



BCS

Exam Questions CTFL4

ISTQB Certified Tester Foundation Level CTFL 4.0 Exam

NEW QUESTION 1

A virtual service emulating a real third-party service and the automated test scripts (aimed at testing the system under test) that interact with that service, are test work products that are typically created during:

- A. Test monitoring and control
- B. Test implementation
- C. Test design
- D. Test analysis

Answer: B

Explanation:

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 2

Which statement is true regarding confirmation testing and regression testing?

- A. Confirmation testing confirms the quality of the test being run while regression testing ensures that the software still works after a change has been made.
- B. Confirmation testing is an optional activity whilst regression testing is not negotiable.
- C. Confirmation testing aims to verify that a defect has been resolved and regression testing ensuring that existing functionality still works after a change.
- D. Testers' involvement is essential whilst running retesting and regression testing.
- E. TESTER Involvement is essential whilst running retesting and regression testing.

Answer: C

Explanation:

Confirmation testing, also known as retesting, is conducted to verify that specific defects have been fixed. Regression testing, on the other hand, is performed to ensure that recent changes have not adversely affected existing features of the software. Both types of testing are crucial for maintaining the integrity and quality of the software after modifications.

NEW QUESTION 3

Consider the following user story about the authentication functionality of an e-commerce website:

"As a logged-in user, I want to change my current password with a new one, so that I can make my account safer".

The following are some of the acceptance criteria defined for the user story:

- [a] After the logged-in user has successfully changed his password, an email confirming the change must be sent to him
 - [b] To successfully change the password, the logged-in user must enter the current password, enter a new valid password, and finally confirm by pressing the 'Change Password' button
 - [c] To be valid, the new password entered by the logged-in user is not only required to meet the criteria related to the length and type of characters, but must also be different from the last 5 passwords of that user
 - [d] A dedicated error message must be presented to the logged-in user when he enters a wrong current password
 - [e] A dedicated error message must be presented to the logged-in user when he enters the correct current password, but enters an invalid password
- Based only on the given information, which of the following ATDD tests is most likely to be written first?

- A. The logged-in user enters a wrong current password and views the dedicated error message
- B. The logged-in user enters the correct current password, enters a valid new password (different from the last 5 passwords), presses the Change Password' button, and finally receives the e-mail confirming that the password has been successfully changed
- C. The logged-in user enters the correct current password, enters an invalid password, and finally views the dedicated error
- D. The logged-in user submits a purchase order containing ten items, selects to pay with a Visa credit card, enters credit card information of a valid card, presses the 'Confirm' button, and finally views the dedicated message confirming that the purchase has been successful

Answer: B

Explanation:

ATDD stands for Acceptance Test-Driven Development, which is a collaborative approach to software development and testing, in which the acceptance criteria of a user story are defined and automated as executable tests before the implementation of the software system. ATDD tests are usually written in a Given-When-Then format, which describes the preconditions, the actions, and the expected outcomes of a test scenario. ATDD tests are intended to verify that the software system meets the expectations and the needs of the users and the stakeholders, as well as to provide feedback and guidance for the developers and the testers. Based on the given information, the ATDD test that is most likely to be written first is the one that corresponds to option B, which is:

Given the logged-in user is on the Change Password page When the user enters the correct current password, enters a valid new password (different from the last 5 passwords), and presses the Change Password button Then the user receives an email confirming that the password has been successfully changed

This ATDD test is most likely to be written first, because it covers the main functionality and the happy path of the user story, as well as the most important acceptance criterion [a]. It also verifies that the user can change the password with a valid new password that meets the criteria related to the length, the type of characters, and the history of the passwords, as specified in the acceptance criterion [c]. The other options are not likely to be written first, because they either cover less critical or less frequent scenarios, such as entering a wrong current password [d] or an invalid new password [e], or they are not related to the user story or the acceptance criteria at all, such as submitting a purchase order [d]. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles1

? ISTQB® Glossary of Testing Terms v4.0, Acceptance Test-Driven Development, User Story, Acceptance Criterion, Given-When-Then2

NEW QUESTION 4

Which of the following statements is an example of testing contributing to higher quality?

- A. A test leader writes a test summary report
- B. A project manager asks to a test leader to estimate the test effort

- C. A tester installs a test ten in the lest environment
- D. A tester finds a bug which is resolved prior to release

Answer: D

Explanation:

? The question is about identifying an example of testing contributing to higher quality. Quality is the degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations¹. Testing is the process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects².

? Therefore, testing contributes to higher quality by verifying and validating that the software products and related work products meet the specified requirements, are fit for purpose and have no defects, or at least have a reduced number of defects. Testing also provides information about the quality of the software products and related work products to the stakeholders, who can make informed decisions based on the test results³.

? Out of the four given statements, only option D is an example of testing contributing to higher quality, as it shows that testing has detected a defect (a flaw in a component or system that can cause the component or system to fail to perform its required function⁴) and that the defect has been resolved (fixed and confirmed) prior to release (delivery of the software product to the customer or end user). This means that testing has prevented a potential failure (an event in which a component or system does not perform a required function within specified limits) from occurring in the operational environment, and thus has improved the quality of the software product.

? Option A is not an example of testing contributing to higher quality, as it is a reporting activity that summarizes the test results and evaluates the test objectives, but does not directly affect the quality of the software product or related work products. A test summary report is a document that records and communicates the outcomes of testing activities, including test completion criteria, test results, incident reports, test summary and evaluation, and lessons learned.

? Option B is not an example of testing contributing to higher quality, as it is a planning activity that estimates the resources and time needed for testing activities, but does not directly affect the quality of the software product or related work products. A test effort estimate is an approximation of the amount of work and/or the duration of time required to perform testing activities.

? Option C is not an example of testing contributing to higher quality, as it is a preparation activity that sets up the test environment (an environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test), but does not directly affect the quality of the software product or related work products. A test environment installation is a process of installing and configuring the test environment according to the test environment specification.

References:

- ? 1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 10
- ? 2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 11
- ? 3: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 12
- ? 4: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 13
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 13
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 77
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 78
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 79
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 80
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 81
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 83
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 84
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 85
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 86
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 87
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 88
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 89
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 90
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 91
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 92
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 93
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 94
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 95
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 96
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 97
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 98
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 99
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 100
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 101
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 102
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 103
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 104
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 105
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 106
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 107

NEW QUESTION 5

Atypical generic skill required for the role of tester is the ability to

- A. Take on the role of developer to meet challenging project deadlines
- B. Assume leadership aimed at imposing decisions on the rest of the team.
- C. Use tools to make the execution of repetitive testing tasks more efficient.
- D. Determine the corrective actions to get a test project on track in case of deviations from the test plan

Answer: C

Explanation:

A key skill for testers is the ability to use various tools to automate repetitive tasks, enhancing the efficiency and effectiveness of testing processes. This includes tools for test execution, test management, and defect tracking. The ISTQB CTFL Syllabus v4.0 emphasizes the importance of using tools to improve productivity and reduce manual effort in repetitive testing tasks, making this a critical skill for testers.

NEW QUESTION 6

Which of the following statements about branch coverage is true?

- A. The minimum number of test cases needed to achieve full branch coverage, is usually lower than that needed to achieve full statement coverage
- B. If full branch coverage has been achieved, then all unconditional branches within the code have surely been exercised
- C. If full branch coverage has been achieved, then all combinations of conditions in a decision table have surely been exercised
- D. Exercising at least one of the decision outcomes for all decisions within the code, ensures achieving full branch coverage

Answer: D

Explanation:

Exercising at least one of the decision outcomes for all decisions within the code, ensures achieving full branch coverage, which is a test coverage criterion that requires that all branches in the control flow of the code are executed at least once by the test cases. A branch is a basic block of code that has a single entry point and a single exit point, and a decision is a point in the code where the control flow can take more than one direction, such as an if-then-else statement, a switch-case statement, a loop statement, etc. The decision outcomes are the possible paths that can be taken from a decision, such as the then branch or the else branch, the case branch or the default branch, the loop body or the loop exit, etc. The other statements are false, because:

? The minimum number of test cases needed to achieve full branch coverage, is usually higher than that needed to achieve full statement coverage, which is a test coverage criterion that requires that all executable statements in the code are executed at least once by the test cases. This is because branch coverage is a stronger criterion than statement coverage, as it implies statement coverage, but not vice versa. For example, a single test case can achieve full statement coverage for an if-then-else statement, but two test cases are needed to achieve full branch coverage, as both the then branch and the else branch need to be exercised.

? If full branch coverage has been achieved, then all unconditional branches within the code have not necessarily been exercised, as unconditional branches are branches that do not depend on any decision, and are always executed, such as a goto statement, a break statement, a return statement, etc. Unconditional branches are not part of the branch coverage criterion, as they do not represent different paths in the control flow of the code. However, they are part of the statement coverage criterion, as they are executable statements in the code.

