□□ "□□ □□□"

Answer: B ([LEAVE A REPLY](#))

The features of the test object to be tested and those excluded from the testing represent information that is usually included in a test plan and, in the given test plan, it is more likely to be specified within "Test Scope" rather than in the other two sections mentioned. The test scope defines the boundaries and limitations of the testing activities, such as the test items, the features to be tested, the features not to be tested, the test objectives, the test environment, the test resources, the test assumptions, the test risks, etc. The test scope helps to establish a common understanding of what is included and excluded from the testing, and to avoid ambiguity, confusion, or misunderstanding among the stakeholders. The other two sections, "Testing Communication" and "Stakeholders", are also important parts of a test plan, but they do not directly address the features of the test object. The testing communication describes the methods, frequency, and responsibilities for the communication and reporting of the testing progress, status, issues, and results. The stakeholders identify the roles and responsibilities of the people involved in or affected by the testing activities, such as the test manager, the test team, the project manager, the developers, the customers, the users, etc. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1

* ISTQB Glossary of Testing Terms v4.0, Test Plan, Test Scope2

NEW QUESTION: 49

□□ □ □□□ □□ □□□ □□□□ □□□ □□□□ □□ □□ □□ □□□□□?

A. □□□ □□

B. □□□□ □□□ □□□ □□

C. □□□□ □□□ □□ □□□□ □□□

D. □□□□ □□ □□□ □

Answer: A ([LEAVE A REPLY](#))

Test exit criteria are the conditions or requirements that must be met before testing can be concluded. Test exit criteria are usually defined in the test plan and agreed by the stakeholders. Test exit criteria can be based on various factors, such as test coverage, defect status, quality level, risk level, etc. Test schedules would be the least likely to be used as the basis for test exit criteria, because test schedules are not directly related to the quality or performance of the software product, but rather to the time or resources allocated for testing. Test schedules can be used as the basis for test entry criteria, which are the conditions or requirements that must be met before testing can start. The other options are more likely to be used as the basis for test exit criteria. Cost of testing performed so far can be used as a basis for test exit criteria, because it can indicate the return on investment or the cost-benefit ratio of testing. Confidence of testers in tested code can be used as a basis for test exit criteria, because it can reflect the level of satisfaction or assurance of the testers about the quality or reliability of the software product. Number of unfixed defects can be used as a basis for test exit criteria, because it can indicate the

level of risk or impact of the remaining defects on the software product. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 13.

NEW QUESTION: 50

□□□ □□□ □□□ □□ □□ □ □□□□ □□□□ □□ □□□ □□ □□ □□ □ □□□ □□ □□ □□□□□?

A. □□□□ □□□ □□ □□□ □□□ □□ □□□□ □□□ □□□ □□□□ □ □□□ □ □□ □□.

B. □□□□ □□ □□□□□ □□□ □□ □□□ □□ □□□ □□□□□□ □□□□ □ □□□ □ □□□□.

C. □□□□ □□□ □□□□□□ □□□ □□ □□□ □□□ □□□ □□□□ □ □□□ □ □□ □□.

D. □□□□ □□□ □□□ □□□ □□ □□□ □□□□ □ □□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

Traceability is the ability to trace the relationships between the items of the test basis, such as the requirements, the design, the risks, etc., and the test artifacts, such as the test cases, the test results, the defects, etc. Traceability can provide various benefits for the testing process, such as improving the test coverage, the test quality, the test efficiency, and the test communication.

However, not all the statements given are true about the value of maintaining traceability between the test basis and test work products. The statement that is not true is option C, which says that test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly. This statement is false, because test objectives are the goals or the purposes of testing, which can vary depending on the test level, the test type, the test technique, the test environment, the test stakeholder, etc. Test objectives can be defined in terms of the test basis, the test coverage, the test quality, the test risk, the test cost, the test time, etc. Test objectives should be specific, measurable, achievable, relevant, and time-bound, and they should be aligned with the project objectives and the quality characteristics. Test objectives should not be the same for all test levels, as different test levels have different focuses, scopes, and perspectives of testing, such as component testing, integration testing, system testing, and acceptance testing. The other statements are true about the value of maintaining traceability between the test basis and test work products, such as:

* Traceability can be useful for assessing the impact of a change to a test basis item on the corresponding tests: This statement is true, because traceability can help to identify which tests are affected by a change in the test basis, such as a new requirement, a modified design, a revised risk, etc., and to determine the necessary actions to update, re-execute, or re-evaluate the tests. Traceability can also help to estimate the effort, the cost, and the time needed to implement the change and to verify its impact on the software system.

* Traceability can be useful for determining how many test basis items are covered by the corresponding tests: This statement is true, because traceability can help to measure the test coverage, which is the degree to which the test basis is exercised by the test cases. Traceability can help to identify which test basis items are covered, partially covered, or not covered by the

tests, and to evaluate the adequacy, the completeness, and the effectiveness of the testing process. Traceability can also help to identify the gaps, the overlaps, or the redundancies in the test coverage, and to prioritize, optimize, or improve the test cases.

* Traceability can be useful to support the needs required by the auditing of testing: This statement is true, because traceability can help to provide evidence, documentation, and justification for the testing activities, results, and outcomes. Traceability can help to demonstrate that the testing process follows the standards, the regulations, the policies, and the best practices that are applicable to the software system, the project, or the organization. Traceability can also help to verify that the testing process meets the expectations, the needs, and the satisfaction of the users and the stakeholders. References:

ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches¹

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.2, Test Monitoring and Control1

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design1

* ISTQB Glossary of Testing Terms v4.0, Traceability, Test Basis, Test Artifact, Test Objective, Test Level, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time2

NEW QUESTION: 51

[illegible]

A. □□□□□ □□□ □□ □□ □□ □□□ □□□ □ □□ □□ □□ □□ □□□ □□□□□.

B. □□ □□□ □□□□ □□□□ □□ □□□□□□.

C. □□□□□ □□□ □□□□ □□ □□□ □□ □□ □□□□ □□□□ □□ □□ □□ □□□ □□
□□□.

D. □□□□□ □□□ □□ □ □□ □□□ □□□ □ □□ □ □□ □□ □□□ □□□□□.

Answer: C (LEAVE A REPLY)

Statement coverage is a metric used in white-box testing that measures the percentage of executable statements in the code that have been executed by the test cases. It is calculated as the number of statements executed by the tests divided by the total number of executable statements in the code, providing an indication of how much of the code has been tested.

NEW QUESTION: 52

□□ □□ □□ □□□ □□ □□ □□ □□ □□ □□□□□?

A. □□□□□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□
 □□□ □□□□ □□□ Agile/DevOps □□□□□□□□ □□□ □ □□□□.

B. □□ □□ □□□ □□ □□□□ □□ □□□□ □□□□ □□ □□□□ □□ □□□ □□
 □□ □□□ □□ □□ □□□□□ □□□□ □□□□ □□□□ □□ □□□ □□□□ □□□□ □□□□
 Shift-Left□□□.

C. □□□□□ □□□□ □□□ □□ □□ □□□ □□□ □□□ □□ □□ □□ □□ □□□□ □
□□ □ □□□ □□□□ □□□ □□□ □□□ □□ □ □□□ □□ □□□□.

D. □□□□ □□□ □□□ □□□ □ □□□□ □□□□ □□□ □ □□□□ □□□□□ □□ □□□ □□□□ □□□ □□□□ □ □□ □□□ □□ □ □□□□.

Answer: D (LEAVE A REPLY)

This answer is correct because shift-left in testing is an approach that aims to perform testing activities as early as possible in the software development lifecycle, in order to find and fix defects faster and cheaper, and to improve the quality of the software product. Continuous integration is a practice that supports shift-left in testing, as it involves integrating and testing the software components frequently, usually several times a day, using automated tools and processes. Continuous integration can reduce the time between the introduction of a defect and its detection, thereby reducing the cost to fix it and the risk of accumulating defects that could affect the functionality or performance of the software product. References: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.3, Section 3.2.1.3

NEW QUESTION: 53

[illegible]

A. □□□□□ □□□ □ □□ □□ □□□□ □□ □□ □□□□ □□ □□ □□□ □□□ □ □□ □□□.

B. ☐☐ ☐☐☐☐ ☐☐☐☐☐ ☐☐ ☐☐☐☐☐ ☐☐ ☐☐☐☐.

C. □□□ □□□ □□□□ □ □□□ □□ □□ □□□□ □□ □□□□ □□
□□□ □ □□ □□□ □□□□ □ □□□ □□□.

D. □□ □□□□ □□ □□□□ □□ □□□□ □□□ □□□ □□□□ □ □□□ □ □□
□□.

Answer: D (LEAVE A REPLY)

This answer is correct because static testing and dynamic testing are both types of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc. Static testing is a type of testing that involves the analysis of software work products, such as requirements, design, code, or test cases, without executing them. Dynamic testing is a type of testing that involves the execution of software work products, such as code or test cases, using inputs and verifying outputs. Both static testing and dynamic testing can be applied to different test levels and test types, and can use different test techniques and tools, to evaluate the non-functional characteristics of the software product. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.1.1, Section 2.2.1.2

NEW QUESTION: 54

☐☐☐ ☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐

* ->

* ->

* $\square\square\square \rightarrow \square\square\square$

* □□ -> □□

* □□ -> □□

* □□□ -> □□□

□□ □□ □□□ □ □□ □□ □□□ □□ □□ □□□ □□□□□(□□ □□□ □□ □□□ □ □□□ □□□ □)?

A. □□ □ -> □□□ -> □□□ -> □□ □ -> □□□ -> □□□

B. □□ □ -> □□ -> □□ □□ -> □□ -> □□ -> □□ □

C. □□□ -> □□ □ -> □□□ -> □□□ -> □□□

D. □□□ -> □□□ -> □□□ -> □□□ -> □□□ -> □□ □ -> □□□

Answer: (SHOW ANSWER)

State transition testing is a black-box testing technique where test cases are designed to cover states and transitions of a state machine.

Given the state model with the following transitions:

* PLACED -> IN PRODUCTION

* IN PRODUCTION -> CANCELLED

* IN PRODUCTION -> SHIPPED

* SHIPPED -> INVOICED

* INVOICED -> CANCELLED

* CANCELLED -> PLACED

To cover all transitions at least once, we need to create a sequence that covers all six transitions.

Option A: IN PRODUCTION -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED -> PLACED

- Misses SHIPPED -> INVOICED and INVOICED -> CANCELLED transitions.

Option B: IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION

- Covers all transitions.

Option C: PLACED -> IN PRODUCTION -> SHIPPED -> CANCELLED (ISTQB not-for-profit association) (Udemy)sses INVOICED -> CANCELLED transition.

Option D: PLACED -> CANCELLED -> PLACED -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED

- Misses SHIPPED -> INVOICED and INVOICED -> CANCELLED transitions.

Given these, Option B covers all the transitions#6source##9source#.

References:

Certified Tester Foundation Level v4.0

10 Sample Exams ISTQB Foundation Level (CTFL) v4.0

NEW QUESTION: 55

□□ □□ □□□ □□□ □□□□□ CONTROL FLOW□ □□□ □□ □□□ □□□□□. □□ □□ □ □□ □□ □□□ □□ □□□ □□ □□□ □□□□□?

A. □□□ □ □□ □□ □□

B. □□ □□ □□□ □□ □□□

C. □□□□ □□

D. □□□□ □ □

Answer: A ([LEAVE A REPLY](#))

A static analysis tool is a tool that analyzes a given program's source code or executable code without executing it. A static analysis tool can perform various types of analysis on a program's code, such as syntax checking, data flow analysis, control flow analysis, complexity measurement, coding standards compliance checking, etc. Control flow analysis is a type of analysis that examines how a program's statements are executed in different paths or branches. One of the most likely outcomes of control flow analysis is identification of unreachable code, which is code that can never be executed due to logical errors or design flaws. Unreachable code can reduce readability and maintainability of the code, as well as increase complexity and size. The other options are not outcomes of control flow analysis, but rather outcomes of other types of analysis. Report on adherence to coding standards is an outcome of coding standards compliance checking. Number of comment lines and number of source code lines are outcomes of complexity measurement. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 8.

NEW QUESTION: 56

□□ □ □□ □□ □□□ □□ □□□ □□ □□ □□□□□?

A. □□ □□ □□□□□□ □□ □□□ □□□ □□□□□ □□□□□.

B. □□ □□□□ □□□ □□ □□ □□□ □□ □□ □□ □□□□□.

C. □□□□□ ell □□ □ □□ stale□ □□□□ □□□□ □□□ □□ □□□□□□.

D. □□ □□ □□ □□□ □□ □□ □□□□□ □□□□□.

Answer: ([SHOW ANSWER](#)**)**

State transition testing is a black-box testing technique used to analyze the behavior of a system by examining the transitions between different states in response to events. In state transition testing, a state table or diagram is used to represent the states of a system and the transitions between these states triggered by events.

Option D is correct because in state transition testing, all transitions between states should be explicitly shown in the state table. This includes valid transitions that the system is expected to make under normal operation and, where relevant, invalid transitions that should be tested to ensure the system handles unexpected or erroneous inputs gracefully. The state table provides a comprehensive view of how the system should behave, making it possible to create tests that cover all defined transitions.

NEW QUESTION: 57

□□ □ □□□□□ □□ □□□□ □□ □ □□□□ □□□ □□□□□?

A. □□□□□□□ □□□□□□ □□ □□□ □□□□□□ □□□□ □□□□ □□□ □□□□ □□□□ □□□□.

B. □□□□□□□ □□□□□ □□ □□□ □ □□ □□□□ □□□ □□□ □□□□□ □□□.

C. □□□□ □□□□□ □□ □□□□□□ □□ □□□□ □□□ □□□ □□□□□ □□□.

D. □□□□□□ □□ □□□□□ □□□□ □□, □□□ □□□□□ □□□□ □□□□.

Answer: B (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation:

Checklist-based testing uses specific test conditions (B) that help testers ensure key aspects are validated.

The checklist items can be derived from past defects, requirements, or regulatory standards.

* (A) is incorrect because general guidelines lack specificity.

* (C) is incorrect because checklists can be used for both manual and automated testing.

* (D) is incorrect because checklists are useful for both functional and non-functional testing (e.g., security, performance).

Checklists help ensure completeness without enforcing strictly scripted execution.

Reference: ISTQB CTFL v4.0 Syllabus, Section 4.4.3 - Checklist-Based Testing

NEW QUESTION: 58

□□□□□ □□□□□ □□□ □□ □□□ □□□□ □□□ □□ □□□ □□□ □□□. □□□ □□ □□ □□ □□□ □□□□□, □□□□ □□ □□ □□ □□□□□ □□□ □□□ □□□ □□□□□. □□□ □□□□ □□□ □□ □□□ □□□ □□□, □□ □□ □□ □□□□ □□□□□ □□□ □□□□ □□□□□. □ □□□ □□□ □□□ □□□□□ □□ □□□□ □□ □□□ □□□□□ □□□□□ □□ □□□ □□□□□ □□□□□ □□ □□□ □□□ □□□□ □□□□.

□ □□□ □□ □□ □□ □ □□ □□ □□□□?

