

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 of the statements correctly describes when a whole team approach may NOT be suitable?

- A. When a high level of test independence may be required.
- B. When acceptance tests need to be created.
- C. When a test automation approach needs to be determined.
- D. When the team dynamics need to be improved.

Answer: A

Explanation:

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

NEW QUESTION 3

A calculator software is used to calculate the result for 5+6. The user noticed that the result given is 6. This is an example of;

- A. Mistake
- B. Fault
- C. Error
- D. Failure

Answer: D

Explanation:

According to the ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 18, a failure is ??an event in which a component or system does not perform a required function within specified limits??. In this case, the calculator software does not perform the required function of calculating the correct result for 5+6 within the specified limits of accuracy and precision. Therefore, this is an example of a failure.

The other options are incorrect because:

? A mistake is ??a human action that produces an incorrect result?? (page 25). A mistake is not an event, but an action, and it may or may not lead to a failure. For example, a mistake could be a typo in the code, a wrong assumption in the design, or a misunderstanding of the requirement.

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

? An error is ??the difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition?? (page 15). An error is not an event, but a difference, and it may or may not result in a failure. For example, an error could be a rounding error in the calculation, a measurement error in the observation, or a deviation error in the condition.

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

NEW QUESTION 4

Which of the following statements about traceability is FALSE?

- A. Traceability between test basis items and the test cases designed to cover them, makes it possible to determine which test basis items have been covered by the executed test cases.
- B. Traceability between test basis items and the test cases designed to cover them, enables experience-based test techniques to be applied
- C. Traceability between test basis items and the test cases designed to cover them, enables identification of which test cases will be affected by changes to the test basis items.
- D. Traceability can be established and maintained through all the test documentation for a given test level, such as from test conditions through test cases to test scripts.

Answer: B

Explanation:

Traceability primarily refers to the ability to link test cases back to their sources in the test basis, such as requirements or design documents. This linkage allows for the determination of coverage, impact analysis, and maintaining consistency across test documentation. However, traceability does not directly enable the application of experience-based test techniques, which are more about using the tester's intuition and experience. The ISTQB CTFL Syllabus v4.0 does not state that traceability enables experience-based techniques, making option B the false statement.

NEW QUESTION 5

Which of the following work products cannot be examined by static analysis?

- A. Test plans
- B. Source code
- C. Compiled code
- D. Formal models

Answer: A

Explanation:

Static analysis is the process of examining the work products of a software development or testing activity without executing them. Static analysis can be applied to various types of work products, such as requirements, design, code, test cases, etc. However, test plans are not suitable for static analysis, because they are high-level documents that describe the test objectives, scope, strategy, resources, schedule, and risks of a testing project. Test plans are not executable or formalized in a way that static analysis tools can analyze them. Therefore, option A is the correct answer.

References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 2.2.1, page 20; ISTQB® Glossary v4.02, page 45.

NEW QUESTION 6

Which of the following statements is TRUE?

- A. User acceptance tests are usually automated and aim to verify the acceptance criteria for user stories
- B. Acceptance criteria for user stories can include details on data definitions, for example by describing the format, allowed values, and default values for a data item
- C. Acceptance criteria for user stories should focus on positive scenarios, while negative scenarios should be excluded
- D. Tests derived from acceptance criteria for user stories are not included in any of the four testing quadrants

Answer: B

Explanation:

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

NEW QUESTION 7

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 8

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 9

A requirement specifies that a certain identifier (ID) must be between 5 and 10 characters long, must contain only alphanumeric characters, and its first character must be a letter. As a tester, you want to apply one-dimensional equivalence partitioning to test this ID. This means that you have to apply equivalence partitioning individually: to the length of the ID, the type of characters contained within the ID, and the type of the first character of the ID.

What is the number of partitions to cover?

- A. 7.
- B. 6.
- C. 5.
- D. 3.

Answer: A

Explanation:

To apply one-dimensional equivalence partitioning to the ID requirement, we need to consider each condition individually:

? Length of the ID: Valid partitions (5-10 characters), Invalid partitions (less than 5, more than 10) = 3 partitions.

? Type of characters: Valid partitions (alphanumeric), Invalid partitions (non-alphanumeric) = 2 partitions.

? First character: Valid partitions (letter), Invalid partitions (non-letter) = 2 partitions.