? If full branch coverage has been achieved, then all combinations of conditions in a decision table have not necessarily been exercised, as a decision table is a test design technique that represents the logical relationships between multiple conditions and their corresponding actions, in a tabular format. A decision table can have more combinations of conditions than the number of decision outcomes in the code, as each condition can have two or more possible values, such as true or false, yes or no, etc. For example, a decision table with four conditions can have 16 combinations of conditions, but the corresponding code may have only two decision outcomes, such as pass or fail. To exercise all combinations of conditions in a decision table, a stronger test coverage criterion is needed, such as condition combination coverage, which requires that all possible combinations of condition outcomes in the code are executed at least once by the test cases.

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

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.3.1, Test Coverage Criteria Based on the Structure of the Software

? ISTQB® Glossary of Testing Terms v4.0, Branch Coverage, Statement Coverage, Branch, Decision, Decision Outcome, Unconditional Branch, Decision Table, Condition Combination Coverage

NEW QUESTION 7

Which of the following statements about the typical activities of a formal review process is TRUE?

- A. Individual review is only mandatory when the size of the work product under review is too large to cover at the review meeting
- B. Various review techniques that may be applied by participants during individual review are described in the ISO/IEC/IEEE 29119-3 standard.
- C. Choosing which standards to follow during the review process is usually made during review planning.
- D. One of the main goals of the review meeting is to make sure that all participants are aware of their roles and responsibilities in the review process

Answer: C

Explanation:

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 8

Testing Quadrants, as a model, is effective in aligning stakeholders within Agile teams. Which of the following examples demonstrates this?

- A. Using Testing Quadrants, the test manager is able to measure and communicate test coverage to all stakeholders.
- B. Using Testing Quadrants, the test manager is able to communicate potential product risk to all stakeholders.
- C. Using Testing Quadrant, the test manager is able to prioritize defects by linking these to a specific type of test.
- D. Using Testing Quadrants, the test manager is able to differentiate and describe the types of tests to all stakeholders.

Answer: D

Explanation:

The Testing Quadrants model helps Agile teams by categorizing different types of tests and their purposes. This differentiation helps test managers explain the testing strategy to all stakeholders, ensuring everyone understands the scope and objectives of each test type. This model aids in planning, executing, and tracking testing activities across different quadrants, making it easier to align with stakeholders' expectations and project goals. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.7

NEW QUESTION 9

In a two-hour uninterrupted test session, performed as part of an iteration on an Agile project, a heuristic checklist was used to help the tester focus on some specific usability issues of a web application.

The unscripted tests produced by the tester's experience during such session belong to which one of the following testing quadrants?

- A. Q1
- B. Q2
- C. Q3
- D. Q4

Answer: C

Explanation:

The unscripted tests produced by the tester's experience during the two-hour test session belong to the testing quadrant Q3. The testing quadrants are a classification of testing types based on two dimensions: the test objectives (whether the testing is focused on supporting the team or critiquing the product) and the test basis (whether the testing is based on the technology or the business). The testing quadrants are labeled as Q1, Q2, Q3, and Q4, and each quadrant represents a different testing perspective, such as unit testing, acceptance testing, usability testing, or performance testing. The testing quadrant Q3 corresponds to the testing types that have the objective of critiquing the product from the business perspective, such as exploratory testing, usability testing, user acceptance testing, alpha testing, beta testing, etc. The unscripted tests performed by the tester in the given scenario are examples of exploratory testing and usability testing, as they are based on the tester's experience, intuition, and learning of the web application, and they focus on some specific usability issues, such as the user interface, the user satisfaction, the user feedback, etc. The other options are incorrect, because:

? The testing quadrant Q1 corresponds to the testing types that have the objective of supporting the team from the technology perspective, such as unit testing, component testing, integration testing, system testing, etc. These testing types are usually performed by developers or testers who have access to the source code, the design, the architecture, or the configuration of the software system, and they aim to verify the functionality, the quality, and the reliability of the software system at different levels of integration.

? The testing quadrant Q2 corresponds to the testing types that have the objective of supporting the team from the business perspective, such as functional testing, acceptance testing, story testing, scenario testing, etc. These testing types are usually performed by testers or customers who have access to the requirements, the specifications, the user stories, or the business processes of the software system, and they aim to validate that the software system meets the expectations and the needs of the users and the stakeholders.

? The testing quadrant Q4 corresponds to the testing types that have the objective of critiquing the product from the technology perspective, such as performance testing, security testing, reliability testing, compatibility testing, etc. These testing types are usually performed by testers or specialists who have access to the tools, the metrics, the standards, or the benchmarks of the software system, and they aim to evaluate the non-functional aspects of the software system, such as the efficiency, the security, the reliability, or the compatibility of the software system under different conditions or environments. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles

? ISTQB® Glossary of Testing Terms v4.0, Testing Quadrant, Exploratory Testing, Usability Testing, Unit Testing, Component Testing, Integration Testing, System Testing, Functional Testing, Acceptance Testing, Story Testing, Scenario Testing, Performance Testing, Security Testing, Reliability Testing, Compatibility Testing

NEW QUESTION 10

Which of the following best describes the way in which statement coverage is measured?

- A. Measured as the number of decision outcomes executed by the tests, divided by the total number of decision outcomes in the test object.
- B. It is not possible to accurately measure statement coverage.
- C. Measured as the number of statements executed by the tests, divided by the total number of executable statements in the code.
- D. Measured as the number of lines of code executed by the test, divided by the total number of lines of code in the test object.

Answer: C

Explanation:

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 10

Which of the following is not an example of a typical generic skill required for testing?

- A. Be able to apply test-driven development
- B. Be able to use test management tools and defect tracking tools
- C. Be able to communicate defects and failures to developers as objectively as possible
- D. Possess the necessary social skills that support effective teamwork

Answer: A

Explanation:

Test-driven development is not an example of a typical generic skill required for testing, but rather an example of a specific technical skill or a development practice that may or may not be relevant for testing, depending on the context and the objectives of the testing activities. Test-driven development 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 skill that is generally expected or needed for testers, especially for testers who are not involved in unit testing or who do not have access to the source code. The other options are examples of typical generic skills required for testing, which are skills that are applicable and beneficial for testing in any context or situation, regardless of the specific testing techniques, tools, or methods used. The typical generic skills required for testing include:

? Be able to use test management tools and defect tracking tools: These are tools that help testers to plan, organize, monitor, and control the testing activities and resources, as well as to record, track, analyze, and resolve the defects detected during testing. These tools can improve the efficiency, the effectiveness, and the communication of the testing process, as well as to provide traceability, metrics, and reports for the testing outcomes.

? Be able to communicate defects and failures to developers as objectively as possible: This is a skill that involves the ability to report and describe the defects and failures found during testing in a clear, concise, accurate, and unbiased manner, using relevant information, evidence, and terminology, without making assumptions, judgments, or accusations. This skill can facilitate the collaboration, the understanding, and the resolution of the defects and failures between the testers and the developers, as well as to prevent conflicts, misunderstandings, or blame games.

? Possess the necessary social skills that support effective teamwork: These are skills that involve the ability to interact, cooperate, and coordinate with other 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. These skills can include communication, negotiation, leadership, motivation, feedback, conflict resolution, etc. These skills can enhance the quality, the productivity, and the satisfaction of the testing process, as well as to foster a positive and constructive testing culture. 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 Lifecycle

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches

? ISTQB® Glossary of Testing Terms v4.0, Test-driven Development, Test Management Tool, Defect Tracking Tool, Defect Report, Failure, Social Skill2

NEW QUESTION 14

Which of the following is a test-first approach, where tests that express a shared understanding from stakeholders of how the application is expected to work, are first written in business-readable language (following the Given/When/Then format), and then made executable to drive development?

- A. Test-Driven Development (TDD)
- B. Acceptance Test-Driven Development (ATDD)
- C. Behavior-Driven Development (BDD)
- D. Domain-Driven Design (DDD)

Answer: C

Explanation:

This answer is correct because Behavior-Driven Development (BDD) is a test-first approach, where tests that express a shared understanding from stakeholders of how the application is expected to work, are first written in business-readable language (following the Given/When/Then format), and then made executable to drive development. BDD is a collaborative approach that involves testers, developers, business analysts, product owners, and other stakeholders in defining the expected behavior of the application using scenarios that describe the preconditions, actions, and outcomes of the application. BDD scenarios are written using a domain-specific language (DSL) that can be translated into executable test cases using tools such as Cucumber or SpecFlow. BDD aims to improve communication, collaboration, and feedback among the team members, and to deliver software that meets the customer's needs and expectations. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.4

NEW QUESTION 15

Which of the following statements is true?