A. □□

B. □□

C. □□□□

D. □□□ □□

Answer: C (LEAVE A REPLY)

This answer is correct because a walkthrough is a type of review where the author of the work product leads the review process and explains the work product to the reviewers. The reviewers are not required to prepare for the review in advance, and the main objective of the walkthrough is to establish a common understanding of the work product and to identify any major defects or issues. A walkthrough is usually informal and does not follow a defined process or roles. In this case, the review for a high-level architectural document written by a software architect matches the characteristics of a walkthrough. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.4.2.2

NEW QUESTION: 59

□□ □ □□□ SDLC(□□□□□ □□ □□ □□) □□□ □□□ □□ □□□ □□□ □□□□□?

A. □□□□ □ □□□ □□ □□□ □□□ □□ □□ □□□ □□□□ □□□ □□ □□□□ □□ □□□ "□□□□ □□" □□□ □□□ □ □□□ □□□.

B. □□ □□□ □□□ □□ □□□ □□□ □□□ SDLC□ □□ □□ □□□□ □□□□□, □□ □□ □□ □□□ □□□ □□ □ □□□□.

C. □□ □□□□□ □□ □□□□ □□ □□□ □□□ □□□, □□ □□ □□ □□ □□□ □□ □□□ □□□□□.

D. □□□ □□□ □□□□ □□□□□ □□□ □□□ □□□ □□ □□□□ □□□□□ □□ □ □□ □□ □□□ □□□ □□□ □□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

Comprehensive and Detailed In-Depth Explanation:

Different test levels (unit, integration, system, acceptance) have specific objectives while sometimes overlapping to ensure comprehensive testing (D). Option A contradicts shift-left principles, B encourages late testing (which is ineffective), and C incorrectly assumes that all development activities require direct test actions.

Reference: ISTQB CTFL v4.0 Syllabus, Section 2.2 - Test Levels

NEW QUESTION: 60

□□ □□ □□□ "X□ Y□ Z□ □□□ □□□□□"□□□.

□□ □ □□ □□□□ □□ □□□ □□□□.

□□ □□ □ □□ □□□ □□ □□ □ □□ □□□ □□□□□?

A. int x = 1.

int y = 2.

int y = 3.

X = y=z;

B. int x = 1.

int y = 2.

int z = 3.

X = z-y

C. int x = 1.

Int y = 2.

Int z = 3.

Z = x +y

D. int y = 2

Int z = 3.

Y = z+y

Answer: A ([LEAVE A REPLY](#))

Static analysis is a technique that analyzes the source code or other software artifacts without executing them.

Static analysis can detect defects such as syntax errors, coding standards violations, potential security vulnerabilities, or logical flaws. Static analysis can catch the bug in the first implementation, as it contains two syntax errors: the variable y is declared twice, and the assignment statement X = y=z is invalid. Static analysis cannot catch the bugs in the other three implementations, as they are logical errors that do not violate any syntax rules, but produce incorrect results. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 3, page 25-26.

□□□ □□□□ □□□ □ □□□ □□ □□□□□ □□□□□ □□ □□□ □□ □□ □□□
 □□□□ □□□□□□. □ □□□□□ □□ □□□□□ □□□□ □□□ □□□□ □□□□ □□□□
 □. □□□ □□□ □□□□ □□ □□□ □ □□ □□ □□□□□.

[b] □□ □□□ □□□□□ □ □□□□ □□ □□□ □□ □□□ □□□ □□ □□□□ □□□ □
□□□ □□□.

[d] □□□□□ □□□ □□ □□□ □□□ □□□□ □□ □□□ 2~80□□□ □□□. □□□ □□
□/□□ □□ □□□ □□ □ □□□□□ □□□ □□ □□ □□□□ □□□□□ □□□.

A. □□ □□□□ □□ □□□ □□□ □□ □□□□□. □, □□□ □□□ □□□□□. □
□□ □□□ □□ □□ □□□ □□ □□□□□ □□□□ □□□ □□□ □□□□ □□□ □□□
□□□ □□ □ □□□□. □□ □□ □□□□□□ □ □□□ □□ □□□□□.

C. □□□□ □□□ □□□ □□□□□ □□□□□ □□□□ Mastercard □□ □□□ □□□□ 5
□ □□□ □□ □□ □□□ □□□□□ □□□□□□□.

Answer: D (LEAVE A REPLY)

* The new user provides all valid data.

This test covers the basic functionality and will help verify that the primary use case is handled correctly before testing invalid or edge cases.

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-

CTFL-KR ☐☐ ☐☐☐ ☐☐☐☐☐☐☐☐☐ ☐☐☐ ☐☐☐☐☐☐☐. ☐☐☐☐ ☐☐☐ ☐☐☐☐ ☐☐
☐ DumpTop ISTQB-CTFL-KR ☐☐☐ ☐☐☐☐☐. [https://www.dumptop.com/ISTQB/ISTQB-](https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html)
[CTFL-KR-dump.html](https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html) (**409 Q&As Dumps**, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 62

□□ □□ □ □□□ □□ □□□ □□□ □□ □□ □□□□□?

- A.** □□□□ □□ □□ □□
- B.** □□ □ □□□□□ □□
- C.** □□ □□□□ □□□ □□□□□□.
- D.** □□ □□□ □□

Answer: C (LEAVE A REPLY)

Technical reviews are structured meetings that aim to examine various aspects of a product or project to identify any defects or improvements. Options A (Pre-meeting preparation by reviewers), B (Using checklists during the meeting), and D (Preparation of a review report) are typical activities in a technical review process.

Inviting end-users to the meeting (C), however, is generally not part of a typical technical review, as these reviews are usually more focused on the technical aspects and are conducted by peers or experts within the development or testing teams rather than end-users.

NEW QUESTION: 63

[illegible]

- A.** □□ □□□ □□ □□□□□□ □□□□ □□□ □□ □□□ □□ □□□ □□□□□□ □□
□□ □□ □□□ □□ □□□ □□ □□□ □□□□ □□□ □□□ □□□ □□□ □□□.
- B.** □□ □□□ □□ □□□□□ □□ □□□ □□□ □□ □□□ □□□□ □□□□□ □□ □□
□ □□□□ □□ □□□□□.
- C.** □□ □□□ □□□□ □□□□□□ □□□□ □□ □□□ □□□ □□ □□□□, □□□□□
□□□ □□□ □□ □□□ □□ □□□□ □□□□ □□□ □ □□□□.
- D.** □□ □□□ □□ □□ □□ □□□□ □□ □ □□□□□ □□□ □□ □□□□ □□□□ □□
□□ □□□ □□□ □□□□ □□□□ □□□ □ □□□ □□□□ □ □□□ □□□.

Answer: D (LEAVE A REPLY)

This answer is correct because configuration management is a process of establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation. This supports testing by providing traceability, consistency, and control over the test artifacts and the software under test. References: : ISTQB Glossary of Testing Terms v4.0, : ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.2

NEW QUESTION: 64

□ ?

- A. Checklist-based testing involves using checklists that contain items, such as potential test conditions, that should be tested. These checklists are often based on insights into what is important to the user, potential areas where software might fail, and specific aspects that need to be tested. It provides a structured yet flexible approach to testing, ensuring key areas are covered while allowing testers to use their experience and understanding of the system. Checklist-based testing is not limited to non-functional testing but can be applied to various types of testing, including functional testing.
- B. Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.
- C. State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques
- D. Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.

Answer: B ([LEAVE A REPLY](#))

Checklist-based testing involves using checklists that contain items, such as potential test conditions, that should be tested. These checklists are often based on insights into what is important to the user, potential areas where software might fail, and specific aspects that need to be tested. It provides a structured yet flexible approach to testing, ensuring key areas are covered while allowing testers to use their experience and understanding of the system. Checklist-based testing is not limited to non-functional testing but can be applied to various types of testing, including functional testing.

References:

ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4.5.

NEW QUESTION: 65

Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.

- A. Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.
- B. Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.
- C. Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.
- D. Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.

Answer: ([SHOW ANSWER](#))

Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.

NEW QUESTION: 66

State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques

- A. State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques
- B. State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques
- C. State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques
- D. State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques

Answer: A ([LEAVE A REPLY](#))

State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques

require the system to be executed with test cases, whereas static testing techniques do not. Static testing techniques can be applied before the code is ready for execution, such as reviews, inspections, walkthroughs, and static analysis. Static testing techniques can help find defects early in the development process, improve the quality of the code, and reduce the cost and effort of dynamic testing. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 4, Section 4.2.1, Page 281; ISTQB Glossary of Testing Terms v4.0, Page 292

NEW QUESTION: 67

□□ □□ □ □□□□ □□□ □ □□ □□□ □□□□□ □□□□ □□ □□ □□□□□?

- A. □□□□ □□□ □□□□ □□□□ □□□□ □□ □□□ □□□□.
- B. □□□□□ □□□□ □□□ □□□ □□□□□.
- C. □□□□□ □□□□ □□□ □□□□ □□ □□□ □□□□ □ □□□ □ □□□□.
- D. □□□□ □□□□ □□ □□ □□ □□□ □□□□ □□□ □□□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

* The testing of software does not demonstrate the absence of defects, but rather the presence of defects or the conformance of the software to the specified requirements¹. Testing can never prove that the software is defect-free, as it is impossible to test all possible scenarios, inputs, outputs, and behaviors of the software². Testing can only provide a level of confidence in the quality of the software, based on the coverage, effectiveness, and efficiency of the testing activities³.

* The other options are correct because:

* A. Properly designed tests that pass reduce the level of risk in a system, as they verify that the system meets the expected quality attributes and satisfies the needs and expectations of the users and clients⁴. Risk is the potential for loss or harm due to the occurrence of an undesirable event⁵. Testing can help to identify, analyze, prioritize, and mitigate the risks associated with the software product and project⁶.

* C. Software testing identifies defects, which can be used to improve development activities, as they provide feedback on the quality of the software and the effectiveness of the development processes⁷. Defects are flaws or errors in the software that cause it to deviate from the expected or required results or behavior. Testing can help to detect, report, track, and resolve the defects, and prevent them from recurring in the future.

* D. Performing a review of the requirement specifications before implementing the system can enhance quality, as it can ensure that the requirements are clear, complete, consistent, testable, and aligned with the needs and expectations of the users and clients. Requirements are the specifications of what the software should do and how it should do it. Testing can help to validate that the requirements are met by the software, and verify that the software is implemented according to the requirements.

References =

* 1 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 10

* 2 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 11

* 3 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 12

- * 4 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 13
- * 5 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 97
- * 6 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 98
- * 7 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 14
- * [8] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 15
- * [9] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16
- * [10] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 17
- * [11] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 18
- * [12] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 19

NEW QUESTION: 68

□□ □ □□ □□ □□□□□?

- A. □□□□ □□□ □□□□ □□□ □□ □□□ □□□□ □ □ □□□□.
- B. □□□ □□□□ □□ □□□ □□ □□□□ □□□ □ □□□□.
- C. □□□□ □□□ □□□□ □□□□ □□□□□ □□□□ □□□□□.
- D. □□□□ □□ □□□□ □□□ □□ □□□□ □□□□ □□□□□.

Answer: ([SHOW ANSWER](#))

Looking for defects in a system does not require ignoring system details, but rather paying attention to them and understanding how they affect the system's quality, functionality, and usability. Ignoring system details could lead to missing important defects or testing irrelevant aspects of the system.

Identifying defects may be perceived as criticism against product, especially by the developers or stakeholders who are invested in the product's success. However, identifying defects is not meant to be a personal attack, but rather a constructive feedback that helps to improve the product and ensure its alignment with the requirements and expectations of the users and clients.

Looking for defects in system requires professional pessimism and curiosity, as testers need to anticipate and explore the possible ways that the system could fail, malfunction, or behave unexpectedly. Professional pessimism means being skeptical and critical of the system's quality and reliability, while curiosity means being eager and interested in finding out the root causes and consequences of the defects.

Testing is often seen as a destructive activity instead of constructive activity, as it involves finding and reporting the flaws and weaknesses of the system, rather than creating or enhancing it. However, testing is actually a constructive activity, as it contributes to the system's improvement, verification, validation, and optimization, and ultimately to the delivery of a high-quality product that meets the needs and expectations of the users and clients.

NEW QUESTION: 69

□□ □ □□□ □□□ □□ □□ □□□ □□□ □□□□□?

- A. □□□ □□□ □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□□ □□
- B. □□ □□□□ □□□□ □□□ □□□□ □□□□ □□□□ □□□ □□
- C. □□□ □□□□ □□□ □□ □□□ □□□□ □□□ □□□ □□ □□□□□ □□□ □□

D. □□□□ □□ □□□ □□ □□□ □□□□ □□□ □□ □□ □□□ □□

Answer: C ([LEAVE A REPLY](#))

Statement coverage is a white-box test technique which focuses on executing all possible statements in the code at least once during testing. This helps in determining the percentage of executable statements that have been executed by the test suite, aiming to ensure that all parts of the program have been tested at least once (ISTQB Main Web).

References:

ISTQB Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION: 70

□□ □□□□ □□ □□□□ □□□□ □□□ □ □□ □□(TAS)□ □□□ □□ □□ □□□□ □ □□□.

300,000€ □□□ □□ □□□□□ □□□□ □□ □□□□ □□□ □□□ □□ □□□ 100%□□ □. □ □□□□□□ "IF (TAS >= 300.000)" □□ "IF (TAS = 300,000)"□□□ □□□□ □ □□ □ □□ □□□□ □□□ □□□□. □ □□□ □□ 3□ □□□ □□□ □□□ □□ □ □□ □ □□ □□□□□(TAS□ □□ □□):

TC1 = 299,999 TC2 = 300,000 TC = 300,001

□□ □ □□ □□□ □□□□□?

A. TC1□ □□□ □□ □□□□□.

B. TC2□ □□□ □□ □□□□□.

C. TC3□ □□□ □□□□□.

D. □ □□ □□□ □□ □ □□ □□ □□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

The requirement specifies that a bonus should be paid if the total amount of sales (TAS) made during the year is 300,000€ or more. The software incorrectly implements this requirement with "IF (TAS = 300,000)" instead of "IF (TAS >= 300,000)". Using boundary value analysis (BVA), which is a common technique in software testing, the three test cases provided (TC1 = 299,999, TC2 = 300,000, and TC3 = 300,001) cover the critical boundary values around the condition.

* TC1 tests just below the boundary (299,999),

* TC2 tests exactly at the boundary (300,000),

* TC3 tests just above the boundary (300,001).

Since the software incorrectly checks for TAS equal to 300,000, only TC2 will fail because the condition is exactly met and highlights the incorrect implementation of the decision logic.

Reference: ISTQB CTFL Syllabus V4.0, Chapter 4.2.2, Boundary Value Analysis (BVA).

NEW QUESTION: 71

□□ □ □□ □□□ □□□□, □□□ □□□□, □□ □□□ □□□□ □□ □ □□ □□□ □□ □□□ □□□□ □□□ □□□□□?

A. □□□

B. □□□

C. □□ □□

D. ☐☐☐

Answer: D ([LEAVE A REPLY](#))

Comprehensive and Detailed In-Depth Explanation:

The Moderator (D) plays a key role in formal reviews by facilitating discussions, ensuring that reviews stay on track, and creating an open environment for feedback. The Manager (A) is responsible for overseeing the testing process but does not directly facilitate reviews. The Reviewer (B) is responsible for examining the work product, while the Review Leader (C) organizes reviews but does not necessarily mediate them.