Adding these partitions, we get a total of 3 (length) + 2 (character type) + 2 (first character) = 7 partitions. Thus, the correct answer is A.

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 10

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 11

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 14

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 19

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 20

Which of the following statements about the test pyramid is TRUE?

- A. Each layer of the test pyramid groups tests related to a single non-functional quality characteristic.
- B. The higher the layer of the test pyramid, the more production code a single automated test within the layer tends to cover
- C. The higher the layer of the test pyramid, the more maintainable a single automated test within the layer tends to be
- D. The higher the layer of the test pyramid, the more isolated a single automated test within the layer tends to be.

Answer: B

Explanation:

The test pyramid concept suggests that there should be more low-level tests (unit tests) and fewer high-level tests (end-to-end tests).

? As we move higher up the pyramid (e.g., from unit tests to integration tests to end-to-end tests), each test covers more production code.

? Higher-level tests (like end-to-end) validate larger parts of the application, including multiple units and their interactions.

This aligns with the principle that higher-level tests provide broader coverage but are fewer in number and more expensive to run and maintain.

Reference: ISTQB CTFL Syllabus V4.0, Chapter 5.1.6, Test Pyramid.

NEW QUESTION 21

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 23

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 27

Which of the following coverage criteria results in the highest coverage for state transition based test cases?

A. Can't be determined

B. Covering all transitions at least once

C. Covering only start and end states

D. Covering all states at least once

Answer: B

Explanation:

Covering all transitions at least once is the highest coverage criterion for state transition based test cases, because it ensures that every possible change of state is tested at least once. This means that all the events that trigger the transitions, as well as the actions and outputs that result from the transitions, are verified.

Covering all transitions at least once also implies covering all states at least once, but not vice versa. Therefore, option D is not the highest coverage criterion.

Option C is the lowest coverage criterion, because it only tests the initial and final states of the system or component, without checking the intermediate states or transitions. Option A is incorrect, because the coverage criteria for state transition based test cases can be determined and compared based on the

number of transitions and states covered. References = CTFL 4.0 Syllabus, Section 4.2.3, page 49-50.

NEW QUESTION 29

What is test oracle?

A. The source of lest 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 32

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 34

Which of the following statements about TDD, BDD and ATDD is TRUE?

- A. Refactonng is a practice that is an integral part of TDD and is applied both to tests and to code wntten to satisfy those tests.
- B. ATDD is a black-box test design technique that is applicable exclusively at acceptance test level.
- C. BDD is a developer practice where business stakeholders are not usually involved as the tests are directly written at unit/component test level.
- D. ATDD is the practice of running the automated acceptance tests as part of a continuous integration process.

Answer: A

Explanation:

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

NEW QUESTION 39

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 agreemet 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 40

Which of the following statements is TRUE'?

- A. Unlike functional testing, non-fundional 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 42