- A. Experience-based test techniques rely on the experience of testers to identify the root causes of defects found by black-box test techniques
- B. Some of the most common test basis used by white-box test techniques include user stories, use cases and business processes
- C. Experience-based test techniques are often useful to detect hidden defects that have not been targeted by black-box test techniques
- D. The primary goal of experience-based test techniques is to design test cases that can be easily automated using a GUI-based test automation tool

Answer: C

Explanation:

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 of defects based on their own insights, heuristics, or exploratory testing. The root causes of defects are usually identified by debugging or root cause analysis, which are activities that involve examining the code or the development process to find and fix the errors that led to the defects.

? Some of the most common test basis used by white-box test techniques include

the source code, the design documents, the architecture diagrams, and the control flow graphs of the software system. White-box test techniques are test design techniques that use the internal structure and implementation of the software system as the test basis, and aim to achieve a certain level of test coverage based on the code elements, such as statements, branches, paths, etc. User stories, use cases, and business processes are examples of test basis used by black-box test techniques, as they describe the functional and non-functional requirements of the software system from the perspective of the users or the stakeholders.

? The primary goal of experience-based test techniques is not to design test cases

that can be easily automated using a GUI-based test automation tool, but rather to design test cases that can reveal defects that are not easily detected by other test techniques, such as boundary value analysis, equivalence partitioning, state transition testing, etc. Test automation is the use of software tools to execute test cases and compare actual results with expected results, without human intervention. Test automation can be applied to different types of test techniques, depending on the test objectives, the test levels, the test tools, and the test resources. However, test automation is not always feasible or beneficial, especially for test cases that require human judgment, creativity, or exploration, such as

those designed by experience-based test techniques. 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 Techniques

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.2, White-box Test Design Techniques

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.3, Experience-based Test Design Techniques

? ISTQB® Glossary of Testing Terms v4.0, Experience-based Test Technique, Black-box Test Technique, White-box Test Technique, Test Basis, Test Coverage, Test Automation

NEW QUESTION 20

Which of the following statements about statement coverage is TRUE?

- A. Achieving 90% statement coverage ensures that 90% branch coverage is achieved.
- B. Achieving 100% statement coverage ensures that no variable within the code has been used without being initialised.
- C. Achieving 100% statement coverage ensures that 100% branch coverage is achieved
- D. Achieving 80% statement coverage ensures that 80% of all executable statements within the code have been exercised.

Answer: D

Explanation:

Statement coverage measures the percentage of executable statements that have been exercised by a test suite. Achieving 80% statement coverage means that 80% of the executable code lines have been tested. This metric helps in understanding how much of the code has been covered during testing. However, it does not guarantee branch coverage, variable initialization, or detection of all possible defects. The ISTQB CTFL Syllabus v4.0 explains statement coverage as a measure of the extent to which the code has been tested, without implying other types of coverage or testing goals.

NEW QUESTION 23

Which of the following is a typical potential risk of using test automation tools?

- A. Reduced feedback times regarding software quality compared to manual testing.
- B. Reduced test execution times compared to manual testing.
- C. Reduced repeatability and consistency of tests compared to manual testing
- D. Underestimation of effort required to maintain test scripts.

Answer: D

Explanation:

One of the common risks associated with test automation tools is the underestimation of the effort required to maintain test scripts. Test scripts can become outdated or broken due to changes in the application, requiring significant effort to update and maintain them. This risk is highlighted in the ISTQB CTFL syllabus under the discussion of the benefits and risks of test automation.

References:ISTQB CTFL Syllabus, Section on test tools and automation.

NEW QUESTION 28

Determining the schedule for each testing activity and test milestones for a test project, using activity estimates, available resources, and other constraints is a typical task performed during

- A. Test execution
- B. Test design.
- C. Test analysis.
- D. Test planning

Answer: D

Explanation:

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 31

A financial institution is to implement a system that calculates the interest rates paid on investment accounts based on the sum invested. You are responsible for testing the system and decide to use equivalence partitioning and boundary value analysis to design test cases. The requirements describe the following expectations:

Investment range| Interest rate
R500 to R1000| 10%

R1001 to R5000| 11%
R5001 to R10000| 12%
R10001 to R50000| 13%

What is the minimum number of test cases required to cover all valid equivalence partitions for calculating the interest?

- A. 5
- B. 4
- C. 8
- D. 16

Answer: B

Explanation:

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.

NEW QUESTION 35

Which of the following statements about estimation of the test effort is WRONG?

- A. Once the test effort is estimated, resources can be identified and a schedule can be drawn up.
- B. Effort estimate can be inaccurate because the quality of the product under tests is not known.
- C. Effort estimate depends on the budget of the project.
- D. Experience based estimation is one of the estimation techniques.

Answer: C

Explanation:

? 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: 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 39

A document describes the test procedures that have been derived for the identified test sets. Among other things, the order in which the test cases in the corresponding test set are to be executed according to the dependencies described by preconditions and postconditions is specified. This document is a typical work product produced as part of:

- A. Test design.
- B. Test analysis
- C. Test Implementation.
- D. Test monitoring and control

Answer: C

Explanation:

Test implementation involves finalizing the test procedures, including the order of execution of test cases based on their dependencies, preconditions, and postconditions. This phase ensures that all necessary test scripts, test data, and test environments are ready for execution. According to the ISTQB CTFL Syllabus v4.0, test implementation is the phase where detailed test procedures are derived and documented, making it a critical step before actual test execution.

NEW QUESTION 42

Which of the following statements about white-box test techniques is true?

- A. Achieving full statement coverage and full branch coverage for a software product means that such software product has been fully tested and there are no remaining bugs within the code
- B. Code-related white-box test techniques are not required to measure the actual code coverage achieved by black-box testing, as code coverage can be measured using the coverage criteria associated with black-box test techniques
- C. Branch coverage is the most thorough code-related white-box test technique, and therefore applicable standards prescribe achieving full branch coverage at the highest safety levels for safety-critical systems
- D. Code-related white-box test techniques provide an objective measure of coverage and can be used to complement black-box test techniques to increase confidence in the code

Answer: D

Explanation:

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 43

Given the following User Story: "As an online customer, I would like to be able to cancel the purchase of an individual item from a shopping list so that it only displays the relevant items, in less than 1 second", which of the following can be considered as applicable acceptance test cases?

- A. 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.i
- B. Click on my online shopping list, select all the items, delete all the items, the unwanted items are deleted from the shopping list in less than 1 second.ii
- C. 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.I
- D. Click on the checkout button, select the payment method, make payment, confirmation received of payment and shipping date.
- E. Click on my shopping list, select the unwanted item, delete the unwanted item, the unwanted item is deleted from the shopping list.Select the correct Answer
- F. I, ii and v
- G. iv
- H. i and iii
- I. v

Answer: C

Explanation:

Reference:ISTQB CTFL Syllabus V4.0, Section 5.2.2

NEW QUESTION 48

Which of the following is a factor that contributes to a successful review?

- A. All participants in the review are aware they will be evaluated based on the defects they will find
- B. The author of the work product to be reviewed leads the review meeting.
- C. All participants in the review are trained to deal with the review type and its objectives.
- D. Review metrics must be collected to improve the review process

Answer: C

Explanation:

A successful review process involves all participants being trained in the review type and understanding its objectives. This ensures that everyone can contribute effectively and understand what is expected from the review. Proper training helps to identify defects accurately and facilitates constructive feedback, leading to a more efficient and effective review process. Hence, statement C is correct according to the ISTQB CTFL syllabus.

NEW QUESTION 52

Which of the following is a test task that usually occurs during test implementation?

- A. Make sure the planned test environment is ready to be delivered
- B. Find, analyze, and remove the causes of the failures highlighted by the tests
- C. Archive the testware for use in future test projects
- D. Gather the metrics that are used to guide the test project

Answer: A

Explanation:

A test task that usually occurs during test implementation is to make sure the planned test environment is ready to be delivered. The test environment is the hardware and software configuration on which the tests are executed, and it should be as close as possible to the production environment where the software system will operate. The test environment should be planned, prepared, and verified before the test execution, to ensure that the test conditions, the test data, the test tools, and the test interfaces are available and functional. The other options are not test tasks that usually occur during test implementation, but rather test tasks that occur during other test activities, such as:

? Find, analyze, and remove the causes of the failures highlighted by the tests: This is a test task that usually occurs during test analysis and design, which is the activity of analyzing the test basis, designing the test cases, and identifying the test data. During this activity, the testers can use techniques such as root cause analysis, defect prevention, or defect analysis, to find, analyze, and remove the causes of the failures highlighted by the previous tests, and to prevent or reduce the occurrence of similar failures in the future tests.