Reference: ISTQB CTFL v4.0 Syllabus, Section 3.2.3 - Roles and Responsibilities in a Review

NEW QUESTION: 72

□□□ □□ □□□ □□□□ □□□□. "□□□ □□□□□ □□ □□□□ □□ □□□ □□□ □ □□□ 1□ □□□ □□ □□□ □□□□□ □□ □□□□." □□ □ □□ □□ □□ □□□ □□ □□□ □□□ □ □□□□?

i. □□□ □□ □□□ □□□□ □□□ □□ □□□ □□□ □, □□□ □□ □□□ □□□□□. □ □□ □□ □□□ 1□ □□□ □□ □□□□ □□□□□.

ii. □□□ □□ □□□ □□□□ □□ □□□ □□□ □ □□ □□□ □□□□□. □□□ □□ □□ □ 1□ □□□ □□ □□□□ □□□□□.

iii. □□□ □□ □□□□ □□□□ Enter□ □□□ □□□ □□ □□□ □□□ □, □□□ □□ □ □□ □□□□□. □□□ □□ □□□ 1□ □□□ □□ □□□□ □□□□□.

4. □□ □□□ □□□□, □□ □□□ □□□□, □□□ □□, □□ □□ □ □□ □□□ □□□□.

v. □ □□ □□□ □□□□, □□□ □□ □□□ □□□□, □□□ □□ □□□ □□□□ □□□ □ □□□ □□ □□□□ □□□□□.

□□□ □□ □□□□□:

A. i, ii ☐ v

B. 4

C. i ☐ iii

D. ☐

Answer: C ([LEAVE A REPLY](#))

Applicable acceptance test cases for the given user story should focus on the specific requirement of deleting an individual item from the shopping list and ensuring that it is removed in less than 1 second. Therefore, the valid test cases are: i. Click on my online shopping list, select the unwanted item, delete the unwanted item, the unwanted item is deleted from the shopping list in less than 1 second. iii. Tab to the online shopping list and press enter, select the unwanted item, delete the unwanted item, the unwanted item is deleted from the shopping list in less than 1 second.

Reference: ISTQB CTFL Syllabus V4.0, Section 5.2.2

NEW QUESTION: 73

□ □□□ □□□ □□□ □□□ □□ □□□□

□□:

Checklist-based testing is a technique where testers use pre-determined checklists to ensure that important aspects of a work product are evaluated. Over time, these checklists should be reviewed and updated periodically to maintain their effectiveness in detecting defects. As systems evolve, outdated checklists may miss new types of defects, thus diminishing their usefulness. Therefore, statement D is true according to the ISTQB CTFL syllabus.

NEW QUESTION: 75

□□□□□□ □□ □□□ □□ □□□ □□□□□?

- A. □□ □□□ □ □□ □□□ □□□ □□ □□□□□□ □□ □□□ □□ □□□□□□. B. □□□ □□□ □□ □□ □□ □ □□ □□ □□□□ □□□ □□ □□□ □ □□□□.
- B. □□ □□□□□ □□□ □□□□, □□ □□□□, □□ □□□ □□ □□□ □□□□□.
- C. □□ □□□□□ □□□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□.

Answer: C ([LEAVE A REPLY](#))

□□:

□□□□□□ □□ □□□ □□□ □□ □□ □□□ □□ □□□ □□□ □□□□ □□ □□ □ □□□ □□□□□. □□ □□□ □□ □□□ □□□ □□□, □□□ □□□□, □□□ □□, □□□ □□ □ □□ □□□□□ □□□ □□□□□ □□□□, □□ □□□ □□□□, □□ □□ □□ □□□ □□□ □ □□□ □□ □□□□.

□□ C□ □□□ □□□ □□□□ □□□□□. □□□□□□ □□ □□ □□□ □□□□□□ □□ □□□ □□□ □□ □□□ □□□□□ □□□□ □□□□ □□□□ □□ □□□ □ □□□□□ □□□ □□□□□ □ □ □□□□. □□□ □□ □□□□□ □□ □□ □□□□ □□□, □□□ □□□ □□□□ □□□, □□ □□□□ □□□□ □□□□ □□□ □□.

□□ A, B, D□ □□□ □□ □ □□□ □□□□□ □□ □□□ □□□□□ □□□□□ □□□ □ □□□ □□ □ □□□ □□□□ □□ □□□□□ □□ □□□ □□ □□□ □□□□ □□□□.

NEW QUESTION: 76

□□□ □□□ □□ □□□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□□. □□ □□□ □□ □□□□ □□□ □□□□.

□□ \$100.00 = 2%

\$100.01 ~ \$500.00 = 4%

\$500.01 ~ \$1,000.00 = 5%

\$1,000.00 □□ = 7%

2□ □□□ □□□ □□□ □, □□ □□□ □□ □□ □ □□ □□ □□□□□ □□ □□ □□□ □□ □□ □□□□□?

- A. \$5.00.\$100.00,\$499.99,\$1,000.00,\$1,000.01
- B. \$100.00. \$100.01, \$100.02, \$500.00, \$999.99
- C. \$100.00, \$500.00, \$1,000.00, \$1,000.01
- D. \$5.00.\$100.00,\$500.00,\$1,000.01

Answer: ([SHOW ANSWER](#))

Boundary Value Analysis (BVA) is a software testing technique in which tests are designed to include values at the boundaries. The concept is to focus on the boundaries since errors tend to occur at the edges of input ranges rather than in the middle.

Given the problem statement:

* Up to \$100.00 = 2%

* \$100.01 to \$500.00 = 4%

* \$500.01 to \$1,000.00 = 5%

D. '□□ □□'□ □□ □□ □□□ □□ □□□ 10□ □□□□□□, □□ □□ □□□ 13□□□□ □.

Answer: B (LEAVE A REPLY)

An incident report during testing should focus on factual observations of failures or defects in the system, including their impacts and how they deviate from expected results. Options A, C, and D describe specific issues that are directly related to the system's behavior or performance and are suitable for inclusion in an incident report. Option B, which describes the user interface as "complicated and confusing" and relates to the tester's personal difficulty in following the test script, is more subjective and relates to the tester's experience rather than an objective observation of a system failure. Therefore, option B is the least appropriate for an incident report.

NEW QUESTION: 78

□□ □ □□□□ □□□□ □□□ □□ □ □□□□ □□ □□□□□?

- A. □□□□ □□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□□ □□ □□□□ □□□ □□□ □□ □□□ □□□□ □□□ □□ □□□□.
- B. □□ □□ □□□ □□□□ □□ □□□□ □□□ □□ □□□□ □□□□ □□ □□□ □□□ □□□□.
- C. □□□□ □□□ □□ □□□ □□ □ □□□□ □□ □□□ □ □□□□.
- D. □□□□ □□□ □□ □□□□□ □□□ □□ □□ □□□ □□□ □□ □□□□ □□□ □□ □□□ □□ □□□ □□ □□□□ □□□□ □□□□ □ □□□□.

Answer: C (LEAVE A REPLY)

Independent testing offers several advantages, such as unbiased testing and detection of different defects.

However, a drawback is that an independent test team may be isolated from the development team and project team. This can lead to communication gaps, reduced collaboration, and a lack of understanding of the project context.

According to the ISTQB Certified Tester Foundation Level (CTFL) syllabus v4.0, an independent test team may not have the same level of understanding of the system as the development team, leading to potential issues in communication and integration (ISTQB not-for-profit association).

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 79

□□□ □□□ □□ □□□ □□□□□ □□ □□□ □□□□ □□□. □□ □□□ □□□ □□□ □□□ □□□ □□ □□□□. □□(□□ □)□ □□□ □□(□□□ □)□ □□□□□.

TopicDescription

A. □□□□ □□ □□□ □□□□ □□.

8: □□ □□B: □□ □□□ □□□□ □□□ □□□□ □□ □□ □□□□ □□□ □□ □□□ □□□□.

B. □□□□ C: □□ □□□□ □□□ □□□ □□□□ □□ □□□□ □□ □□□ □□□ □□□ □□□.

C. □□□□ □□D; □□□ □□□□ □□ □□ □□ □□□ □□ □□□□□.

D. □□ □□□□□□E: □□ □□ □ □□□ □□ □□□ □□ □□ □□□□ □□□□□ □□□ □□ □□□□□.

E. $A \square \square C \square$, $B \square \square A \square$, $C \square \square B \square$, $D \square \square E \square$, $E \square \square D \square$

F. $A \subseteq E$, $B \subseteq A$, $C \subseteq B$, $D \subseteq C$, $E \subseteq D$

G. A□□ C□, B□□ A□, C□□ B□, D□□ D□, E□□ E□

H. $A \square \square A \square$, $B \square \square B \square$, $CTOB$, $D \square \square E \square$, $E \square \square D \square$

Answer: ([SHOW ANSWER](#))

The correct matching of the topics with their descriptions is as follows:

- * Functional behaviour: The externally observable behaviour with user actions as input operating under certain configurations (A to C).
- * Quality characteristics: How the system performs the specific behaviour (B to A).
- * Scenarios: A sequence of actions between an external actor and the system, to accomplish a specific goal or business task (C to B).
- * Business rules: Activities that can only be performed in the system under certain conditions defined by outside procedures and constraints (D to E).
- * External interfaces: Description of the connections between the system to be developed and the outside world (E to D).

- * Quality characteristics: How the system performs the specific behaviour (B to A).

- * Scenarios: A sequence of actions between an external actor and the system, to accomplish a specific goal or business task (C to B).

- * Business rules: Activities that can only be performed in the system under certain conditions defined by outside procedures and constraints (D to E).

- * External interfaces: Description of the connections between the system to be developed and the outside world (E to D).

NEW QUESTION: 80

☐☒ ☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐☐☐ ☐☐ ☐☐☐☐☐

A. □□□□ □□□ □□□ □□□ □ □□□□ □□□□ □□□ □□□ □ □□□□.

B. □□□ □□ □□□ □□□ □ □ □□ □□□ □□ □□□ □□□□ □□ □ □□□□.

C. □□ □□□ □□□□ □□□ □□ □□□□□.

D. □□ □□ □□□ □□ □□ □ □□□□□.

Answer: (SHOW ANSWER)

- * Effort estimate does not depend on the budget of the project, but rather on the scope, complexity, and quality of the software product and the testing activities¹. Budget is a constraint that may affect the feasibility and accuracy of the effort estimate, but it is not a factor that determines the effort estimate. Effort estimate is the amount of work required to complete the testing activities, measured in terms of person-hours, person-days, or person-months².
- * The other options are correct because:
 - * A. Once the test effort is estimated, resources can be identified and a schedule can be drawn up, as they are interrelated aspects of the test planning process³. Resources are the people, tools, equipment, and facilities needed to perform the testing activities⁴. Schedule is the time frame and sequence of the testing activities, aligned with the project milestones and deadlines⁵.
 - * B. Effort estimate can be inaccurate because the quality of the product under tests is not known, as it affects the number and severity of the defects that may be found and the rework that may be needed to fix them⁶. Quality is the degree to which the software product satisfies the specified requirements and meets the needs and expectations of the users and clients⁷.

- * The other options are correct because:

- * A. Once the test effort is estimated, resources can be identified and a schedule can be drawn up, as they are interrelated aspects of the test planning process³. Resources are the people, tools, equipment, and facilities needed to perform the testing activities⁴. Schedule is the time frame and sequence of the testing activities, aligned with the project milestones and deadlines⁵.

- * B. Effort estimate can be inaccurate because the quality of the product under tests is not known, as it affects the number and severity of the defects that may be found and the rework that may be needed to fix them⁶. Quality is the degree to which the software product satisfies the specified requirements and meets the needs and expectations of the users and clients⁷.

* D. Experience based estimation is one of the estimation techniques, which relies on the judgment and expertise of the testers and other project stakeholders to estimate the test effort based on similar projects or tasks done in the past. Experience based estimation can be useful when there is a lack of historical data, formal methods, or detailed information about the software product and the testing activities.

References =

- * 1 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 154
- * 2 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 155
- * 3 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 156
- * 4 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 157
- * 5 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 158
- * 6 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 159
- * 7 ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16
- * [8] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 160
- * [9] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 161

NEW QUESTION: 81

□□ □ □□ □□□ □□ □□ □□□□?

- A. □□□□ □□□ □□□ □□ □□ □□□□ □□□□.
- B. □□□ □□□ □□□□□ □□□ □□□ □□□ □ □□ □□□□ □□□.
- C. □□□□ □□□□ □□□□ □□□□ □□ □□□□.
- D. □□□□ □□□ □□□□□□ □□ □□□□□ □□□ □□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

Statement B contradicts the general principles of testing, because running the same test set more often will not increase the chances of finding new defects, unless there are some changes in the system or environment that affect the test results. Running different test sets with different inputs, outputs or conditions would be more effective in finding new defects. Statements A, C and D are consistent with the general principles of testing.

Statement A states that most defects are found in a small subset of a system's modules, which is true according to the defect clustering principle. Statement C states that testing is better if it starts at the beginning of a project, which is true according to the early testing principle. Statement D states that how testing is done, is based on the situation in a particular project, which is true according to the context-dependent testing principle. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, pages 4-6.

NEW QUESTION: 82

□□□□ □□ □□ □□□ □□□□□. □□□ □□□ □□ □□□□□.

- □□ □□(□□, □□□, □□, □□, □□, □□)
- □□ □□ (□□ □□ □□)
- □□ (□ 1. 2. 3)
- □□□ □□ (□□, □□)

□□ □ □□ □□□ □□□□□ □ □□ □□□ □□□ □□□ □□□□□?

- A. □□ □□
- B. □□ □□ □□□
- C. □□ □□
- D. □□ □□ □□□

Answer: ([SHOW ANSWER](#))

Equivalence partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component. Equivalence partitioning can be used to reduce the number of test cases by selecting one representative value from each partition. Equivalence partitioning is suitable for testing the price computation, as it can identify different partitions based on the passenger type, the travelling type, the distance and the kind of transport. Equivalence partitioning is not statement coverage, which is a technique that measures how many executable statements in a source code are executed by a test suite. Statement coverage is not appropriate for testing the price computation, as it does not consider the input data or output results.

Equivalence partitioning is not state transition testing, which is a technique that models how a system transitions from one state to another depending on events or conditions. State transition testing is not relevant for testing the price computation, as it does not involve any states or transitions. Equivalence partitioning is not use case testing, which is a technique that tests how users interact with a system to achieve a specific goal.

Use case testing is not applicable for testing the price computation, as it does not focus on a single function or component. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION: 83

□□ □□ □ □□□□ □□□ □□□ □□ □ □□□□ □□ □□□□□?

- A. □□□□ □□□ □□□ □□ □□□ □□ □□ □ □□□ □□□ □□ □□□ □□□□ □□□.
- B. □□□□ □□□ □□□□□ □□□□□ □□□ □□□□ □□□ □□□□ □□□□.
- C. □□□□ □□□ □□□ □□□ □□□ □□ □□□ □□□ □□ □□ □ □□□ □□□ □□□ □□□ □□□ □□□ □□□□ □ □□□□□.
- D. □□□□ □□□ □□□ □□□ □□□ □□ □□□ □□ □□ □ □□ □□□ □□□ □□□ □ □□□ □□ □□□□□.