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 44

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

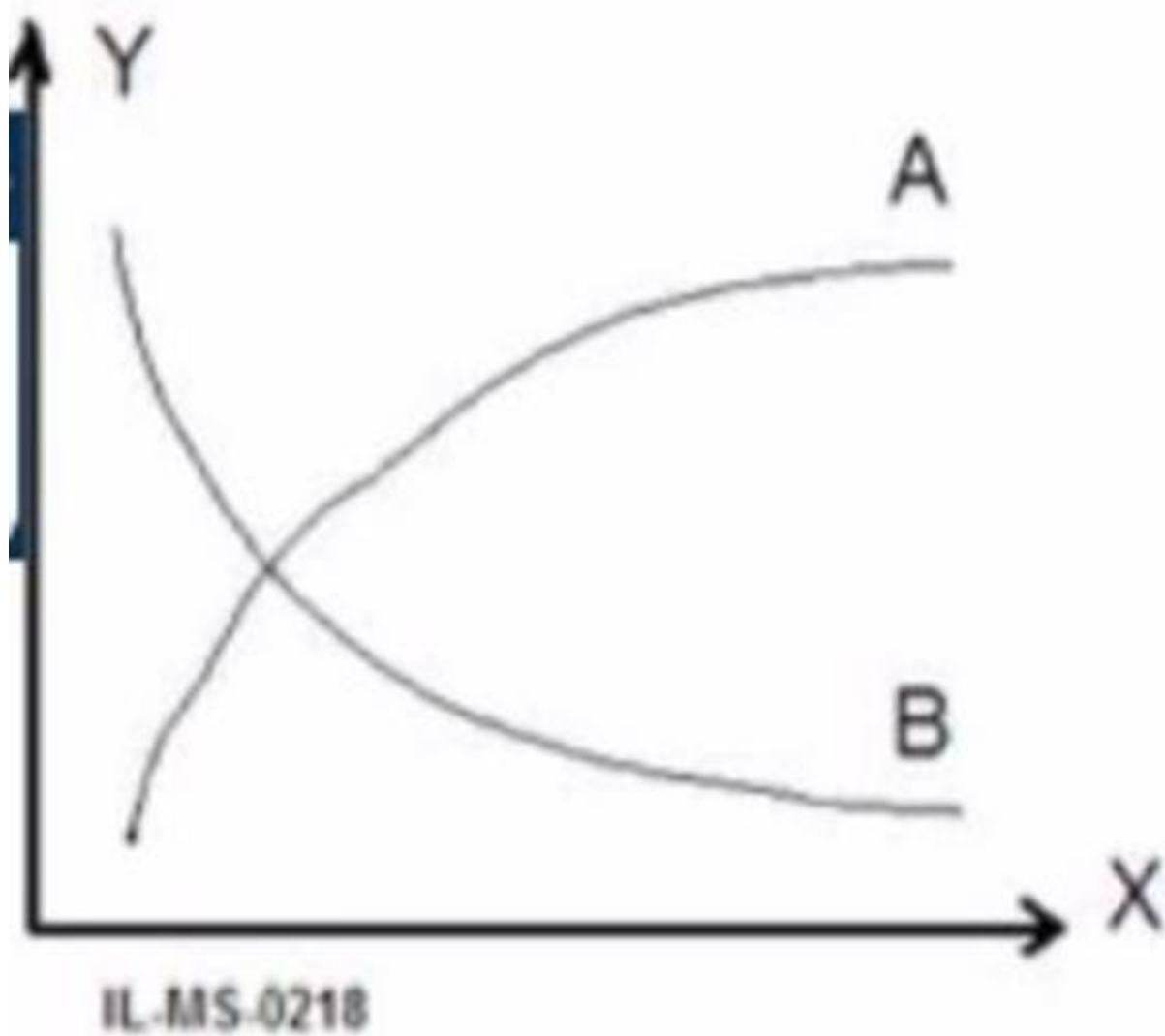
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles1

? 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 49

The following chart represents metrics related to testing of a project that was completed. Indicate what is represented by tie lines A, B and the axes X.Y



A)

- X - Time
Y - Cost
A - Cost of test (per week)
B - Cost of finding a single bug (per week)
- B)
- X - Time
Y - Number of defects
A - Number of open defects
B - Number of closed defects
- C)
- X - Time
Y - Percent
A - % of functional tests in the test suite
B - % of non-functional tests in the test suite
- D)
- X - Time
Y - Count
A - Total number of executed tests
B - Number of open bugs

- A. Option A
B. Option B
C. Option C
D. Option D

Answer: D

Explanation:

Option D correctly explains what is represented by the lines A, B and the axes X, Y in a testing metrics chart. According to option D:

? X-axis represents Time

? Y-axis represents Count

? Line A represents Number of open bugs

? Line B represents Total number of executed tests

This information is essential in understanding and analyzing the testing metrics of a completed project.

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 2.5.1, Page 35.

NEW QUESTION 50

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 53

What type of testing measures its effectiveness by tracking which lines of code were executed by the tests?

- A. Acceptance testing
B. Structural testing
C. Integration testing
D. Exploratory testing

Answer: B

Explanation:

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

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

NEW QUESTION 58

The following 4 equivalence classes are given:

$x \leq -100$
 $-100 < x < 100$
 $100 \leq x < 1000$
 $x \geq 1000$

Which of the following alternatives includes correct test values for x. based on equivalence partitioning?

- A. -100; 100;1000; 1001
- B. -500; 0; 100; 1000
- C. -99; 99;101; 1001
- D. -1000; -100; 100; 1000

Answer: D

Explanation:

? The question is about selecting the correct test values for x based on equivalence partitioning. Equivalence partitioning is a software test design technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In this case, the given equivalence classes are:
Option D provides a value from each of these partitions:

? For ($x \leq -100$), it gives -1000.