? Archive the testware for use in future test projects: This is a test task that usually occurs during test closure, which is the activity of finalizing and reporting the test results, evaluating the test process, and identifying the test improvement actions. During this activity, the testers can archive the testware, which are the test artifacts produced during the testing process, such as the test plan, the test cases, the test data, the test results, the defect reports, etc., for use in future test projects, such as regression testing, maintenance testing, or reuse testing.

? Gather the metrics that are used to guide the test project: This is a test task that usually occurs during test monitoring and control, which is the activity of tracking and reviewing the test progress, status, and quality, and taking corrective actions when necessary. During this activity, the testers can gather the metrics, which are the measurements of the testing process, such as the test coverage, the defect density, the test effort, the test duration, etc., that are used to guide the test project, such as planning, estimating, scheduling, reporting, or improving the testing process. 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® 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, Test Environment, Test Condition, Test Data, Test Tool, Test Interface, Failure, Root Cause Analysis, Defect Prevention, Defect Analysis, Testware, Regression Testing, Maintenance Testing, Reuse Testing, Test Coverage, Defect Density, Test Effort, Test Duration2

NEW QUESTION 53

In addition to thorough testing of the requirements specification, a development team aims to involve users as early as possible in the development process, using practices such as prototyping, to ensure that the software systems being developed will meet the users' expectations. This approach is especially useful at mitigating the risks associated with one of the seven testing principles, which one?

- A. Tests wear out
- B. Absence-of-errors fallacy
- C. Working software over comprehensive documentation.
- D. Defects cluster together

Answer: B

Explanation:

The absence-of-errors fallacy is the mistaken belief that just because a software system is free of defects, it will meet the user's needs and expectations. Involving users early through practices like prototyping helps ensure that the development team is building the right system that meets user expectations, not just a system that is defect-free. This approach aligns with the testing principle that emphasizes understanding the users' needs and ensuring the system fulfills them. This principle is explained in the ISTQB CTFL Syllabus v4.0.

NEW QUESTION 56

A test status report SHOULD:

- A. Specify the impediments to carrying out the planned test activities in the reporting period and the corresponding solutions put in place to remove them
- B. Be produced as part of test completion activities and report unmitigated product risks to support the decision whether or not to release the product
- C. Always be based on the same template within an organisation, as its structure and contents should not be affected by the audience to which the report is presented.
- D. Specify the lines of communication between testing, other lifecycle activities, and within the organisation that were chosen at the outset of the test project.

Answer: A

Explanation:

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 61

What is test oracle?

- A. The source of test objectives
- B. The source for the actual results
- C. The source of expected results
- D. The source of input conditions

Answer: C

Explanation:

A test oracle is a mechanism or principle that can be used to determine whether the observed behavior or output of a system under test is correct or not1. A test oracle can be based on various sources of expected results, such as specifications, user expectations, previous versions, comparable systems, etc2. References: ISTQB Certified Tester Foundation Level(CTFL) v4.0 Syllabus, Section 1.2.1, Page 91; ISTQB Glossary of Testing Terms, Version 4.0, Page 332.

NEW QUESTION 64

Which of the following best describes the relationship between a test progress report and a test summary report?

- A. The test report prepared during a test activity may be referred to as a test progress report, while a test report prepared at the end of a test activity may be referred to as a test summary report.
- B. The test report prepared during a test activity may be referred to as a test summary report, while a test report prepared at the end of a test activity may be referred to as a test progress report.
- C. There is no difference between a test progress report and a test summary report.
- D. Both the test progress report and the test summary report should always be generated via an automated tool.

Answer: A

Explanation:

Reference:ISTQB CTFL Syllabus V4.0, Section 5.3.2

NEW QUESTION 67

Consider a review for a high-level architectural document written by a software architect. The architect does most of the review preparation work, including distributing the document to reviewers before the review meeting. However, reviewers are not required to analyze the document in advance, and during the review meeting the software architect explains the document step by step. The only goal of this review is to establish a common understanding of the software architecture that will be used in a software development project.

Which of the following review types does this review refer to?

- A. Inspection
- B. Audit
- C. Walkthrough
- D. Informal review

Answer: C

Explanation:

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 68

Which of the following are the phases of the ISTQB fundamental test process?

- A. Test planning and control, Test analysis and design, Test implementation and execution, Evaluating exit criteria and reporting
- B. Test closure activities
- C. Test planning, Test analysis and design
- D. Test implementation and control
- E. Checking test coverage and reporting, Test closure activities
- F. Test planning and control, Test specification and design
- G. Test implementation and execution, Evaluating test coverage and reporting, Retesting and regression testing, Test closure activities
- H. Test planning
- I. Test specification and design
- J. Test implementation and execution
- K. Evaluating exit criteria and reporting
- L. Retesting and test closure activities

Answer: A

Explanation:

The ISTQB fundamental test process consists of five main phases, as described in the ISTQB Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15:

? Test planning and control: This phase involves defining the test objectives, scope, strategy, resources, schedule, risks, and metrics, as well as monitoring and controlling the test activities and results throughout the test process.

? Test analysis and design: This phase involves analyzing the test basis (such as requirements, specifications, or user stories) to identify test conditions (such as features, functions, or scenarios) that need to be tested, and designing test cases and test procedures (such as inputs, expected outcomes, and execution steps) to cover the test conditions. This phase also involves evaluating the testability of the test basis and the test items (such as software or system components), and selecting and implementing test techniques (such as equivalence partitioning, boundary value analysis, or state transition testing) to achieve the test objectives and optimize the test coverage and efficiency.

? Test implementation and execution: This phase involves preparing the test environment (such as hardware, software, data, or tools) and testware (such as test cases, test procedures, test data, or test scripts) for test execution, and executing the test procedures or scripts according to the test plan and schedule. This phase also involves logging the outcome of test execution, comparing the actual results with the expected results, and reporting any discrepancies as incidents (such as defects, errors, or failures).

? Evaluating exit criteria and reporting: This phase involves checking if the planned test activities have been completed and the exit criteria (such as quality, coverage, or risk levels) have been met, and reporting the test results and outcomes to the stakeholders. This phase also involves making recommendations for the release or acceptance decision based on the test results and outcomes, and identifying any residual risks (such as known defects or untested areas) that need to be addressed or mitigated.

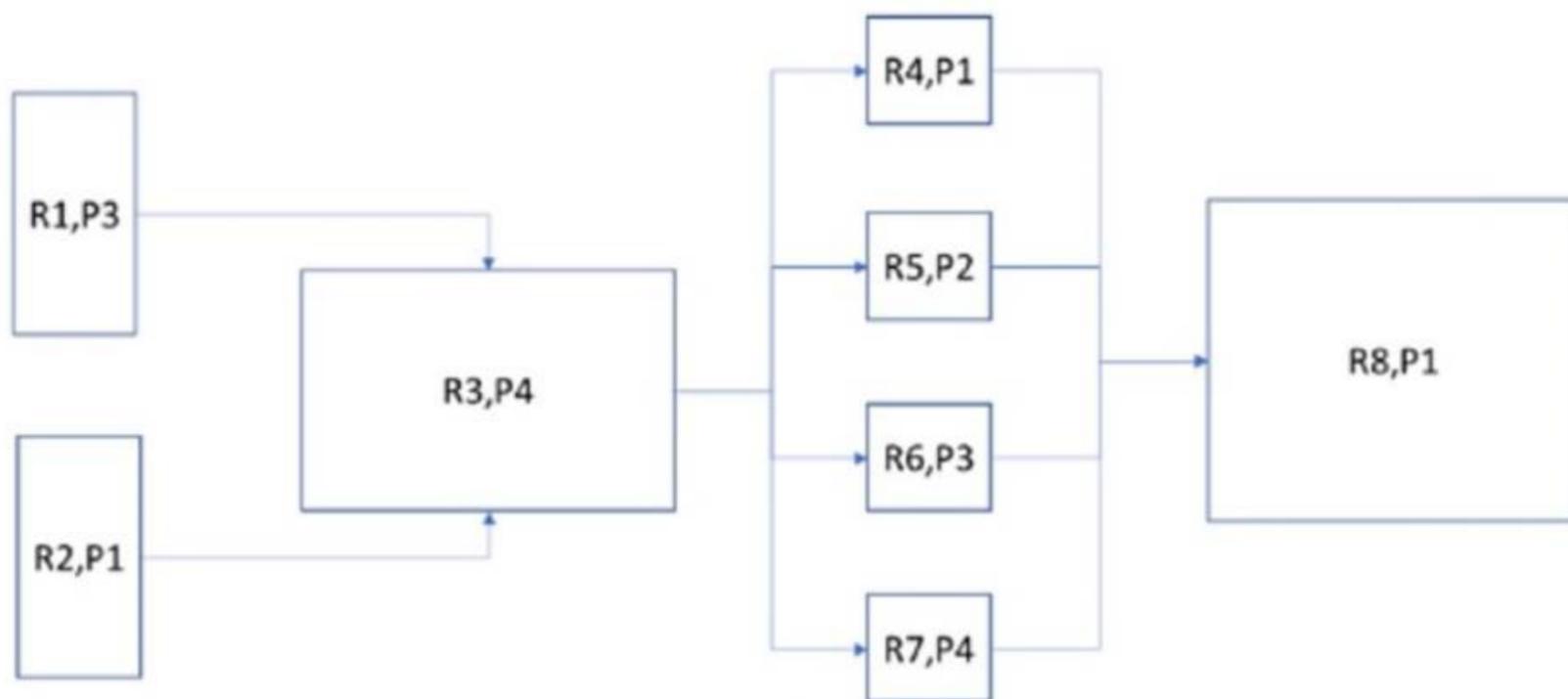
? Test closure activities: This phase involves finalizing and archiving the testware and test environment for future reuse, and evaluating the test process and the test project against the test objectives and the test plan. This phase also involves identifying any lessons learned and best practices, and communicating the findings and suggestions for improvement to the relevant parties.