Answer: D ([LEAVE A REPLY](#))

Comprehensive and Detailed In-Depth Explanation:

Black-box testing techniques focus on testing the functionality of the software without knowledge of its internal workings (D). They derive test cases from specifications, requirements, or expected behavior.

Option A describes exploratory testing, B contradicts the definition by focusing on implementation, and C incorrectly includes "implied behavior," which is not a core characteristic of black-box testing.

Reference: ISTQB CTFL v4.0 Syllabus, Section 4.2 - Black-Box Testing Techniques

NEW QUESTION: 84

□□ □□□ □□□□ 1 "□□□□ □□□□"□ □□□□□.

□□□ □□□ TC_59□ □□□ □, □□ □□ fort = 35degrees Celsius□ □□ □□□ □□

OUTPUT = X□□□.

□□ □□□□ □□□□□□ □ □□ □□□ □□□□□?

A. □□□ □□ □ □□□ □□ □□

B. □□□ □□□ □□ □□□ □□□ □□□ □□

C. □□□□□ □□□ □□ □□

D. □□ □□□ □□ □□

Answer: C ([LEAVE A REPLY](#))

Comprehensive and Detailed In-Depth Explanation:

A defect report should contain relevant details to help developers reproduce and fix the defect efficiently.

The essential elements include:

- * Test object & environment (A) - To ensure reproducibility.
- * Title & summary (B) - For quick identification.
- * Expected vs. actual results (D) - To describe the discrepancy.

The structure of the test team (C) is irrelevant for defect tracking and resolution.

Reference: ISTQB CTFL v4.0 Syllabus, Section 5.5 - Defect Management

NEW QUESTION: 85

□□ □□□ □□□ □□□ □□□.

1) □□ □□ □□□

2) □□ □□□

3) □□□ □□□

4) □□ □□□

□□ □ □□ □□□ □□□□□?

A. V-□□□ □□□□ □□ □ □□□ □□□□ □□□□□.

□□□ □□ □□□ □□□□ □□ □□ □□□□ □□ □□□□ □□□ □ □□□□.

B. □□ □□□ □□□ □□□□□ □□□ □□ V-□□□ □□ □□□□□.

□□ □□□□ □□ □□□□ □□□ □ □□□□.

C. □□ □□□□ □□ □□□□□ □□ □□□ □□□ □ □□□□.

□□ □□ □ □□□ □□□□ V□□□□ □□□□□.

D. □□□□□ □□ □□ □□ □□□□(V□□, □□□, □□□) □ □□ □□ □□□□ □□ □□ □ □□□ □□ □□□□□.

Answer: D ([LEAVE A REPLY](#))

All testing levels are applicable, independent of which software development life-cycle process

(V-model, iterative, incremental) is used. Testing levels are defined based on the scope and

objectives of testing, not on the software development model. Component testing, integration

testing, system testing and acceptance testing are common testing levels that can be applied to

any software development model, as long as they are planned and executed properly. The V-model is a software development model that emphasizes the relationship between each development phase and its corresponding testing phase. Iterative and incremental models are software development models that divide the development process into smaller cycles or iterations, where each iteration produces a working version of the software that can be tested and evaluated.

Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 18.

NEW QUESTION: 86

□□□□ □□□ □□□ □□□□ □□□ □□ □□ □□ □ □□□ □□ □□□□□?

- A. □□□□ □□ □□□ □□□□□.
- B. □□□□ □□ □□□ □□ □□□ □□□ □□□□□.
- C. □□□□ □□□ □□□□ □ □□ □□□ □□□□ □□□□□.
- D. □□□□ □□□ □□□□ □□□ □□□□ □□□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 87

□□ □ □□ □□□ □□□□?

- A. □□□□ □□□ □□ □□□ □□ □□□ □□□□ □□□□□ □□ □□ □□□□.
- B. □□□□□ □□ □□□ □□ □□□□□□ □□ □□ □□□□ □□□□ □□ □□ □□□ □□ □□ □□□□.
- C. □□□ □□□□ □□ □□□ □□□ □□□□ □□□□ □ □□□ □□□ □□□□ □ □□□ □.
- D. □□ □□□ □□□ □□□□□ □□□□ □□ □□□ □□ □□□ □ □□□□.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 88

□□ □□ □□□□ □□□□□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□□ □□□ □□□□(□□□ □□ □□□ □□□□ □□□ □)□ □□□□□ □□ □□ □□□□ □□□ □□ □□□□□.

- A. □□□ □□□□ □ □□
- B. □□□ □□
- C. □□□ □□
- D. □□□ □□

Answer: ([SHOW ANSWER](#))

This answer is correct because test implementation is the activity where test work products, such as test cases, test data, test scripts, test harnesses, test stubs, or virtual services, are created and verified. Test implementation also involves setting up the test environment and preparing the test execution schedule. A virtual service emulating a real third-party service and the automated test scripts that interact with that service are examples of test work products that are typically

created during test implementation. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.3

NEW QUESTION: 89

□□ □□ □□ 4□□ □□ □□□□ □□□□□.

$0 < x < 100$

$100 \leq x \leq 200$

$200 < x < 500$

$x \geq 500$

□□ □□ □ □□□ □□ □□□ □□□ □□ □□□ □□□ □□□ □□□ □□ □□□□□?

A. 50; 100; 200. 1000

B. 0. 1.99, 100.200,201.499, 500;

C. 0.50; 100; 150.200.350.500;

D. 50; 100; 250; 1000

Answer: (SHOW ANSWER)

The correct set of data for valid equivalence class partitions should include one value from each equivalence class, and no value from outside the range. Option C satisfies this condition, as it has one value from each of the four equivalence classes (50, 100, 250, 500). Option A has two values from the same equivalence class (100 and 200), option B has values outside the range (0 and 0.99), and option D has two values from the same equivalence class (1000 and 500). Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 35.

NEW QUESTION: 90

□□ □□□ □□ □□□□□□ □□ □□ □□□ □□□□□□□. □□□□□ □□ □ □□ □□ □□ □□□□ □□□ □□□□□. □□ □□□□ □□ □ □□ □□□□ □□□ □□ □□□□□?

A. □□□□□□ □□□ □□ □□□□ □□ □□□

B. □□ □□□ □□□ □□□ □□□

C. □□□□ □□□□ □□□ □□□□ □□ □□□□□ □□□

D. □□□ □□ □□□ □□ □□□ □□□□□ □□□ □□□□ □□ □□□

Answer: B (LEAVE A REPLY)

Confirmation testing, also known as re-testing, is performed to verify that specific defects have been successfully fixed.

Option A: "Testing due to the application of a new version of the interface" would typically involve regression testing, not confirmation testing.

Option B: "Testing that fixes resolved the defects in the search function" fits the description of confirmation testing as it focuses on ensuring that specific issues have been addressed.

Option C: "Testing if a system still works after update of an operating system" is an example of regression testing, as it checks the overall system behavior after an update.

Option D: "Testing to ensure the adding of new functionalities haven't broken existing functions" is another example of regression testing (ISTQB not-for-profit association) (Udemy) it checks for unintended consequences of new changes.

Therefore, the correct answer is B.

References:

Certified Tester Foundation Level v4.0

ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 91

Which of the following is a failure?

- A. A defect in the software system that causes a failure.
- B. A defect in the software system that causes a failure, but only if the user uses the software system incorrectly.
- C. A defect in the software system that causes a failure, but only if the user uses the software system in a way that is not specified in the requirements.
- D. A defect in the software system that causes a failure, but only if the user uses the software system in a way that is not specified in the requirements and the user uses the software system incorrectly.

Answer: (SHOW ANSWER)

Failures can be caused by defects, but also by environmental conditions. A failure is an event in which the software system does not perform a required function or performs a function incorrectly, according to the expected behavior. A defect is a flaw in the software system or a deviation from the requirements or the specifications, that may cause a failure. However, not all failures are caused by defects, as some failures may be caused by environmental conditions, such as hardware malfunctions, network interruptions, power outages, incompatible configurations, etc. Environmental conditions are factors that affect the operation of the software system, but are not part of the software system itself. The other statements are false, because:

* A defect does not always produce a failure, while a bug always produces a failure. This statement is false, because a defect may or may not produce a failure, depending on the inputs, the outputs, the states, or the scenarios of the software system, and a bug is just another term for a defect, so it has the same possibility of producing a failure as a defect. For example, a defect in a rarely used feature or a hidden branch of the code may never produce a failure, while a defect in a frequently used feature or a critical path of the code may produce a failure often. A bug is not a different concept from a defect, but rather a synonym or a colloquial term for a defect, so it has the same definition and implications as a defect.

* A defect may cause a failure which, when occurring, always causes an error. This statement is false, because an error is not a consequence of a failure, but rather a cause of a defect. An error is a human action or a mistake that produces a defect in the software system, such as a typo, a logic flaw, a requirement misunderstanding, etc. An error is not observable in the software system, but rather in the human mind or the human work products, such as the code, the design, the documentation, etc. A failure is not a cause of an error, but rather a result of a defect, which is a result of an error. For example, an error in the code may cause a defect in the software system, which may cause a failure in the software behavior.

by the development process. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 2, page 18-19.

NEW QUESTION: 93

□□□□□ □□□ □□□ □□□ □□□□□ □□□□□□. □□□ □□ □□ □□□ □
□□□□□ □□□□□ □□□□ □□ □ □□□ □□ □□□□ □□□□ □□□□□ □□□ □□
□□.

A. □□ □□□ □□□□□□□. □□ □□□ □□ □□□□ □□□ □□□ □□□ □□□□ □□
□.

B. □□ □□□ □□□□. □□ □□□ CMM □□□ □□ □□ □□□ □□□□ □□□.

C. □□ □□□ □□□□□□□. □□□ □□□□□ □□ □□ □□□ □□□□ □□□ □ □□□
□.

D. □□ □□□ □□□□. □□□□ □□ □□□ □□□□ □□□ □□□□□ □□□□ □ □□□
□□□□□.

Answer: C ([LEAVE A REPLY](#))

The thought process of the software company is incorrect, because it assumes that each artifact can be reviewed using only one review method, and that the review method depends solely on the criticality of the artifact. This is a simplistic and rigid approach that does not consider the benefits and limitations of different review methods, the context and purpose of the review, and the feedback and improvement opportunities that can be gained from multiple reviews. According to the CTFL 4.0 Syllabus, the selection of review methods should be based on several factors, such as the type and level of detail of the artifact, the availability and competence of the reviewers, the time and budget constraints, the expected defects and risks, and the desired outcomes and quality criteria. Moreover, the same artifact can be reviewed using different review methods at different stages of the development lifecycle, to ensure that the artifact meets the changing requirements, standards, and expectations of the stakeholders. For example, a requirement specification can be reviewed using an informal review method, such as a walkthrough, to get an initial feedback from the users and developers, and then using a formal review method, such as an inspection, to verify the completeness, correctness, and consistency of the specification. Therefore, the software company should adopt a more flexible and context-sensitive approach to selecting and applying review methods for different artifacts, rather than following a fixed and arbitrary rule. References = CTFL 4.0 Syllabus, Section 3.2.1, page 31-32; Section

3.2.2, page 33-34; Section 3.2.3, page 35-36.

NEW QUESTION: 94

□□ □ □□□□ □□□ □□ □□ □□□□□?

A. □□ □□

B. □□□□□ □□ □□ □□

C. □□□□□ □□ □□□ □□ □□□ □□

D. □□ □□ □□ □ □□

Answer: D (LEAVE A REPLY)

Analyzing and removing the cause of failures is not an objective of testing, but rather a task of development or maintenance. A failure is an event or behavior that deviates from the expected or specified result of a system under test. A failure is caused by an error (also known as a mistake or a fault) in the software code, design, or specification. Analyzing and removing the cause of failures is a process of locating and fixing errors in the software code, design, or specification, which is also known as debugging or defect resolution.

Analyzing and removing the cause of failures does not aim to find or report defects, but rather to correct or prevent them. The other options are objectives of testing. Finding defects is one of the main objectives of testing, as it helps to improve the quality and reliability of the software product. Providing information for decision-making is another objective of testing, as it helps to support decision making and risk management.

Gaining confidence about the level of quality of the software is another objective of testing, as it helps to assure that the software product meets its requirements and customer or user needs and expectations. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 3.

NEW QUESTION: 95

□□□ □□□ □□□□ □□ □ □□□□□?

A. □□□□□ □□□□□ □□□ □ □□ □□□ □□□□ □□ □□□□. □□□ □□□ □□□ □□ □□□ □ □□□ □□□.

B. □□□□□ □□□□□□□ □□□□□ □□□ □□□ □□□ □□□ □□□ □□□ □□□ □□□ □□□ □.

C. □□□ □□□□ □□□ □□□ □□□ □□□□ □□□ □□ □□□□ □ □□ □□□□ □□ □ □□□ □□□□□ □□□ □□□□□.

D. □□□ □□□ □□ □□□ □□□□, □□ □□□□ □□□ □□□ □□□□.

Answer: C (LEAVE A REPLY)

Selecting the right test technique is crucial because different techniques are suited to different types of testing and can significantly increase the effectiveness of the testing process by creating tests that are more likely to find defects. While reducing the number of tests (A) and defining the number of regression cycles (D) are considerations in the testing process, they are not the primary reasons for selecting a test technique. The assertion that the only way to test a software application is by using well-proven test techniques (B) is too restrictive and does not acknowledge the adaptability required in testing to suit different contexts and objectives. Therefore, option C is the most comprehensive reason, as it focuses on the effectiveness and efficiency of testing, leading to the creation of high-quality tests that have a higher chance of finding bugs.

NEW QUESTION: 96

□□□ □□ □□□ □□ □□□ □□□ □□:

a. □□ □□

b. □□□□

c. □□ □□

d. □□

1. □□□

2. □□□

3. □□□□ □□ □□ □□ □□□ □□□ □□□□ □□□ □□□□(□: □□□□□□ □□□)□
□□□□ □□ □□□□□.

4. □□□□ □□□□□ □□ □□ □ □□ □□□ □□□ □□□□□. □□ □ □□ □□□ □□
□□□ □□ □ □□□□ □□ □□□□□?

A. a-1. b-4, c-3. d-2

B. a-4, b-3. c-2. d-1

C. a-2, b-3, c-4, d-1

D. a-2, b-4, c-3. d-1

Answer: C ([LEAVE A REPLY](#))

Pair reviews are informal and typically involve two people reviewing the work product together, often in an informal setting. Walkthroughs are more formal and aim to educate stakeholders and evaluate the product, serving the dual purpose of improving the product and training participants. Technical reviews have a strong focus on improving the product's quality, often involving technical stakeholders. Inspections are the most formal review type and are aimed primarily at detecting defects.

References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2.3 "Review Types".

NEW QUESTION: 97

□□ □ □□□ □□ □□□ □□ □□□ □□ □□□□□?

A. □□□□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□□ □□□□□□□.

B. □□□□ □□□□□ □□□ □□□ □□□□ □□□ □□□ □□□□□.

C. □□ □□□□ □□ □□□□ □□□□ □□ □□□ □□□ □□□ □□□ □□□□□.

D. □□□ □□ □ □□□ □□□ □□ □□

Answer: D ([LEAVE A REPLY](#))

Test design is one of the fundamental activities in software testing, focusing on how tests will be structured and what strategies will be employed. As per the ISTQB syllabus, identifying appropriate test execution and automation tools is a task closely associated with test design activities because it directly influences how tests will be conducted, managed, and executed efficiently. This activity involves deciding on the tools that will best support the testing goals and requirements.

References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4 "Test Design Techniques".

NEW QUESTION: 98

□□ □ □□□□ □□□□□ □□□ □□□ □□□□ □□□□ □□ □□□□□?

A. □□□ - □□ □□ □□ □□□ □□□ □□□□□.

- B. □□ - □□ □□ □□ □ □□□ □□□ □□□ □□□□□.
- C. □□ □□ - □□ □□ □□ □□□ □□□□□.
- D. □□□ - □□ □□ □□ □□□ □□□ □□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

In formal reviews, the scribe's role is to collate potential defects and other findings during the review process.

This position is crucial as it ensures all observations and defects are recorded accurately, facilitating efficient analysis and resolution of issues identified during the review.

References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2.4 "Roles and Responsibilities in a Formal Review".

NEW QUESTION: 99

□□ □□ □□□ □□□□□.

□□□□ ID: 2 8

□□ □□ □□ □□□

□□□ □□

□□□□ S3□ □□□□□ □□□□□,

1) 7□ □□ □□□□ □□□□□.

2) 7□□□ 13□□□□ □□□□ □□ □□□□ 20% □□□ □□□□.

3) 65□ □□ □□□□ □□ □□□□ 50% □□□ □□□□.

□□□ 0 □□□ □□□□ □□□.

□□□ □□□□□ □□□□□□ □□□□□.

□□ □ □□ □□□ □□ □□□□?

A. □□□□ □□□ □□ □□ □□□□.

B. □□ □□□ □□ □□□ □□□ □□ □□ 6□□ □□□ □□□ □□□□□ □□□□□.

C. \$3.01□ □□□ □□ □□ □□□□.

D. 7□ 13□ 10□□ □□□ □□ □□□ □□□□□□.

Answer: A ([LEAVE A REPLY](#))

Boundary value analysis is a technique that tests boundary values between partitions of equivalent data.

Boundary values are values at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Boundary value analysis can be applied to both input and output values. Based on the given requirement, we can identify two input values: age and weekend. Age should be an integer of 0 or above, and weekend means Friday to Sunday inclusive. The following statement is not correct:

* A) Thursday is a valid input boundary value. This statement is not correct, as Thursday is not a boundary value for the input weekend. The boundary values for the input weekend are Friday and Sunday, as they are at the edge of the equivalence partition that represents weekend days. The following statements are correct:

* B) A minimum of 6 valid test cases are derived from boundary value analysis based on input age. This statement is correct, as we can derive six valid test cases based on input age by using

* C) \$3.01 is a valid output boundary value. This statement is correct, as \$3.01 is a boundary value for the output additional fee. The additional fee can have four possible values depending on the input age: \$0 (for visitors aged under 7), \$2.40 (for visitors aged 7 to 13 inclusive with a 20% discount), \$1.50 (for visitors aged greater than 65 with a 50% discount), and \$3 (for visitors aged between 14 and 65). The boundary values for the output additional fee are \$0 and \$3, as they are at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Therefore, \$3.01 is a valid output boundary value, as it is at the smallest incremental distance above \$3.

NEW QUESTION: 100

1. ☐ ☐ ☐

III □□□ □□□ □□□□ □□□

IV API □□□□ □□□□□ □□□ □□

A. I, III

B. I. II

C. II. IV

D. III. IV

Answer: ([SHOW ANSWER](#))

System testing is a level of testing performed to evaluate the behavior and quality of a whole software product or system. System testing can include various types of testing, such as:

* I) Usability testing: A type of testing that evaluates how easy, efficient and satisfying it is to use the software product or system from the user's perspective.

* II) Requirements based scenarios testing: A type of testing that verifies that the software product or system meets its specified requirements or user stories by executing realistic scenarios or workflows.

System testing does not include the following types of testing, as they are more suitable for lower levels of testing, such as unit testing or integration testing:

* III) Testing parts of the code in isolation: A type of testing that verifies the functionality and quality of individual software components or units by isolating them from other components or units.

* IV) Correct order of parameters in API calls: A type of testing that verifies the functionality and quality of software components or units that communicate with each other through application programming interfaces (APIs) by checking the correct order and format of parameters in API calls. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 2, page 20-21; Chapter 4, page 34-35.

NEW QUESTION: 101

□□□ □□□□□ □□□□□ □□ □□□ □□□□□□. □□□□ □□□ □□□ □ □□□ □
□□□□ □□□□ □□ □□□ □□ □□□□□ □□□□□□. □□ □ □□ □□ □□□□?

- A. □□□□□ □□ □□□□ □□□□ □□□ □□□□□□ □□□□ □□ □□□□□ □ □□
□.
- B. □□□ □□□□□ □□□□ □□ □□□□□ □ □□, □□□ □□□□ □□□□ □□□.
- C. □□□ □□□□□ □□□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□ □□□□ □
□ □□□□ □□□.
- D. □□□ □□□□□ □□□□ □□ □□□□ □□□. □□□□ □□ □□□ □□□□ □□□□
□□ □□ □□□□□.

Answer: C ([LEAVE A REPLY](#))

The test engineer should re-run the tests, in order to ensure that new defects have not been introduced by the fix. This is also known as regression testing, which is a type of testing that verifies that previously tested software still performs correctly after a change. Regression testing helps to detect any side effects or unintended consequences of a fix or a modification. The other options are incorrect reasons for re-running the tests. The test engineer should not re-run the tests, as they have already been run, and results recorded, because this ignores the possibility of new defects caused by the fix. The test engineer should not re-run the tests, they should be part of the developer tests, because this assumes that developer tests are sufficient and reliable, which may not be true. The test engineer should not re-run the tests, because the defect shows that the test cases need to be updated, because this does not address the impact of the fix on other test cases or functionalities. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 41.

NEW QUESTION: 102

□□ □□□ □□□□ □□□□□ □□□ □□ □□ □□□□ □□□ □□□□□?

- A. □□ □□□
- B. □□ □□□
- C. □□ □□□
- D. □□ □□□

Answer: ([SHOW ANSWER](#)**)**

Structural testing is a type of testing that measures its effectiveness by tracking which lines of code were executed by the tests. Structural testing, also known as white-box testing or glass-box testing, is based on the internal structure, design, or implementation of the software. Structural testing aims to verify that the software meets the specified quality attributes, such as performance, security, reliability, or maintainability, by exercising the code paths, branches, statements, conditions, or data flows. Structural testing uses various coverage metrics, such as function coverage, line coverage, branch coverage, or statement coverage, to determine how much of the code has been tested and to identify any untested or unreachable parts of the code. Structural testing can be applied at any level of testing, such as unit testing, integration testing, system testing, or acceptance testing, but it is more commonly used at lower levels, where the testers have access to the source code.

The other options are not correct because they are not types of testing that measure their effectiveness by tracking which lines of code were executed by the tests. Acceptance testing is a type of testing that verifies that the software meets the acceptance criteria and the user requirements. Acceptance testing is usually performed by the end-users or customers, who may not have access to the source code or the technical details of the software. Acceptance testing is more concerned with the functionality, usability, or suitability of the software, rather than its internal structure or implementation. Integration testing is a type of testing that verifies that the software components or subsystems work together as expected. Integration testing is usually performed by the developers or testers, who may use both structural and functional testing techniques to check the interfaces, interactions, or dependencies between the components or subsystems. Integration testing is more concerned with the integration logic, data flow, or communication of the software, rather than its individual lines of code. Exploratory testing is a type of testing that involves simultaneous learning, test design, and test execution. Exploratory testing is usually performed by the testers, who use their creativity, intuition, or experience to explore the software and discover any defects, risks, or opportunities for improvement. Exploratory testing is more concerned with the behavior, quality, or value of the software, rather than its internal structure or implementation. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54; Chapter 1: Fundamentals of Testing, Section 1.4: Testing Throughout the Software Development Lifecycle, Pages 11-13; Chapter 3: Static Testing, Section 3.4: Exploratory Testing, Pages 40-41.

NEW QUESTION: 103

□□ □ □□□□□ □□□ □□ □□□□ □□□□ □□ □□□ □□□□□?

- A. □□□□ □ □□ □□
- B. □□ □□□ □□ □□□ □□
- C. □□□ □□□□□ □□ □□□ □□□□ □□□ □□□
- D. □□□ □□ □□ □□□ □□

Answer: B ([LEAVE A REPLY](#))

Comprehensive and Detailed In-Depth Explanation:

A test progress report provides an overview of testing activities, metrics, and identified risks. It focuses on tracking testing progress rather than evaluating overall product quality (B), which is typically included in a test summary report after testing is completed.

* (A) is correct because obstacles (challenges) are reported to ensure test execution stays on track.

* (C) is correct as test metrics help stakeholders track execution progress.

* (D) is correct because new or changed risks impact test focus and priorities.

A test progress report tracks execution and informs stakeholders about ongoing testing activities.

Reference: ISTQB CTFL v4.0 Syllabus, Section 5.3 - Test Monitoring and Control

NEW QUESTION: 104

□□ □□ □ □□□ □□□□ □□ □□□ □□ □ □□□□ □□ □□□□□?

A. □□□ □□□ □□□□ □□□ □□□□ □□□□□.

B. □□ □□□ □□□□□ □□□ □□□□ □□□□□.

C. □□□ □□□□ □□ □□□ □ □□ □□□□□ □□□□□.

D. □□□□ □□□□ □□□ □□□ □□□□□ □□□ □□□□.

Answer: ([SHOW ANSWER](#))

Comprehensive and Detailed In-Depth Explanation:

Non-functional testing evaluates attributes like performance, security, and usability, which are typically assessed at the system test level or higher (A). It is not restricted to acceptance testing (B), goes beyond security and performance testing (C), and covers more than just user experience (D).

Reference: ISTQB CTFL v4.0 Syllabus, Section 2.2.2 - Non-functional Testing

NEW QUESTION: 105

□□□ □□□ □□□□□:

* □□□ □□ □□□ □□□□ "□□"□□□ □□□.

* □□□ □□□ □□ □□□ □□□ □ □□□□.

* □□□□ □□□ □□□□ □□□□ □□□ □□□□ □□ □□□.

* □□□□ □□□ □□□□ □□□ □□□ □ □□□□.

* □□□ □□ □□□ □□ □□□□□.

A. 2, 4

B. □, □

C. 4, 5

D. 2, 3, 4

Answer: A ([LEAVE A REPLY](#))

* The question is about marking the correct sentences among the given statements related to defects, failures, and mistakes. According to the ISTQB glossary, the definitions of these terms are1:

* Defect: A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system.

* Failure: An event in which a component or system does not perform a required function within specified limits.

* Mistake: A human action that produces an incorrect result.

* Therefore, out of the five given statements, only two are correct, namely:

* A human mistake may produce a defect: This is true, as a mistake is a source or cause of a defect, e.g. a programmer may make a mistake in writing a code statement, which results in a defect in the software component.

* When a defect exists in a system it may result in a failure: This is true, as a defect is a potential or actual cause of a failure, e.g. a defect in the software component may cause the system to fail to perform a required function when the defect is encountered during execution.

* The other three statements are incorrect, namely:

* Defects are a result of environmental conditions and are also referred to as "Failures": This is false, as defects are not a result of environmental conditions, but of mistakes or other factors, and defects are not the same as failures, but rather the causes of failures.

* A system will totally fail to operate correctly when a failure exists in it: This is false, as a system may not necessarily fail completely or stop operating when a failure occurs, but may continue to operate with reduced functionality or performance, or with incorrect results.

* Defects occur only as a result of technology changes: This is false, as defects can occur due to various reasons, not only technology changes, such as human mistakes, design flaws, requirement changes, hardware failures, etc.

$$\vdots$$

1: ISTQB Glossary of Testing Terms 4.0, 2023, available at ISTQB) and ASTQB).

NEW QUESTION: 106

[illegible]

□□□ □□□ □□□□ □□□□ □□ □□ □ □□□ □□□ □□□□ □□□ □□□□ □□□
□□ □□□□□□. □□ □□□ □□□ □□ □□ □□□ □□□□□.

□ □ □ □ | □ □ □

R500 ~ RIO 00010%

RIO 001 ~ R50 00011%

R50,001 ~ R100000012%

R100 000 | ~ R500 000 | 13%

□□□ □□□□ □□ □□ □□□ □□ □□□ □□□□ □ □□□ □□ □□□ □□□ □□ □□
 □□□?

A.5

B. 4

C. 8

D. 16

Answer: ([SHOW ANSWER](#))

Using equivalence partitioning, the investment ranges are divided into four partitions:

* R500 to R10,000 (10%)

* R10,001 to R50,000 (11%)

* R50,001 to R100,000 (12%)

* R100,001 to R500,000 (13%)

Thus, the minimum number of test cases required to cover all valid equivalence partitions for calculating the interest is 4.

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ □ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 107

□□ □□□ □□□□□□

int □□□□=2500;

□□ (□□<30)

{

□□□□ = □□□□ +1500:

}

□□ □□ □□ □□□ □□□ □□□ □□□ □□□□□? □ □□ □□□ □□□ □□□□
□ □□ □□ □□□ □□□.

A. 29, 1500

30. 2500

B. 29.4000

30. 2500

C. 29,2500

30. 1500

D. 30, 1500

31,2500

Answer: B ([LEAVE A REPLY](#))

In the given code snippet, the premium is increased by 1500 if the age is less than 30. Therefore, at the boundary value of age 29, the premium should be $2500 + 1500 = 4000$, and at age 30, the premium should remain at its initial value of 2500, as the condition is no longer met. Option B correctly reflects this with 29, 4000 and 30, 2500 as the boundary value and the expected results, respectively.

NEW QUESTION: 108

□□ □ □□ □□ □□□ □□ □□ □□□□□?

A. □□□ □□.

B. □□ □□

C. □□ □□□

D. □□ □□

Answer: ([SHOW ANSWER](#))

Boundary value analysis is not an experience-based technique, but rather a specification-based technique (also known as black-box technique). Experience-based techniques are techniques that rely on the tester's knowledge and intuition to derive and select test cases based on their experience with similar systems, technologies, domains, risks, etc. Some examples of experience-based techniques are error guessing, exploratory testing, fault attack, checklist-based testing, etc. Specification-based techniques are techniques that rely on the tester's analysis and interpretation of the requirements or specifications of the system under test to derive and select test cases based on some criteria or rules. Some examples of specification-based techniques are equivalence partitioning, boundary value analysis, decision table testing, state transition testing, etc. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 31.

NEW QUESTION: 109

□□□□ □□□□ □□ □ □□ □□□ □□□□□□.

i. □□□□ □□□ □□ □□□□ □□□□□.

ii. □□□□ □□□ □□□□ □□ □□□□ □□□ □□□□ □ □□□ □□□.

iii. □□□ □□□ □□□ □□□ □□□ □ □□□□.

iv □□ □□□□ □□□ □□ □□□ □□□ □□□□□.