References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 37-38;

ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 88, page 32.

NEW QUESTION 71

The following diagram displays the logical dependencies between requirements and the individual requirement priorities. For example, "R2->R3" means that R3 is dependent on R2. Priority is indicated by the number next to the letter ??P" i.e. P1 has a higher priority than P2.



Which one of the following options best describes the test execution sequence using both requirement dependency and priority

- A. R2, R1, R3, R4, R5, R6, R7, R8.
- B. R1, R2, R3, R4, R5, R6, R7, R8.
- C. R2, R4, R8, R5, R1, R6, R3, R7.
- D. R2, R1, R3, R7, R6, R5, R4, R8.

Answer: D

Explanation:

The correct test execution sequence should consider both the dependencies between the requirements and their priorities. According to the diagram, the sequence begins with R2 (P1) as it is a prerequisite for R3 (P4). Then R1 (P3) can be tested. R3 follows as it depends on R2. Next, R7 (P4) should be tested before R6 (P3) and R5 (P2), as indicated by their dependencies. Finally, R4 (P1) and R8 (P1) can be tested. Therefore, the best sequence is R2, R1, R3, R7, R6, R5, R4, R8. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.5

NEW QUESTION 72

During iteration planning, a scrum team uses an estimation technique called planning poker to estimate the effort required to deliver a critical user story. In advance of the estimation session, the team agreed on some ground rules to limit the number of poker rounds and save time.

The team agreed on the following:

- * 1. They will use the following progression for estimation: Extra-small, Small, Medium, Large, Extra-large, and Extra-extra-large.
- * 2. If estimation values differ significantly, the highest score will be used for estimation purposes.

The result of the first round of planning poker: Team Member Estimation

Business Large Development Extra-extra-large Testing Extra-extra-large

Which of the following options best represent the team's next actions?

- A. The fact that all estimations are high indicate that the user story is not well understood or should be broken down into multiple smaller stories.
- B. The pre-agreed rules state that the highest score should be used for estimation, resulting in the user story being categorised as Extra-extra-large.
- C. Since the business representative is likely to have the most informed view of the requirement, the user story is categorised as a Large.
- D. the team discusses the differences in the estimates and repeats the poker round until an agreement is reached.

Answer: D

Explanation:

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 74

Which of the following statements is TRUE?

- A. Unlike functional testing, non-functional testing can only be applied to conventional systems, not artificial intelligence-based system.
- B. Functional testing focuses on what the system is supposed to do, while white-box testing focuses on how well the system does what it is supposed to do
- C. Functional testing can be applied to all test levels, while non-functional testing can be applied only to system and acceptance test levels.
- D. Black-box test techniques and experience-based test techniques may be applicable to both functional testing and non-functional testing

Answer: D

Explanation:

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 76

Which of the following is an advantage of the whole team approach?

- A. It helps avoid the risk of tasks associated with a user story not moving through the Agile task board at an acceptable rate during an iteration.
- B. It helps team members understand the current status of an iteration by visualising the amount of work left to do compared to the time allotted for the iteration
- C. It helps the whole team be more effective in test case design by requiring all team members to master all types of test techniques.
- D. It helps team members develop better relationships with each other and make their collaboration more effective for the benefit of the project.

Answer: D

Explanation:

The whole team approach, often advocated in Agile methodologies, emphasizes collaboration and collective responsibility among all team members. This approach enhances the relationships within the team and improves overall collaboration, which in turn benefits the project's success. According to the ISTQB CTFL Syllabus v4.0, the whole team approach fosters better communication and cooperation, leading to more effective problem-solving and higher-quality outcomes.

NEW QUESTION 79

Test automation allows you to:

- A. demonstrate the absence of defects
- B. produce tests that are less subject to human errors
- C. avoid performing exploratory testing
- D. increase test process efficiency by facilitating management of defects

Answer: B

Explanation:

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 81

Which of the following is a task the Author is responsible for, as part of a typical formal review?

- A. Determining the people who will be involved in the review
- B. Recording the anomalies found during the review meeting
- C. Identifying potential anomalies in the work product under review
- D. Fixing the anomalies found in the work product under review

Answer: C

Explanation:

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 84

Which of the following statements is true?

- A. A defect does not always produce a failure, while a bug always produces a failure
- B. A defect may cause a failure which, when occurring, always causes an error
- C. Failures can be caused by defects, but also by environmental conditions
- D. Bugs are defects found during component testing, while failures are defects found at higher test levels

Answer: C

Explanation:

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.

? Bugs are defects found during component testing, while failures are defects found at higher test levels. This statement is false, because bugs and failures are not different types of defects, but rather different terms for defects and their manifestations. As mentioned before, bugs are just another word for defects, and failures are the events in which the software system does not perform as expected due to defects. Bugs and failures can be found at any test level, not only at component testing or higher test levels. 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. Defects and failures can occur and be detected at any test level, depending on the test objectives, the test basis, the test techniques, and the test environment. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality1
 ? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles1
 ? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles1
 ? ISTQB® Glossary of Testing Terms v4.0, Failure, Defect, Bug, Environmental Condition, Error, Test Level2

NEW QUESTION 88

Consider a given test plan which, among others, contains the following three sections: "Test Scope", "Testing Communication", and "Stakeholders". The features of the test object to be tested and those excluded from the testing represent information that is:

- A. not usually included in a test plan, and therefore in the given test plan it should not be specified neither within the three sections mentioned, nor within the others
- B. 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
- C. usually included in a test plan and, in the given test plan, it is more likely to be specified within "Testing Communication" rather than in the other two sections mentioned
- D. usually included in a test plan and, in the given test plan, it is more likely to be specified within "Stakeholders" rather than in the other two sections mentioned

Answer: B

Explanation:

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 91

Which of the following types of tools is BEST suited for determining source code compliance with the guidelines provided by a coding standard?

- A. Containerisation tool
- B. Fault seeding tool.
- C. Static analysis tool.
- D. Test data preparation tool

Answer: C

Explanation:

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 95

Match each objective to the correct test level Objective:

- A) Verifying whether the functional and non-functional behaviors of the system are as designed and specified.
 - B) Verifying whether the functional and non-functional behaviors of the interfaces are as designed.
 - C) Verifying whether the functional and non-functional behaviors of the components are as designed and specified.
 - D) Establishing confidence in the quality of the system as a whole. Test Level:
- * 1. Component testing. 2. Integration testing. 3. System testing. 4. Acceptance testing.

- A. A3, B2, C4, D1
- B. A2, B3, C1, D4
- C. A3, B2, C1, D4

Answer: C

Explanation:

The test levels and their objectives can be matched as follows:

? Verifying whether the functional and non-functional behaviors of the system are as designed and specified (A3: System testing).
 ? Verifying whether the functional and non-functional behaviors of the interfaces are as designed (B2: Integration testing).
 ? Verifying whether the functional and non-functional behaviors of the components are as designed and specified (C1: Component testing).
 ? Establishing confidence in the quality of the system as a whole (D4: Acceptance testing).

NEW QUESTION 100

Which of the following statements about the testing quadrants is TRUE?

- A. The higher the number of the testing quadrant, the more important the tests associated with this quadrant are
- B. Automated acceptance tests produced during BDD and ATDD are classified in quadrant Q2.
- C. Exploratory tests are classified in quadrant Q3, and they are usually included in a continuous integration process.
- D. Automated unit tests produced during TDD are classified in quadrant Q4 as they are technology facing.