□□ □ □□ □□ □□□ □□□□ □□□□□?

A. i □ iv

B. i □ iii

C. ii □ iv

D. iii □ iv

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 110

□□ □ □□ □□□ □□□□□?

A. □□ □□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□ □□□ □□□□ □□□ □ □□□ □□□ □□□□.

B. □□□ □□□□□ □□□ □□ □□□ □□□□ □□ □□ □□□ □□ □□□□ □□□□□.

C. □□ □□□□ □□□□ □□□□ □□□□□ □□□ □□□□ □□□□ □□□□□ □ □□ □□□ □□□ □□□□ □□□ □□□.

D. Non-functional testing includes testing of both technical and non-technical quality characteristics. Non-functional testing is the process of testing the quality attributes of a system, such as performance, usability, security, reliability, etc. Non-functional testing can be applied at any test level and can use both black-box and white-box test techniques. Non-functional testing can cover both technical aspects, such as response time, throughput, resource consumption, etc., and non-technical aspects, such as user satisfaction, accessibility, compliance, etc. Therefore, option B is the correct answer.

Answer: (SHOW ANSWER)

Non-functional testing includes testing of both technical and non-technical quality characteristics. Non-functional testing is the process of testing the quality attributes of a system, such as performance, usability, security, reliability, etc. Non-functional testing can be applied at any test level and can use both black-box and white-box test techniques. Non-functional testing can cover both technical aspects, such as response time, throughput, resource consumption, etc., and non-technical aspects, such as user satisfaction, accessibility, compliance, etc. Therefore, option B is the correct answer.

ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 1.3.1, page 13; ISTQB Glossary v4.

02, page 40.

NEW QUESTION: 111

Which of the following is a characteristic of user stories?

A. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

B. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

C. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

D. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

Answer: B (LEAVE A REPLY)

Acceptance criteria for user stories often include detailed specifications about data definitions, such as the format, allowed values, and default values for a data item. This helps ensure that the developed feature meets the expected requirements and provides a clear understanding for both developers and testers on what needs to be validated. Therefore, statement B is true as per the ISTQB CTFL syllabus.

NEW QUESTION: 112

Which of the following is a characteristic of user stories?

A. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

B. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

C. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

D. User stories are written from the perspective of the user. They are short, simple, and concise. They are written in a format that is easy to read and understand. They are written in a format that is easy to read and understand.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 113

□□ □□□ □□□□ □□ □□□ □□□□?

A. □□□ □□ □□□ □□□ □□□ □□□ □ □□□□.

B. □□□ □□ □□□ □□ □□□□ □□□□ □□□ □□□□ □□□ □ □□□□.

C. □□ □□□ □□□□□□ □□□□ □□ □□□□□□ □□□□□.

D. □□□ □□□□□ □□ □□ □□□ □□□ □□ □□□□.

Answer: B ([LEAVE A REPLY](#))

Configuration management is a process that establishes and maintains consistency among work products throughout their life cycle. Configuration management affects testing in various ways, such as:

- * Proper configuration management ensures that testers can uniquely identify the tested item, which can help traceability, reproducibility and accountability.
- * Proper configuration management ensures that testers have access to consistent versions of software components and testware, which can help reliability, compatibility and efficiency.
- * Proper configuration management ensures that testers can track changes and defects in software components and testware, which can help verification, validation and reporting.
- * Proper configuration management ensures that testers can control the configuration of the test environment, which can help stability, security and performance. Configuration management is not a prerequisite for test planning, as test planning can proceed without configuration management, although it may be less effective or accurate. Configuration management is not important for developers only, but for testers as well, as it affects the quality and consistency of the testing process and products.

Configuration management has a significant influence on the test project, as it affects various aspects of testing, such as traceability, reproducibility, reliability, compatibility, efficiency, verification, validation, reporting, stability, security and performance. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 6, page 60-61.

NEW QUESTION: 114

□□ □ □□□ □□ □□□ □□ □□□ □□ □□ □□□□□?

A. Wideband Delphi □□ □□□ □□ □□ □□ □□□ □ □□□□.

B. Wideband Delphi □□ □□□ □□□ □□ □□ □□□ □□□□.

C. Agile □□□ □□□□ Bumdown □□□ □□ □□ □□ □□□ □ □□□□.

D. Agile □□□ □□□□ □□□□ □□□ □□□ □□ □□ □□□ □ □□□□.

Answer: B ([LEAVE A REPLY](#))

There are two main approaches to test estimation:

- * Expert-based approach:
 - * The Wideband Delphi estimation technique is an example of this approach.
 - * Involves consulting expert (ISTQB not-for-profit association)ate testing efforts.
 - * This technique gathers estimates from multiple experts, discusses their differences, and achieves consensus.
- * Metrics-based approach:
 - * Relies on historical data and metrics to predict future testing efforts.

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, the expert-based approach relies on experts' experience and knowledge, which aligns with the Wideband Delphi technique#6source#.

References:

Certified Tester Foundation Level v4.0

ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 115

Shripriya is a software tester. She is working on a project where the requirements are not clear. She is confused about the requirements and is asking the project manager for clarification. The project manager is not able to provide a clear answer. Shripriya is feeling frustrated and is thinking of leaving the project.

- A. She should ask the project manager for clarification.
- B. She should ask the project manager for clarification.
- C. She should ask the project manager for clarification.
- D. She should ask the project manager for clarification.

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 116

Shripriya is a software tester. She is working on a project where the requirements are not clear. She is confused about the requirements and is asking the project manager for clarification. The project manager is not able to provide a clear answer. Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

1. Shripriya is feeling frustrated and is thinking of leaving the project.

2. Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

Shripriya is feeling frustrated and is thinking of leaving the project.

A. Shripriya is feeling frustrated and is thinking of leaving the project.

B. Shripriya is feeling frustrated and is thinking of leaving the project.

C. Shripriya is feeling frustrated and is thinking of leaving the project.

D. Shripriya is feeling frustrated and is thinking of leaving the project.

Answer: D ([LEAVE A REPLY](#))

In a planning poker session, if there is a significant difference in the estimations, it indicates that there may be misunderstandings or different perspectives on the complexity of the user story.

According to the agile principles, the team should discuss these differences to reach a common understanding. The goal is to ensure that all team members have a shared understanding of the user story's scope and complexity before finalizing the estimate.

NEW QUESTION: 117

□□ A□ □□□□□ □□□□ □□□□□□□□. □□ □□□ □□ □□□□ □□□□ □□□□□ □.

□□ □□□ □□□ □□□ □□□□□□?

A. □□: □□□ □□□ □□□□□ □□□□ □ □□ □ □□ □□□

□□: □□□ □□□□□□ □□□

B. □□: □□ □□□ □ □□□

□□□□ □□□□ □□□□ □□

C. □□: □□ □□□□ □□□ □□□□ □□□ □□, □□: □□□□ □□□ □□ □□□□ □□

D. □□: □□□□ □□□ □□ □ □□ □□□ □□□; □□: □□ □□□□ □□□ □□□□ □□ □ □□□

Answer: B ([LEAVE A REPLY](#))

Being part of the support team means that the testers are closer to the customer perspective, which is an advantage for testing, as they can better understand the user needs and expectations, and identify more realistic scenarios and risks. However, being part of the support team also means that they have less independence in deciding what and how to test, as they may be influenced by the customer's preferences or requests, which could compromise the objectivity and effectiveness of testing. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 6.

NEW QUESTION: 118

□□□ □□□□ □□□ □□□□ □□ □□, □□□ □ □□□ □□□ □□□ □□□□□□.

□□□ □□□□ □□ □□□□ □□ □□ □□□□□ □□ □□□□□□□ □□□□□?

A. V□□

B. □□□

C. □□

D. □□□□□ □□

Answer: A ([LEAVE A REPLY](#))

The test manager is most likely following the V-model for software development. The V-model is a software development model that defines four testing levels that correspond to four development phases: component testing corresponds to component design, integration testing corresponds to architectural design, system testing corresponds to system requirements specification, and acceptance testing corresponds to user requirements specification. The V-model also defines the test planning and test execution activities for each testing level. Agile is a software development model that follows an iterative and incremental approach, where testing is integrated into each iteration and adapts to changing requirements and feedback. Waterfall is a software development model that follows a sequential and linear approach, where testing is performed after the

development phase is completed. Prototyping is a software development model that involves creating a simplified version of the software to elicit user feedback and validate requirements before developing the final product. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 18.

NEW QUESTION: 119

□□ □ □□□ □□ □□□□ □□□ □ □□ □□ □□□□□?

I □□ □□ □□ □□□ □□□ □□ □□.

II. □□□□ □□□ □□□□ □□ □ □□.

III. □□, □□ □□ □□ □□□ □□ □□□ □□.

IV □□ □□□ □□ □□ □□ □□ □ □□ □□.

A. I, III, IV

B. I, II, III

C. III, IV

D. II, III, IV

Answer: ([SHOW ANSWER](#))

An exit criterion is a condition that defines when a test activity has been completed or when a test phase can be concluded. An exit criterion can be based on various factors, such as:

* I) Estimates of defect density or reliability measures. These are quantitative measures that indicate how many defects are present in the software product or how likely it is to fail under certain conditions.

These can be used as exit criteria to ensure that the software product meets a certain level of quality or performance before moving to the next phase or releasing it to the customer.

* III) Accuracy measures, such as code coverage, functionality coverage or risk coverage. These are quantitative measures that indicate how much of the software product has been tested in terms of its code, functionality or risk. These can be used as exit criteria to ensure that the test suite is adequate or complete before moving to the next phase or releasing it to the customer.

* IV) Residual risks, such as lack of code coverage in certain areas, unresolved defects or unknown factors. These are qualitative measures that indicate the remaining risks or uncertainties associated with the software product after testing. These can be used as exit criteria to ensure that the residual risks are acceptable or manageable before moving to the next phase or releasing it to the customer. The following factor is not a valid exit criterion:

* II) The completion and publication of an exhaustive Test Report. This is not a valid exit criterion, as it does not reflect the quality or completeness of the testing process or product. A test report is a document that summarizes the results and outcomes of a test activity or phase. A test report can be used as an input for deciding whether to exit a test activity or phase, but it is not a condition that defines when to exit. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 2, page 13; Chapter 6, page 58-59.

NEW QUESTION: 120

□□ □□ □□□ "□□□□"□ □□ □□ □□□ □□□□□.

1 - □□□ □□□□ □□□ 5□□ □□□□□.

2 - □□□ □□□□ □□ □□□□ □□□ □□□□ □□□.

□□ □□□ □□□ □□□ □□, □ □□□ □□□□□ □ □ □□ □□□ □□□□ □□□□□?

A. 8

B. 3

C. 6

D. 4

Answer: D ([LEAVE A REPLY](#))

Equivalence classes partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component. Equivalence classes partitioning can be used to reduce the number of test cases by selecting one representative value from each partition. For the ZIP code field, there are four equivalence classes based on the given rules:

* Valid ZIP code format and valid ZIP code value (e.g., 12345)

* Valid ZIP code format and invalid ZIP code value (e.g., 99999)

* Invalid ZIP code format and valid ZIP code value (e.g., 1234)

* Invalid ZIP code format and invalid ZIP code value (e.g., ABCDE) Therefore, four test cases are required to test this field, one for each equivalence class. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION: 121

□□ □□□□ □□ □□□□ □□ □□ □□ □ □□ □□ □□□□□?

A. □□□□ □□ □□□□ □□□ □ □□ □□□ □□□□ □□ □□ □□□□□ □□ □□□□ □□□□□.

B. □□ □□□□ □□ □□□ □□ □□□□ □□□ □ □□ □□, □□ □□□□ □□ □□□□ □□ □□□□ □□□ □ □□□□.

C. □□ □□□□ □□ □□□□ □□ □□□ □□□□□□, □□ □□□□ □□ □□□ □□□□ □□□□□□ □ □□ □□□□□.

D. □□ □□□□ □□□□□□ □□ □□ □□ □□□ □ □□□□ □□ □□□□ □□ □□ □□ □□ □□□□ □□□ □ □□□□.

Answer: A ([LEAVE A REPLY](#))

Static testing, such as code reviews and static analysis, is particularly effective at identifying issues related to code structure and modularization. These techniques allow for the inspection of the code without executing it, making it easier to spot problems related to how the code is organized. Dynamic testing, on the other hand, focuses on the execution of code and is better suited for identifying runtime issues but does not easily reveal structural problems. The ISTQB CTFL Syllabus v4.0 highlights the strengths of static testing in uncovering such structural issue

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □ □ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 122

□□ □ □□□ □□□ □□□□ □□□ □□ □□ □□□□□?

- A. □□ □□ □□□ □□ □□
- B. □□□□ □□ □□
- C. □□□□ □□□□ □□ □□□
- D. □□□□ □ □ □□ □□

Answer: ([SHOW ANSWER](#)**)**

The extent of testing required for a software product depends on various factors, such as the level of risk, the budget, and the time available. The level of risk reflects the potential impact of failures on the stakeholders and the environment. The budget determines how much resources can be allocated for testing. The time available defines the schedule and deadlines for testing activities. The particular tester involved in testing is not a deciding factor for the extent of testing required, as testing should be based on objective criteria and not on personal preferences or abilities.

Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 2, page 14-15.

NEW QUESTION: 123

□□ □□ □ □□ □□□□□ □□ □□□□□□□ □□□□ □□ □□□ □□□ □□□□ □□ □□□□□?

- A. □ □□□ □□□□ □□□□□ □□ □□□□□□ □□□ □□ □□□□ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□□ □□□ □□□.
- B. □□ □□□ □□□ □□ □□□ □□ □ □□□ □□□ □□□□ □□ □□□ □□ □□□ □□ □□ □□□ □ □□ □□ □□ □□□□□ □□□.
- C. □□□ □□□ □□□ □□□□ □□ □□□□□ □□□□ □□□ □□ □□□□□.
- D. □□ □□□□ □□□□□□ □□ □□ □□□□□ □□ □ □□□□ □□ □□□□□ □□□ □□□□□ □□□.

Answer: A ([LEAVE A REPLY](#)**)**

Good testing practice dictates that each test level (e.g., unit testing, integration testing, system testing) should have distinct test objectives aligned with the phase of the software development lifecycle it addresses. This ensures that testing is effective and relevant at each stage. According to the ISTQB CTFL Syllabus v4.0, establishing clear test objectives that are consistent with the development phase helps in achieving specific goals and improving the overall quality of the software product.

NEW QUESTION: 124

□□□ □□ □□□□ □□□ □□□□ □□□.

A. □□ □□ □□ □□□ □□□ □□□□ □ □□□ □□ □□□ □□ □□□□ □□ □□ □□□□ □□□□□□.

B. □□□ □□ □□□ □□□ □□□□ □□ □□ □□□ □□□□ □ □□□ □□ □□□□ □□ □□ □□□ □□□□□.

C. □□□□ □□□ □□□ □□□□ □□□□ □□□ □□ □□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□□ □□□□ □□ □□□.

D. □□□ □□□□ □□ □ □□□ □□□, □□ □□□□□□ □□ □ □□ □□ □□ □□□□□ □ □□□ □□□□□.

Answer: A ([LEAVE A REPLY](#))

A test status report is a document that provides a snapshot of the testing activities and their progress during a particular period. It should include information about any impediments encountered during the test execution and the actions taken to resolve them, which helps stakeholders understand the challenges and how they were addressed .

Option B describes an activity related to test completion rather than ongoing status reporting.

Option C is incorrect because the structure and contents of the report may vary based on the audience's needs. Option D, while important, is not the primary purpose of a test status report, which focuses more on the current status and impediments.

NEW QUESTION: 125

□□□□□ □□□ □□□ □□ □□ □□ □ □□ □□ □□□□□?

A. □□□□□ □□□ □□ □□ □□ □□ □ □□ □□ □□□ □□□□□ □□ □□ □□□□□ □□□ □□□ □□□□□□□□ □□ □□ □ □□ □□□ □□□ □□ □□□□□.

B. □□□□ □□□□ □□□ □□ □□ □□□□□ □□□□ □□ □□ □□ □□□□□ □□□ □□□ □□□□ □□□□. □□□□ □□ □□□□□ □□□□ □□□ □□□ □□□ □□□□ □□□ □□□□ □□□ □ □□ □□□□□.

C. □□ □□□ □□ □□□ □□ □□ □□□□□ □□□ □□□□□ □□ □□□ □□□ □□□ □□□ □□□□ □□ □□ □□ □□ □□ □□□□ □□ □□ □□□ □□□□□ □□□□□.

D. □□ □□ □□□□□ □□□ □□□ □□□□□ □□□□ □□□ □□□□ □□□ □□ □□ □□ □□□ □□ □□□□ □□□ □□□ □□□ □□□ □□□ □ □□□ □ □□□□.

Answer: D ([LEAVE A REPLY](#))

This answer is correct because code-related white-box test techniques are test design techniques that use the structure of the code to derive test cases. They provide an objective measure of coverage, such as statement coverage, branch coverage, or path coverage, which indicate how much of the code has been exercised by the test cases. Code-related white-box test techniques can be used to complement black-box test techniques, which are test design techniques that use the functional or non-functional requirements of the system or component to derive test cases. By combining both types of techniques, testers can increase their confidence in the code and find more defects. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.2

NEW QUESTION: 126

- □□□□□□□ □□□ S100□ □□□□ □ □□□□ 50% □□□ □□ □□□□□ □. □□□ □□ □□□ □□□ □□□□ 5%□ □□□□ □□□ □□□ □□□□□□. □□ □□ □□ □□ □□□□□□ □□ □□□ 5%□ □□ 50%□ □□ □□□□ □□□ □□□□□□. □□ □ □ □□□□□ □□ □□□□ □□□□ □□ □□□□□?
- A. □□ □□□ □□ □□□□, 50%□ □□□ □□□ □□□□, □□□ □□ □□□ □□□□□.
- B. □□□ □□ □□□ □□ □□ □□□ □□□□ □ □□□ □□ □□□□□.
- C. □□□ □□□ □□ □□□□□ □□□□□ □□, □□ □ □□ □□□ □□□□ □□ □□ □□ □□□□□.
- D. □□□ □□□ □□□ □□□ □□ □□□□ □□□□□□□.

Answer: B ([LEAVE A REPLY](#))

According to the ISTQB CTFL syllabus, a defect is a deviation from the expected result which in this scenario is the incorrect discount applied to the customers. The root cause, as per the ISTQB definition, is the originating cause of a defect, which in this case is the incorrect requirement stating 50% instead of 5%.

Therefore, the incorrect requirement is the root cause and the customer receiving the wrong discount is the effect of this root cause.

References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.3 "Defects, Root Causes, and Effects".

NEW QUESTION: 127

- □□ □ "□□□" □□□ □□ □□□□ □□□□ □□ □□□□□?
- "□□□"?
- A. □□□□ □□□□□□ □□□□ □□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□ □□(□□)□ □□, □□□ □□□ □□□□, □□□□ □□ □□□ □□□□.
- B. □□□□ □□□□□□ □□□□ □□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□(□□)□ □□□ □□, □□□ □□□ □□□□, □□□□ □□ □□□ □□□□.
- C. □□□□ □□□□□□ □□□ □□ □□□□ □□□ □□□□ □□ □□□□ □ □□□ □□ □ □□ □□□□ □□□□ □□ □□□ □□□□.
- D. □□□□ □□□□□□ □□□ □□ □□□□ □□□ □□□□ □□ □□□□ □ □□□ □□ □ □□ □□□□ □□□□ □□ □□□ □□□□.

Answer: ([SHOW ANSWER](#)**)**

Comprehensive and Detailed In-Depth Explanation:

Testing and debugging are separate but related activities. Testing executes the software to identify failures that result from defects (A). Debugging is the developer's responsibility and involves finding the cause of a failure (defect), analyzing it, and fixing the defect. The ISTQB syllabus explicitly differentiates these activities. Testing does not modify the software, whereas debugging does.

Reference: ISTQB CTFL v4.0 Syllabus, Section 1.1.2 - Testing and Debugging

NEW QUESTION: 128

DevOps□ □□□□ □□□ □□□ □□ □□ □□ □ □□ □□ □□□□?

- A.** DevOps □ □□□ □□□ □□□□ □□□ □□ □□ □□□□ □□□ □ □□□ □□□.
- B.** DevOps □ □□□ □□□□□ □ □□□□ □□ □□ □□□□ □□□□ □ □□□ □□□.
- C.** DevOps □ □□□ □□ □□□□ □□□ □□□ □ □□□ □□□.
- D.** DevOps □ □□ □□□□ □□□ □□ □□□□ □□□□ □ □□□ □□□.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 129

□□ □□ □ □□ □□ □□□□□?

- A.** □□ □□□□ □□ □□□ □□□□ □□ □□□□□ □□□ □ □□□, □□ □□ □□ □□□ □□ □□□ □ □□□□.
- B.** □□ □□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□, □□□□□ □□□□ □□ □□ □□ □ □□ □□□ □ □□□□□□ □□□ □□□□.
- C.** □□ □□□□ □□ □□□ □□□ □□□ □ □□□, □□□ □□□□ □□□ □ □□ □□□ □□□□ □□□ □ □□□□.
- D.** □□□□ □□□ □□□ □□ □□ □□□ □□□ □□ □□□□ □□□ □□□ □□□ □□□ □□□ □□□□.

Answer: D (LEAVE A REPLY)

Statement D is correct. According to the ISTQB CTFL syllabus, both black-box test techniques (which focus on testing without internal knowledge of the application) and experience-based test techniques (which rely on testers' experience and intuition) can be applied to both functional and non-functional testing. Functional testing is concerned with what the system does, whereas non-functional testing looks at how the system performs under certain conditions. These techniques are versatile and can be employed to address both these aspects.

NEW QUESTION: 130

□□ □□ □□ □□□ □□□□ □□□ □□□□□?

- A.** □□ 300□□ □□ □□□ □□□□□ □□□□□□□ □□ □□□□ □□□□ □□ □ □□□ □.
- B.** □□□ □□□ □□□ □□ □□ □□
- C.** □□□ □□□ □□□ □□□□ □□ □□□ □□ □□□□□□ □□ □□□ □□□ □□□ □ □□□□.
- D.** □□□□ □□□ □□□ □□ □□□ □□ □□□ □□□□□.

Answer: A (LEAVE A REPLY)

This question relates to identifying product risks, which are potential problems associated with the product itself, such as software functionality, reliability, usability, and performance. Option A describes a scenario where the application might not meet performance requirements under specific conditions (up to 300 concurrent users), which directly impacts the product's ability to perform its intended function. This is a classic example of a product risk, as it concerns the product's quality and its ability to meet user needs.

Options B, C, and D, on the other hand, relate to project risks, which are concerns related to the management and execution of the project, such as tool acquisition, environment configuration, and team expertise, rather than the quality of the product itself.

NEW QUESTION: 131

□□ □ □□□□ □□□ □□ □□□ □□□□□?

A. □□□ □□

B. □□□

C. □□ □□

D. □□□ □□

Answer: B ([LEAVE A REPLY](#))

A key characteristic of informal reviews is low cost. Informal reviews are a type of review that does not follow a formal process or have any formal documentation. Informal reviews are usually performed by individuals or small groups of peers or colleagues who have some knowledge or interest in the product under review. Informal reviews can be done at any time and for any purpose, such as checking for errors, clarifying doubts, sharing ideas, etc. Informal reviews have low cost, as they do not require much time, effort, or resources to conduct. The other options are not key characteristics of informal reviews. Kick-off meeting is a characteristic of formal reviews, such as inspections or walkthroughs. Kick-off meeting is a meeting that is held before the review process starts, where the roles and responsibilities of the participants are defined, the objectives and scope of the review are agreed, and the logistics and schedule of the review are planned. Individual preparation is a characteristic of formal reviews, such as inspections or walkthroughs. Individual preparation is an activity that is performed by the reviewers before the review meeting, where they examine the product under review and identify any issues or questions that need to be discussed or resolved during the review meeting. Metrics analysis is a characteristic of formal reviews, such as inspections or walkthroughs.

Metrics analysis is an activity that is performed after the review process is completed, where the data and results of the review are collected and analyzed to measure the effectiveness and efficiency of the review, as well as to identify any improvement actions or lessons learned for future reviews. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 9.

NEW QUESTION: 132

□□□□ □□□ □□□ □ □□ □□□ □□ □□ □□ □□□□□.

* □□□ □□□□ □□(□□ □□□ □□)□□ □□□□□.

* □□ □□□ □□□ □□ □□□ □□ □□

* □□□ □□□ □□□ □□□ □□□□ □□□ □□□□□ □□□□□. □ □□□ □□ □□ □ □
□□ □□□□ □□□?

A. □□□ □□ □□

B. □□□ □□□□ □□□□ □□ □□□ □□□□ □□□ □□□ □□

C. □□□ □□□ □□

D. □□□ □□□ □□□ □□□ □ □□ □□ □□

Answer: ([SHOW ANSWER](#))

Comprehensive and Detailed In-Depth Explanation:

During iteration planning, testers provide input on risk analysis, testability, and the feasibility of test automation. Participating in risk analysis (B) helps identify areas needing more extensive testing.

* (A) is incorrect because the test strategy is usually set at a higher level (project or release level).

* (C) is incorrect as release planning is broader than iteration planning.

* (D) is incorrect since writing user stories is mainly the responsibility of the product owner, though testers contribute to defining acceptance criteria.

Risk-based test prioritization ensures testing efforts are aligned with business needs and software risks.

Reference: ISTQB CTFL v4.0 Syllabus, Section 5.2.3 - Risk-Based Testing

NEW QUESTION: 133

□□□□□ □□ □□□ □□□□ □ □□□□□(□□ □□/□□, □□□□ □□ □□/□□, □□□ □□□ □□ □). □□ □□□ □□ □□□ □□ □□□□. □□ □□, □□ □□□ □□ □ □□□ □ □□ □□ □□ □□□ □□□□. □□□□ □□□ □□ □□(□□ □□□ □□□□ □□ □□) □ □□□□ □□□□□ □□□□□ □□□□□.

□ □□□ □□□ □□□ □□□□□ □□ □□ □□ □□□ □□□□□?

A. □□ □□□□

B. □□□ □□ □□

C. □□ □□

D. □□ □□

Answer: ([SHOW ANSWER](#))

Branch coverage is a type of structural coverage metric that measures the percentage of branches or decision outcomes that are executed by the test cases. A branch is a point in the code where the control flow can take two or more alternative paths based on a condition. For example, an if-else statement is a branch that can execute either the if-block or the else-block depending on the evaluation of the condition. Branch coverage ensures that each branch is taken at least once by the test cases, and thus reveals the behavior of the software under different scenarios. Branch coverage is also known as decision coverage or all-edges coverage.

Branch coverage is suitable for testing the cases where a specific action is not allowed, because it can verify that the test cases cover all the possible outcomes of the conditions that determine the action. For example, if the program has a condition that checks if the manufacturing stage can be stopped, then branch coverage can ensure that the test cases cover both the cases where the stage can be stopped and where it cannot be stopped.

This way, branch coverage can help identify any missing or incorrect branches that may lead to undesired or unsafe actions.

The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types

of coverage metrics, such as statement coverage, branch coverage, data flow coverage, etc. Code coverage does not specify which type of coverage metric is used for the analysis. Data flow coverage is a type of structural coverage metric that measures the percentage of data flow paths that are executed by the test cases. A data flow path is a sequence of statements that define, use, or kill a variable. Data flow coverage is useful for testing the correctness and completeness of the data manipulation in the software, but not for testing the conditions that determine the actions. Statement coverage is a type of structural coverage metric that measures the percentage of statements or lines of code that are executed by the test cases. Statement coverage ensures that each statement is executed at least once by the test cases, but it does not reveal the behavior of the software under different scenarios.

Statement coverage is a weaker criterion than branch coverage, because it does not account for the branches or decision outcomes in the code. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

NEW QUESTION: 134

□□□□□ □□□□ □□□□ □ □□□ □□□□ □□ □□ □□□ □□□□. □□□ □□□ □ □□ □□□□□□ 12□□ □□ □□□□ □□□□□. □□□ □□ □□□ □□□ □□ □□ □ □ □□□□.

'□□ □□□□ □□□□ □ □□□ □□□ 2□ □□□□□ □□□.' □□ □ □□□ □□ □□□ □ □□ □□□ □□□□□?

- A. □□□ □□□ □ □□ □□□□ □□□□ □□□□.
- B. □□ □□□□ □□□□ □ □□□ □□ □□ □□□□.
- C. □□□ □□□ □□□□ □□ □□□□ □□□□□□.
- D. □□□ □□□ □□□ □□□□ □□ □□ □□□□ □□□□□□.

Answer: ([SHOW ANSWER](#))

A failure is an event in which a component or system does not perform a required function within specified limits¹. A requirement is a condition or capability needed by a user to solve a problem or achieve an objective². In this case, the requirement is that the diagnostics tests should execute in less than 2 seconds.

Therefore, any event that violates this requirement is a failure. The only option that clearly violates this requirement is B. The diagnostic tests take too much time to execute. If the diagnostic tests take more than 2 seconds to complete, then they do not meet the specified limit and thus fail. The other options are not necessarily failures related to the specified requirement. Option A. The diagnostic tests fail to start after a system reset is a failure, but not related to the time limit. It is related to the functionality of the self- diagnostics module. Option C. The diagnostic tests that measure the speed of the memory, fail is also a failure, but not related to the time limit. It is related to the accuracy of the memory tests. Option D. The diagnostic tests fail due to incorrect implementation of the test code is also a failure, but not related to the time limit. It is related to the quality of the test code. References = ISTQB Certified Tester Foundation Level Syllabus v4.0, Requirements Engineering Fundamentals.

NEW QUESTION: 135

□□□ □□ □□□ □, □□□ □□□□ □□□ □□□ □□□□ □□□ □□□□□?

A. □□□□

B. □□□

C. □□

D. □□

Answer: B ([LEAVE A REPLY](#))

In defect reporting, the attribute that indicates the degree of impact that the defect has on the system is the severity. Severity reflects the seriousness of the defect in terms of its impact on the operation of the system, ranging from minor issues that do not significantly affect the system's functionality to critical defects that can cause system failure. Therefore, option B is the correct answer.