Answer: B

Explanation:

The correct statement is B. According to the ISTQB CTFL syllabus, the testing quadrants help to categorize tests based on their purpose and whether they are technology-facing or business-facing, and whether they support the team or critique the product. Quadrant Q2 includes tests that are business-facing and support the team, such as automated acceptance tests produced during Behavior-Driven Development (BDD) and Acceptance Test-Driven Development (ATDD) . Quadrant Q3 includes business-facing tests that critique the product, such as exploratory testing, usability testing, and user acceptance testing. These tests are typically manual and focus on evaluating the product from a user perspective, rather than being part of a continuous integration process. Quadrant Q4 includes technology-facing tests that critique the product, such as performance tests, security tests, and other non-functional tests, which can be automated but are not related to unit tests produced during TDD .

NEW QUESTION 102

A company wants to reward each of its salespeople with an annual bonus that represents the sum of all the bonuses accumulated for every single sale made by that salesperson. The bonus for a single sale can take on the following four values: 3%, 5%, 7% and 10% (the percentage refers to the amount of the single sale). These values are determined on the basis of the type of customer (classified as "Basic" or "Premium") to which such sale was made, and on the amount of such sale classified into the following three groups G1, G2 and G3:

- [G1]: less than 300 euros
- [G2]: between 300 and 2000 euros
- [G3]: greater than 2000 euros

Which of the following is the minimum number of test cases needed to cover the full decision table associated with this scenario?

- A. 12
- B. 6
- C. 4
- D. 3

Answer: B

Explanation:

The minimum number of test cases needed to cover the full decision table associated with this scenario is 6. This is because the decision table has 4 conditions (type of customer and amount of sale) and 4 actions (bonus percentage). The conditions have 2 possible values each (Basic or Premium, and G1, G2 or G3), so the total number of combinations is $2 \times 2 \times 2 \times 2 = 16$. However, not all combinations are valid, as some of them are contradictory or impossible. For example, a sale cannot be both less than 300 euros and greater than 2000 euros at the same time. Therefore, we need to eliminate the invalid combinations and keep only the valid ones. The valid combinations are:

Type of customer	Amount of sale	Bonus percentage
Basic	G1	3%
Basic	G2	5%
Basic	G3	7%
Premium	G1	5%
Premium	G2	7%
Premium	G3	10%

These 6 combinations cover all the possible values of the conditions and actions, and they are the minimum number of test cases needed to cover the full decision table. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents,

NEW QUESTION 107

You are performing the role of tester on an Agile project. Which of the following tasks would be your responsibility?

- A. Understanding, implementing, and updating the test strategy.
 - B. Ensuring the proper use of testing tools.
 - C. Coaching other team members in the relevant aspects of testing.
 - D. Actively collaborating with developers and business stakeholders to clarify requirements, especially in terms of testability, consistency, and completeness.
 - E. Participating proactively in team retrospective meeting, suggesting and implementing improvements.
- Select the correct Answer:
- F. i, iv and v
 - G. i, ii and iii
 - H. i, iii and v
 - I. ii
 - J. iv and v

Answer: A

Explanation:

In an Agile project, a tester's responsibilities include understanding, implementing, and updating the test strategy (i), actively collaborating with developers and business stakeholders to clarify requirements, especially in terms of testability, consistency, and completeness (iv), and participating proactively in team retrospective meetings, suggesting and implementing improvements (v). These activities ensure that testing is integrated into the development process, promoting continuous feedback and improvement. The ISTQB CTFL syllabus underlines the collaborative nature of Agile testing and the tester's role in contributing to the team's overall quality goals.

References:ISTQB CTFL Syllabus, Section on Agile Testing Practices.

NEW QUESTION 110

Which of the following statements refers to good testing practice to be applied regardless of the chosen software development model?

- A. Tests should be written in executable format before the code is written and should act as executable specifications that drive coding

- B. Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level
- C. Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly
- D. Involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle

Answer: D

Explanation:

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 115

A requirement specifies that if the total amount of sales (TAS) made during the year by a corporate seller is 300,000€ or more, the bonus that must be paid to the seller is 100% of a certain amount agreed upon at the beginning of the year. The software contains a fault as it implements this requirement with the decision "IF (TAS = 300,000)" instead of "IF (TAS >= 300,000)". The application of the 3-value boundary value analysis to this problem consists of the following three test cases (TAS is an integer variable):

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

Which of the following statements is TRUE?

- A. TC1 would highlight the fault
- B. TC2 would highlight the fault
- C. TC3 would highlight the fault
- D. None of the three test cases would highlight the fault.

Answer: B

Explanation:

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 119

A new web app aims at offering a rich user experience. As a functional tester, you have run some functional tests to verify that, before releasing the app, such app works correctly on several mobile devices, all of which are listed as supported devices within the requirements specification. These tests were performed on stable and isolated test environments where you were the only user interacting with the application. All tests passed, but in some of those tests you observed the following issue: on some mobile devices only, the response time for two web pages containing images was extremely slow.

Based only on the given information, which of the following recommendation would you follow?

- A. You should open a defect report providing detailed information on which devices and by running which tests you observed the issue
- B. The issue is related to performance efficiency, not functionalit
- C. Thus, as a functional tester, you should not open any defect report as all the functional tests passed
- D. You should not open any defect report as the problem is most likely due to poor hardware equipment on the devices where you observed the issue
- E. You should not open any defect report and inform the test manager that the devices on which you observed the issue should no longer be supported so that they will be removed from the requirements specification

Answer: A

Explanation:

As a functional tester, you should open a defect report providing detailed information on which devices and by running which tests you observed the issue. A defect report is a document that records the occurrence, nature, and status of a defect detected during testing, and provides information for further investigation and resolution. A defect report should include relevant information such as the defect summary, the defectdescription, the defect severity, the defect priority, the defect status, the defect origin, the defect category, the defect reproduction steps, the defect screenshots, the defect attachments, etc. Opening a defect report is a good practice for any tester who finds a defect in the software system, regardless of the type or level of testing performed. The other options are not recommended, because:
? The issue is related to performance efficiency, not functionality, but that does not mean that as a functional tester, you should not open any defect report as all the functional tests passed. Performance efficiency is a quality characteristic that measures how well the software system performs its functions under stated conditions, such as the response time, the resource utilization, the throughput, etc. Performance efficiency is an important aspect of the user experience, especially for web applications that run on different devices and networks. Even if the functional tests passed, meaning that the software system met the functional requirements, the performance issue observed on some devices could still affect the user satisfaction, the usability, the reliability, and the security of the software system. Therefore, as a functional tester, you have the responsibility to report the performance issue as a defect, and provide as much information as possible to help the developers or the performance testers to investigate and resolve it.

NEW QUESTION 124

A company runs a pilot project for evaluation of a test automation tool. Which of the following is NOT a valid object of this pilot project?

- A. Get familiar with the functionality and options of the tool
- B. Check haw the tool fits to the existing test processes
- C. Train all testers on using the tool
- D. Decide upon standards for tool implementation

Answer: C

Explanation:

? A pilot project is a small-scale experiment or trial that is conducted to evaluate the feasibility, effectiveness, and suitability of a test automation tool before implementing it on a larger scale1.

? The objectives of a pilot project may vary depending on the context and scope of the test automation initiative, but some common ones are2:

? Therefore, option C is not a valid objective of a pilot project, as it is not necessary to train all testers on using the tool at this stage. Training all testers on using the tool would be more appropriate after the tool has been selected and approved for full-scale implementation, and after the standards and guidelines have been established. Training all testers on using the tool during the pilot project would be inefficient, costly, and premature, as the tool may not be suitable or effective for the intended purpose, or may be replaced by another tool later.

References:

? 1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82

? 2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 83

? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 84

? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 85

NEW QUESTION 126

In which one of the following test techniques are test cases derived from the analysis of the software architecture?

- A. Black-box test techniques.
- B. Experience-based test techniques.
- C. Checklist-based test techniques.
- D. White-box test techniques.

Answer: D

Explanation:

White-box test techniques are test design techniques where the test cases are derived from the internal structure of the software, including its architecture, code, and logical flow. These techniques involve the tester having knowledge of the internal workings of the software to create test cases that ensure all possible paths and conditions are tested. This is in contrast to black-box test techniques, which focus on input-output behavior without considering the internal structure.Reference:ISTQB CTFL Syllabus V4.0, Section 4.3

NEW QUESTION 129

Which one of the following statements relating to the benefits of static testing is NOT correct?

- A. Static testing enables early detection of defects before dynamic testing is performed.

- B. Static testing reduces testing costs and time.
- C. Static testing increases development costs and time.
- D. Static testing identifies defects which are not easily found by dynamic testing.