NEW QUESTION: 136

TDD, BDD, ATDD□ □□ □□ □□ □ □□ □□ □□□□□?

A. □□□□□ TDD□ □□□□ □□□ □□□□ □□□□ □□ □□□□ □□□□□ □□ □□□ □□□□□.

B. ATDD□ □□ □□□ □□□□□ □□□ □ □□ □□□□ □□□ □□ □□□□□.

C. BDD□ □□□□ □□/□□ □□ □□□ □□□□ □□ □□□□□ □□□□ □□ □□□□ □ □□□□ □□□□ □□ □□□ □□□□□.

D. ATDD□ □□□□ □□ □□□□□ □□□ □□□□ □□ □□□□ □□□□ □□□□□.

Answer: (SHOW ANSWER)

Test-Driven Development (TDD) emphasizes writing tests before code and includes refactoring as a key practice to improve both the tests and the code. This ensures that the codebase remains clean and maintainable. The ISTQB CTFL Syllabus v4.0 discusses TDD as a practice that includes writing tests first, coding to satisfy those tests, and then refactoring the code to improve its structure and readability while keeping the tests intact.

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □ □ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 137

□□ □□□ □□□ □□□□□□ □□□□ □ □□ □□□□ □□ □ □□□(TAS□□ □)□ □□ □□□□□.

TAS 50k€ 80k€ 10%. TAS 80k€ 120k€ 15%. TAS 120k€ 140k€ 22%.

40k€, 15%. TAS

15% 22%.

TAS (1,000 = 1,000).

?

A. TC1 = 81 k€; TC2= 97k€; TC3=111k€; TC4=118k€

B. TC1 = 40k€; TC2= 46k€; TC3=51k€; TC4=53k€

C. TC1 = 79k€; TC2= 80k€; TC3=81k€; TC4=82k€

D. TC1 = 90k€; TC2= 110k€; TC3=125k€; TC4=140k€

Answer: A (LEAVE A REPLY)

This answer is correct because equivalence partitioning is a test design technique that divides the input domain of a system or component into partitions of equivalent data, such that each partition is expected to produce the same output or behavior. Equivalence partitioning aims to reduce the number of test cases by selecting one representative value from each partition. In this case, the input domain of the TAS can be divided into four partitions based on the bonus rules: less than 50k€, between 50k€ and 80k€, between 80k€ and 120k€, and more than 120k€. The test cases in the answer belong to the same partition, which is between

80k€ and 120k, and they are expected to produce the same output, which is a bonus of 15%.

References:

ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.1

NEW QUESTION: 138

.

a.

b.

c.

d.

1.

2.

3.

4.

5.

6.

?

A. 2-a, 4-a, 5-b, 6-c, 1-d, 3-d

B. 2-a, 5-a, 1-b, 4-b, 3-c, 6-d

C. 1-, 4-, 5-, 6-, 2-, 3-

D. 2-a, 4-b, 5-c, 1-d, 3-d, 6-d

Answer: (SHOW ANSWER)

Matching the main review process activities with the specific review activities, we see that planning includes estimating effort and timeframe (2) and selecting people to participate (4). Initiating a review involves distributing work products and other material (5). Issue communication and analysis includes evaluating the review findings (6). Fixing and reporting would entail creating defect reports (1) and recording the updated status of defects (3).

References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2 "Review Process".

NEW QUESTION: 139

□□ □ □□ □□ □□□ □□ □□□ □□ □□ □□ □□□□□?

- A. □□□ □□□□ □□□ □□□□□.
- B. □□□□□□ □□□ □□□ □ □□□□.
- C. □□ □□ □□ □□ □□□□□□ □□□□□ □□□□□□.
- D. □□□ □□ □□ □□□ □□□□ □□□ □ □□□□.

Answer: C ([LEAVE A REPLY](#))

Component testing (also known as unit testing or module testing) is a level of testing that focuses on verifying the functionality and quality of individual software components (such as modules, classes, functions, methods, etc.). Component testing mainly tests interfaces and interaction between components, as well as internal logic and data structures of the components.

Component testing may be applied using a test-first approach (such as test-driven development or behavior-driven development), where tests are written before the code is implemented.

Component testing does not identify defects in modules and classes, as this is a result of component testing, not an objective. Simulators and stubs may be required for component testing, as they can simulate or replace missing or incomplete components or external systems that are needed for testing. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page

19.

NEW QUESTION: 140

□□ □□□ □□ □□□ □□□ □□□ □□□□□? □□□□ 1□□ 15□□□ □□ □□□□ □□□.

- A. 0.1 □□ ~ 14. 15 □□
- B. 1.1 □□ ~ 14. 15 □□
- C. □□. 1~15. 15 □□
- D. 1.1 □□ ~ 15. 15 □□

Answer: D ([LEAVE A REPLY](#))

Equivalence partitioning is a black-box test design technique where inputs to the software or system are divided into groups that are expected to exhibit similar behavior. For an integer field that should accept values from 1 to 15, the valid equivalence class is 1 through 15. The invalid equivalence classes are numbers less than 1 and numbers more than 15. Therefore, option D, "Less than 1, 1 through 15, more than 15," correctly identifies the valid equivalence class along

with the two invalid classes, covering all possible input scenarios for the field. Options A, B, and C either do not accurately capture the valid range or incorrectly specify the range boundaries.

NEW QUESTION: 141

□□□ □□□□ □□□□ □□□ □□□ □ □□□□.

A. □□□ □□□ □□□□□.

B. □□□ □□□ □ □□□□ □□□□ □□□□□.

C. □□□ □□□ □□□ □□□□

D. □□ □□□ □□□□ □□ □□□ □□□□ □□□□ □□□□.

Answer: B ([LEAVE A REPLY](#))

Test automation allows you to produce tests that are less subject to human errors, as they can execute predefined test scripts or test cases with consistent inputs, outputs, and expected results. Test automation can also reduce the manual effort and time required to execute repetitive or tedious tests, such as regression tests, performance tests, or data-driven tests. Test automation does not demonstrate the absence of defects, as it can only verify the expected behavior of the system under test, not the unexpected or unknown behavior. Test automation does not avoid performing exploratory testing, as exploratory testing is a valuable technique to discover new information, risks, or defects that are not covered by automated tests. Test automation does not increase test process efficiency by facilitating management of defects, as defect management is a separate activity that involves reporting, tracking, analyzing, and resolving defects, which may or may not be related to automated tests. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 3.3.1, Test Automation1

* ISTQB Glossary of Testing Terms v4.0, Test Automation2

NEW QUESTION: 142

□□□□□□ □ □□ □□□ □□□□ □□ □□□ □□□ □□□□□.

- E; □□□ □□ □□

- P : □□□ □

-T □□□ □□

P□ □□ □□ 15□ 350 □□□ □□ □□ □□ □ □□□□.

□□ □□ □ P □□□ □□ □□ □□ □□□ □□□ □□□□□?

A. 14,351

B. 14,15,350,351

C. 15,350

D. 5.175.500

Answer: B ([LEAVE A REPLY](#))

A correct list of boundary values for the P input should include the minimum and maximum values of the valid range (15 and 350), as well as the values just below and above the boundaries (14 and 351). Boundary value analysis is a test design technique that involves testing the values at or near the boundaries of an input domain or output range, as these values are more likely to cause

errors than values in the middle. Option B satisfies this condition, as it has all four boundary values (14, 15, 350, 351). Option A has only two boundary values (14 and 351), option C has only two boundary values (15 and 350), and option D has no boundary values at all. Verified
References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 34.

NEW QUESTION: 143

□□□□□ □□ □□ □□□ □□ 100% □□ □□□ □□□□ □□ □□ □□ □ □□ □□
100%□ □□□□□?

- A. 100% □□□ □□ □□
- B. 100% □□ □□□ □□ □□
- C. 100% □□□ □□
- D. 100% □□ □□

Answer: D ([LEAVE A REPLY](#))

Statement coverage is a structural coverage metric that measures the percentage of executable statements in the source code that are executed by a test suite¹. Decision coverage is another structural coverage metric that measures the percentage of decision outcomes (such as branches or conditions) in the source code that are executed by a test suite¹. Decision coverage is a stronger metric than statement coverage, because it requires that every possible outcome of each decision is tested, while statement coverage only requires that every statement is executed at least once². Therefore, if a test suite achieves 100% decision coverage, it also implies that it achieves 100% statement coverage, because every statement in every branch or condition must have been executed. However, the converse is not true: 100% statement coverage does not guarantee 100% decision coverage, because some branches or conditions may have multiple outcomes that are not tested by the test suite². For example, consider the following pseudocode: if (x > 0) then print("Positive") else print("Non-positive") end if
A test suite that executes this code with x = 1 and x = -1 will achieve 100% statement coverage, because both print statements are executed. However, it will not achieve 100% decision coverage, because the condition x

> 0 has only been tested with two outcomes: true and false. The third possible outcome, x = 0, has not been tested by the test suite. Therefore, the test suite may miss a potential bug or error in the condition or the branch.

The other options, such as state transition coverage, equivalence class coverage, and boundary value coverage, are not guaranteed to be 100% by achieving 100% decision coverage. State transition coverage is a structural coverage metric that measures the percentage of transitions between states in a state machine that are executed by a test suite³. Equivalence class coverage is a functional coverage metric that measures the percentage of equivalence classes (or partitions) of input or output values that are tested by a test suite⁴. Boundary value coverage is another functional coverage metric that measures the percentage of boundary values (or extreme values) of input or output ranges that are tested by a test suite⁴. These metrics are independent of decision coverage, because they are based on different aspects of the system under test, such as its behavior, functionality, or specification. Therefore, achieving 100% decision coverage does

not imply achieving 100% of any of these metrics, and vice versa. References = ISTQB Certified Tester Foundation Level Syllabus v4.0, Test Coverage in Software Testing - Guru99, Structural Coverage Metrics - MATLAB & Simulink - MathWorks India, Test Design Coverage in Software Testing - GeeksforGeeks.

NEW QUESTION: 144

□□□ □□ □□ □□□□□□ □□ □□ □□ □□□□ □□□□ □□□ □□□□□□. □□□□ □□ □□ □□ □□□ □□ □□□□ □□□□ □□□□.

□□ □□ □ □□ □□ □□□ □□ □□□ □□□□ □□□□□?

- A. □□□ □□□ □□ □□□□ □□ □□□ □□ □□
- B. □□ □□□ □□□ □ □□□ □□□ □□ □□□□.
- C. □□ □□ □ □□□ □□□ □□ □□□ □□ □□□ □□
- D. □3□ □□□ □□□ □□□□ □□□ □□ □□□ □□□□□.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 145

□□ □□ □ □ □□ □□□ □□□(Q3)□ □□ □ □□□□ □□ □□□□□?

- A. □ □□□ □□□□□ □□□ □□□□ □□□ □□□(□□□□ □□□ □□)□ □□□□□□. □□ □ □□□□□ □□ □□□□□□.
- B. □ □□□ □□□□ □□□□ □□□□□ □□□□ □□□□□. □□ □□□, □□□□ □□□, □ □□ □□ □□□□ □□□□ □□□□.
- C. □ □□□ □□□□□ □□ □□□, □□, □□□ □□□ □□□, □□□ □□ □□□□□, API □□□ □ □□□□□□ □□□□ □□□□. □□□ □□□□ □□ □□□ □□□□ □□ □□ □ □□□ □ □□□□.
- D. □ □□□ □□□□ □□ □□□□ □□ □□□□□. □□□ □□□□ □□□□□□ □□ CI(Continuous Integration) □□□□□ □□□□□ □□□.

Answer: ([SHOW ANSWER](#))

Comprehensive and Detailed In-Depth Explanation:

The Agile Testing Quadrants framework categorizes tests based on their purpose and audience:

- * Quadrant 1 (Q1): Technology-facing tests (unit and component tests).
- * Quadrant 2 (Q2): Business-facing tests supporting development (e.g., BDD tests).
- * Quadrant 3 (Q3): Business-facing tests that critique the system (B).
- * Includes usability testing, exploratory testing, and UAT to ensure software meets user expectations.
- * Quadrant 4 (Q4): Technology-facing tests that critique the system (e.g., performance, security testing).

Option B correctly defines Q3 since it focuses on evaluating the user experience, exploring the system, and validating business expectations.

Reference: ISTQB CTFL v4.0 Syllabus, Section 5.1.7 - Agile Testing Quadrants

NEW QUESTION: 146

□□ □ □□ □□ □□□ □□□□□?

- A. □□□, □□□ □□ □ □□□□□ □□ □□ □□ □□ □□
- B. □□□□ □□ □ □□□ □□, □□ □ □□ □□□ □□□□□□□.
- C. □□□ □□□□ □□ □□□ □□ □□ □□□□ □□□□□ □□□□□.
- D. □□ □□□□ □□□ □□□ □□□□ □□ □□□ □□□□□.

Answer: ([SHOW ANSWER](#))

A dynamic analysis tool is a tool that performs analysis of a software product based on its behavior during execution. A dynamic analysis tool can monitor various aspects of a program's run-time performance, such as memory usage, CPU load, response time, or resource leaks. A dynamic analysis tool can monitor the allocation, use and de-allocation of memory during run-time of a program, which can help detect defects such as memory leaks, buffer overflows, or memory corruption. A dynamic analysis tool cannot provide support for traceability of tests, test results and incidents to source documents, as this is a function of a test management tool. A dynamic analysis tool cannot execute programs step-by-step in order to reproduce failures and find corresponding defects, as this is a function of a debugging tool. A dynamic analysis tool cannot provide support for release of baselines consisting of configuration items, as this is a function of a configuration management tool. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 6, page 56-57.

NEW QUESTION: 147

□□ □ □□ □□□ □□□□□?

- A. □□ □□ □□□ □□□ □□□□ □□□ □□□ □□□ □□ □□□ □□□□ □□ □□ □□□ □□□□□.
- B. □□□□□ □□□ □□□□ □□ □□□□□ □□□□ □□□ □□□□ □□□ □□□, □□ □□ □ □□□□ □□□□□ □□□□□.
- C. □□ □□ □□□ □□□ □□□□ □□□ □□□ □□□ □□ □□ □□□ □□□ □□□□ □□ □□□□□.
- D. □□ □□ □□□ □□□ □□ □□□ GUI □□ □□□ □□□ □□□ □□□□ □□ □□□□ □□ □□□ □□□ □□□ □□□□ □□□□.

Answer: C ([LEAVE A REPLY](#))

Experience-based test techniques are test design techniques that rely on the experience, knowledge, intuition, and creativity of the testers to identify and execute test cases that are likely to find defects in the software system. Experience-based test techniques are often useful to detect hidden defects that have not been targeted by black-box test techniques, which are test design techniques that use the external behavior and specifications of the software system as the test basis, without considering its internal structure or implementation. Experience-based test techniques can complement black-box test techniques by covering aspects that are not explicitly specified, such as usability, security, reliability, performance, etc. The other statements are false, because:

* Experience-based test techniques do not rely on the experience of testers to identify the root causes of defects found by black-box test techniques, but rather to identify the potential sources