Answer: C

Explanation:

The statement that "static testing increases development costs and time" is NOT correct. Static testing actually helps to reduce development costs and time by identifying defects early in the development process before dynamic testing is performed. Early detection of defects reduces the cost and effort required to fix them and prevents the propagation of defects to later stages, thus reducing overall testing and development costs. References:ISTQB CTFL Syllabus, Section 3.1.2, "The Value of Static Testing."

NEW QUESTION 133

Which of the following lists factors That contribute to PROJECT risks?

- A. skill and staff shortages; problems in defining the right requirements, contractual issues.
- B. skill and staff shortages; software does not perform its intended functions; problems in defining the right requirements.
- C. problems in defining the right requirements; contractual issues; poor software quality characteristics.
- D. poor software quality characteristics; software does not perform its intended functions.

Answer: A

Explanation:

Project risks are the uncertainties or threats that may affect the project objectives, such as scope, schedule, cost, and quality. According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, some of the factors that contribute to project risks are:

? Skill and staff shortages: This factor refers to the lack of adequate or qualified human resources to perform the project tasks. This may result in delays, errors, rework, or low productivity.

? Problems in defining the right requirements: This factor refers to the difficulties or ambiguities in eliciting, analyzing, specifying, validating, or managing the requirements of the project. This may result in misalignment, inconsistencies, gaps, or changes in the requirements, affecting the project scope and quality.

? Contractual issues: This factor refers to the challenges or disputes that may arise from the contractual agreements between the project parties, such as clients, suppliers, vendors, or subcontractors. This may result in legal, financial, or ethical risks, affecting the project delivery and satisfaction.

The other options are not correct because they list factors that contribute to PRODUCT risks, not project risks. Product risks are the uncertainties or threats that may affect the quality or functionality of the software product or system. Some of the factors that contribute to product risks are:

? Poor software quality characteristics: This factor refers to the lack of adherence or compliance to the quality attributes or criteria of the software product or system, such as reliability, usability, security, performance, or maintainability. This may result in defects, failures, or dissatisfaction of the users or stakeholders.

? Software does not perform its intended functions: This factor refers to the deviation or discrepancy between the expected and actual behavior or output of the software product or system. This may result in errors, faults, or malfunctions of the software product or system.

References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 1: Fundamentals of Testing, Section 1.5: Risks and Testing, Pages 14-16.

NEW QUESTION 136

Confirmation testing is performed after:

- A. a defect is fixed and after other tests do not find any side-effect introduced in the software as a result of such fix
- B. a failed test, and aims to run that test again to confirm that the same behavior still occurs and thus appears to be reproducible
- C. the execution of an automated regression test suite to confirm the absence of false positives in the test results
- D. a defect is fixed, and if such testing is successful then the regression tests that are relevant for such fix can be executed

Answer: D

Explanation:

Confirmation testing is performed after a defect is fixed, and if such testing is successful then the regression tests that are relevant for such fix can be executed. Confirmation testing, also known as re-testing, is the process of verifying that a defect has been resolved by running the test case that originally detected the defect. Confirmation testing is usually done before regression testing, which is the process of verifying that no new defects have been introduced in the software as a result of changes or fixes. Therefore, option D is the correct answer.

References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 2.4.1, page 28; ISTQB® Glossary v4.02, page 15.

NEW QUESTION 138

The statement: "Test activities should start in the early stages of the lifecycle, adhering to the testing principle of early testing?? is relevant to which of the recognized software development models?

- A. Sequential development model.
- B. Iterative development model.
- C. Incremental development model.
- D. All the above

Answer: D

Explanation:

The principle of early testing is applicable to all recognized software development models, including sequential, iterative, and incremental models. Starting test activities early in the lifecycle helps in identifying and addressing defects as soon as possible, which can save time and costs by preventing defects from propagating to later stages of development. This proactive approach enhances the overall quality and efficiency of the software development process.Reference:ISTQB CTFL Syllabus V4.0, Section 1.3

NEW QUESTION 141

Following a risk-based testing approach you have designed 10 tests to cover a product risk with a high-risk level. You want to estimate, adopting the three-point

test estimation technique, the test effort required to reduce the risk level to zero by executing those 10 tests. You made the following three initial estimates:

- most optimistic = 6 person hours
- most likely = 30 person hours
- most pessimistic = 54 person hours

Based only on the given information, which of the following answers about the three-point test estimation technique applied to this problem is true?

- A. The final estimate is between 22 person hours and 38 person hours
- B. The final estimate is exactly 30 person hours because the technique uses the initial most likely estimate as the final estimate
- C. The final estimate is between 6 person hours and 54 person hours
- D. The final estimate is exactly 30 person hours because the technique uses the arithmetic mean of the three initial estimates as the final estimate

Answer: A

Explanation:

The three-point test estimation technique is a method of estimating the test effort based on three initial estimates: the most optimistic, the most likely, and the most pessimistic. The technique uses a weighted average of these three estimates to calculate the final estimate, which is also known as the expected value. The formula for the expected value is:

Expected value = (most optimistic + 4 * most likely + most pessimistic) / 6 Using the given values, the expected value is:

Expected value = (6 + 4 * 30 + 54) / 6 Expected value = 30 person hours

However, the expected value is not the only factor to consider when estimating the test effort. The technique also calculates the standard deviation, which is a measure of the variability or uncertainty of the estimates. The formula for the standard deviation is: Standard deviation = (most pessimistic - most optimistic) / 6 Using the given values, the standard deviation is:

Standard deviation = (54 - 6) / 6 Standard deviation = 8 person hours

The standard deviation can be used to determine a range of possible values for the test effort, based on a certain level of confidence. For example, using a 68% confidence level, the range is:

Expected value ?? standard deviation Using the calculated values, the range is: 30 ?? 8 person hours

Therefore, the final estimate is between 22 person hours and 38 person hours, which is option A.

References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 2.3.2, page 24-25; ISTQB® Glossary v4.02, page 33.

NEW QUESTION 142

Which of the following is the most correct statement about static testing techniques?

- A. Static techniques can be used before all code is ready for execution
- B. Static techniques find more defects than dynamic techniques.
- C. Static techniques can be used by inexperienced users.
- D. Static techniques are always cheaper than dynamic techniques.

Answer: A

Explanation:

Static 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 147

Which of the following statements is NOT true about Configuration management and software testing?

- A. Configuration management helps maintain consistent versions of software artifacts.
- B. Configuration management supports the build process, which is essential for delivering a test release into the test environment.
- C. When testers report defects, they need to reference version-controlled items.
- D. Version controlled test ware increases the chances of finding defects in the software under test.

Answer: D

Explanation:

Reference:ISTQB CTFL Syllabus V4.0, Section 5.4

NEW QUESTION 148

Which of the following statements best describes the difference between product risk and project risk in software testing?

- A. Product risk refers to the risk associated with the project's schedule, budget, and resources, while project risk refers to the risk associated with the quality and functionality of the software product.
- B. Product risk refers to the risk associated with issues such as delays in work product deliveries, inaccurate estimates, while project risk refers to the risk associated with the project's schedule, budget, and resources.
- C. Product risk and project risk are essentially the same and can be used interchangeably.
- D. Product risk refers to the risk associated with delays in elements such as work product deliveries and inaccurate estimates, while project risk refers to the risk associated with issues such as user dissatisfaction.

Answer: B

Explanation:

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 151

Which of the following statements are true?

- * 1. Early and frequent feedback helps to avoid requirements misunderstanding.
- * 2. Early feedback allows teams to do more with less.
- * 3. Early feedback allows the team to focus on the most Important features.
- * 4. Early and frequent feedback clarifies customer feedback by applying static testing techniques

Select the correct Answer:

- A. 3
- B. 2
- C. 1
- D. 4

Answer: C

Explanation:

The statement "Early and frequent feedback helps to avoid requirements misunderstanding" is true. Early feedback from stakeholders, through reviews and other static testing techniques, helps clarify requirements and ensures that any misunderstandings are addressed promptly. This practice aligns with Agile principles and contributes to developing software that meets user needs more accurately. References: ISTQB CTFL Syllabus, Section 2.1.1, "The Influence of Development Models on Testing" and Section 3.2.1, "The Advantages of Early Feedback."

NEW QUESTION 154

Which of the following about typical information found within a test plan is FALSE?

- A. The need to temporarily have additional test personnel available for specific test phases and/or test activities
- B. The conditions that must be met in order for the test execution activities to be considered completed.
- C. The list of the product risks which have not been fully mitigated at the end of test execution.
- D. The conditions that must be met for part of all the planned activities to be suspended and resumed.

Answer: C

Explanation:

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 157

A typical objective of testing is to ensure that:

- A. testing is used to drive the development of a software
- B. a software has been tested using a combination of test techniques
- C. there are no defects in a software that is about to be released
- D. a software has been properly covered

Answer: B

Explanation:

This answer is correct because a typical objective of testing is to ensure that a software has been tested using a combination of test techniques, such as black-box, white-box, or experience-based techniques, that are appropriate for the test objectives, test levels, and test types. Testing using a combination of test techniques can increase the effectiveness and efficiency of testing, as different techniques can target different aspects of the software quality, such as functionality, usability, performance, security, reliability, etc. Testing using a combination of test techniques can also reduce the risk of missing defects that could be detected by one technique but not by another. References: ISTQB Foundation Level Syllabus v4.0, Section 2.3.1.1, Section 2.3.2

NEW QUESTION 159

The whole-team approach:

- A. promotes the idea that all team members should have a thorough understanding of test techniques
- B. is a consensus-based approach that engages the whole team in estimating the user stories
- C. promotes the idea that all team members should be responsible for the quality of the product
- D. is mostly adopted in projects aimed at developing safety-critical systems, as it ensures the highest level of testing independence

Answer: C

Explanation:

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 162

During component testing of a program if 100% decision coverage is achieved, which of the following coverage criteria is also guaranteed to be 100%?

- A. 100% State transition coverage
- B. 100% Equivalence class coverage
- C. 100% Boundary value coverage
- D. 100% Statement coverage

Answer: D

Explanation:

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 stale transition coverage, equivalence class coverage, and boundary value coverage, are not guaranteed to be 100% by achieving 100% decision coverage. Stale 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 167

A software company decides to invest in reviews of various types. The thought process they have is that each artifact needs to be reviewed using only one of the review methods depending on the criticality of the artifact.

- A. The thought process is incorrect
- B. The whole company should adopt same standard for review of all artifacts.
- C. The thought process is correct
- D. The whole company should decide on the review method based on their CMM level.
- E. The thought process is incorrect
- F. Same artifact can be reviewed using different review methods
- G. The thought process is correct
- H. It wastes time to review same artifact using different review methods

Answer: C

Explanation:

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 172

Which of the following statements about static testing and dynamic testing is true?

- A. Unlike dynamic testing, which can be also performed manually, static testing cannot be performed without specialized tools
- B. Static testing is usually much less cost-effective than dynamic testing
- C. Unlike dynamic testing, which focuses on detecting potential defects, static testing focuses on detecting failures which may be due to actual defects
- D. Both static testing and dynamic testing can be used to highlight issues associated with non-functional characteristics

Answer: D

Explanation:

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 174

After being in operation for many years, a document management system must be decommissioned as it has reached its end of life. This system will not be replaced by any other new system. A legal obligation provides that all documents within the system must be kept for at least 20 years in a state archive. Which of the following statements about maintenance testing for decommissioning of this system is TRUE?

- A. No maintenance testing is required as this system will not be replaced.
- B. Data migration testing is required as part of maintenance testing
- C. Confirmation testing is required as part of maintenance testing.
- D. Regression testing is required as part of maintenance testing

Answer: B

Explanation:

Data migration testing is a critical part of maintenance testing during the decommissioning of a system. When a system is decommissioned, data often needs to be transferred to another system or archived securely. This process ensures that the data remains intact, accessible, and secure in its new location. Therefore, statement B is true as it aligns with the ISTQB CTFL syllabus guidelines on handling system decommissioning and data preservation.

NEW QUESTION 175

The following decision table is used to assist a doctor in determining the drug therapy to prescribe for a patient (aged 6 to 65 years) diagnosed with acute sinusitis. The table consists of three Boolean conditions and six actions

	1	2	3	4	5	6	7	8
Conditions								
Is the patient over 18 years old?	F	F	F	F	T	T	T	T
Is the patient allergic to Penicillin?	F	F	T	T	F	F	T	T
Is the patient taking anticoagulant therapy?	F	T	F	T	F	T	F	T
Actions								
Amoxicillin is the therapy of choice					X			
Levofloxacin is the therapy of choice			X				X	
Cefuroxime is the therapy of choice	X							
Necessary consultation with the hematologist		X		X		X		X
Full dosage recommended for 10 days					X		X	
Half of the full recommended dosage for 10 days	X		X					

Based only on the given information, which of the following statements is TRUE?

- A. Column 7 represents an impossible situation and thus can be deleted
- B. Columns 1 and 3 can be merged into a single column
- C. Columns 2, 4, 6 and 8 can be merged into a single column
- D. Columns 5 and 7 can be merged into a single column

Answer: B

Explanation:

Decision tables are used to model complex decision logic by considering different combinations of conditions and actions. Based on the given decision table for prescribing drug therapy:

? Column 1 and Column 3 both result in the same actions (prescribing Amoxicillin).

? These columns can be merged because the actions taken do not depend on whether the patient is taking anticoagulant therapy (both are 'T' for this condition). Thus, combining these columns simplifies the decision table without losing any information. Reference: ISTQB CTFL Syllabus V4.0, Chapter 4.2.3, Decision Table Testing.

NEW QUESTION 180

You are testing a room upgrade system for a hotel. The system accepts three differed types of room (increasing order of luxury): Platinum. Silver and Gold Luxury. ONLY a Preferred Guest Card holder s eligible for an upgrade.

Below you can find the decision table defining the upgrade eligibility:

Conditions

Preferred Guest Card holder	YES	YES	NO	NO
Room Type	Silver	Platinum	Silver	Platinum

Offer upgrade to Gold Luxury	YES	NO	NO	NO
Offer upgrade to Silver	N/A	YES	N/A	NO

What is the expected result for each of the following test cases? Customer A: Preference Guest Card holder, holding a Silver room Customer B: Non Preferred Guest Card holder, holding a Platinum room

- A. Customer A; doesn't offer any upgrade; Customer B: offers upgrade to Gold luxury room
- B. Customer A: doesn't offer any upgrade; Customer B: doesn't offer any upgrade.
- C. Customer A: offers upgrade to Gold Luxury room; Customer B: doesn't offer any upgrade
- D. Customer A: offers upgrade to Silver room; Customer B: offers upgrade to Silver room.

Answer: C

Explanation:

According to the decision table in the image, a Preferred Guest Card holder with a Silver room is eligible for an upgrade to Gold Luxury (YES), while a non-Preferred Guest Card holder, regardless of room type, is not eligible for any upgrade (NO). Therefore, Customer A (a Preferred Guest Card holder with a Silver room) would be offered an upgrade to Gold Luxury, and Customer B (a non-Preferred Guest Card holder with a Platinum room) would not be offered any upgrade. References = The answer is derived directly from the decision table provided in the image; specific ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents are not referenced.

NEW QUESTION 181

For the same financial institution in Question 12, with the same requirements and expectations, what would be the most likely investment values used in testing if two-point boundary value analysis is used to design test cases specific to the 13% interest rate equivalence partition?

- A. R100 000, R100 001, R500 000, R500 001.
- B. R99 999, R100 000, R499 999, R500 000.
- C. R100 000. R500 000.
- D. R99 000, R500 001.

Answer: A

Explanation:

For boundary value analysis, the test cases should include the boundary values just inside and just outside the equivalence partition for the 13% interest rate range:

- ? R100,000 (just inside the previous range)
- ? R100,001 (start of the 13% range)
- ? R500,000 (end of the 13% range)
- ? R500,001 (just outside the range)

These values ensure that both the edges of the partition are tested.

NEW QUESTION 183

An application is subjected to a constant load for an extended period of time as part of a performance test While running this test, the response time of the application steadily slows down, which results in a requirement not being met This slowdown is caused by a memory leak where the application code does not properly release some of the dynamically allocated memory when it is no longer needed. Which of the following statements is TRUE?

- A. The slowdown is a failure while the memory leak is a defect
- B. The slowdown is a defect while the memory leak is an error.
- C. The slowdown is an error; the memory leak is a defect.
- D. The slowdown is a defect; the memory leak is a failure.

Answer: A

Explanation:

In software testing terminology, a failure is an observable deviation of the software from its expected behavior. A defect (or bug) is the cause of the failure in the software's code. In this case, the observed slowdown is the failure, while the underlying memory leak in the application code is the defect causing this failure. This distinction is clearly outlined in the ISTQB CTFL Syllabus v4.0, which differentiates between failures (observable issues) and defects (underlying issues in the code).

NEW QUESTION 185

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

CTFL4 Practice Exam Features:

- * CTFL4 Questions and Answers Updated Frequently
- * CTFL4 Practice Questions Verified by Expert Senior Certified Staff
- * CTFL4 Most Realistic Questions that Guarantee you a Pass on Your First Try
- * CTFL4 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The CTFL4 Practice Test Here](#)