? For ($-100 < x < 100$), it gives -100 and 100.

? For ($100 \leq x < 1000$), it gives 500.

? For ($x \geq 1000$), it gives 1500.

So, option D covers all four given equivalence classes with appropriate values. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents available at ISTQB and ASTQB.

? 1: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 38

? 2: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 39

? : ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 40

NEW QUESTION 60

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.I
- B. Ensuring the proper use of testing tools.H
- C. Coaching other team members in the relevant aspects of testing.i
- 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 65

During which main group of test activity are the following tasks performed?

- Checking test results and logs against specified coverage criteria.
- Assessing the level of component or system quality based on test results and logs.
- Determining whether more tests are needed. Select the correct Answer:

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

Answer: D

Explanation:

The activities of checking test results and logs against specified coverage criteria, assessing the level of component or system quality based on test results and logs, and determining whether more tests are needed fall under the category of test monitoring and control. This phase involves ongoing assessment and adjustment of the test activities to ensure they meet the test objectives and quality goals.

NEW QUESTION 70

A Test Manager conducts risk assessment for a project. One of the identified risks is: The sub-contractor may fail to meet his commitment". If this risk materializes, it will lead to delay in completion of testing required for the current cycle.

Which of the following sentences correctly describes the risk?

- A. It is a product risk since any risk associated with development timeline is a product risk.
- B. It is no longer a risk for the Test Manager since an independent party (the sub- contractor) is now managing it
- C. It is a object risk since successful completion of the object depends on successful and timely completion of the tests
- D. It is a product risk since default on part of the sub-contractor may lead to delay in release of the product

Answer: D

Explanation:

? A product risk is a risk that affects the quality or timeliness of the software product being developed or tested1. Product risks are related to the requirements, design, implementation, verification, and maintenance of the software product2.

? The risk of the sub-contractor failing to meet his commitment is a product risk, as it could cause a delay in the completion of the testing required for the current cycle, which in turn could affect the release date of the product. The release date is an important aspect of the product quality, as it reflects the customer satisfaction and the market competitiveness of the product3.

? The other options are not correct because: References =

? 1 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 97

? 2 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 98

? 3 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 99

? 4 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 100

? 5 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 101

? 6 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 102

NEW QUESTION 75

A test manager decided to skip static testing since he believes bugs can be found easily by doing dynamic testing. Was this decision right or wrong?

- A. The decision was wron
- B. Ensuring quality mandates that static testing is performed after performing the dynamic testing.
- C. The decision was righ
- D. Static testing is usually redundant if a product is planned to go through a full-cycle of dynamic testing.
- E. The decision was righ
- F. Most of the bugs are easier to identify during the dynamic testing.
- G. The decision was wron
- H. Static testing can find defects early in the development process, reducing the overall cost of testing and development

Answer: D

Explanation:

Static testing is a form of testing that does not involve executing the software or system under test. It includes activities such as reviews, inspections, walkthroughs, and analysis of documents, code, and models. Static testing can find defects early in the development process, before they become more expensive and difficult to fix in later stages. Static testing can also improve the quality of the software or system by preventing defects from being introduced in the first place. Static testing can complement dynamic testing, which involves executing the software or system under test and checking the results against expected outcomes. Dynamic testing can find defects that static testing may miss, such as performance, usability, or integration issues. However, dynamic testing alone is not sufficient to ensure quality, as it may not cover all possible scenarios, inputs, or paths. Therefore, a test manager who decides to skip static testing is making a wrong decision, as he or she is ignoring the benefits of static testing and relying solely on dynamic testing, which may not be effective or efficient enough to find and prevent

defects. References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.1.1, page 14; ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 36; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 3, page 9.

NEW QUESTION 76

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 81

Which of the following statements about white-box testing is FALSE?

- A. Static testing can benefit from using code-related white-box test techniques during code reviews.
- B. White-box testing allows suggesting test cases for increasing coverage levels which are based on objective measures
- C. Achieving full code coverage for a component or a system ensures that it has been fully tested
- D. Black-box testing can benefit from using code-related white-box test techniques to increase confidence in the code.

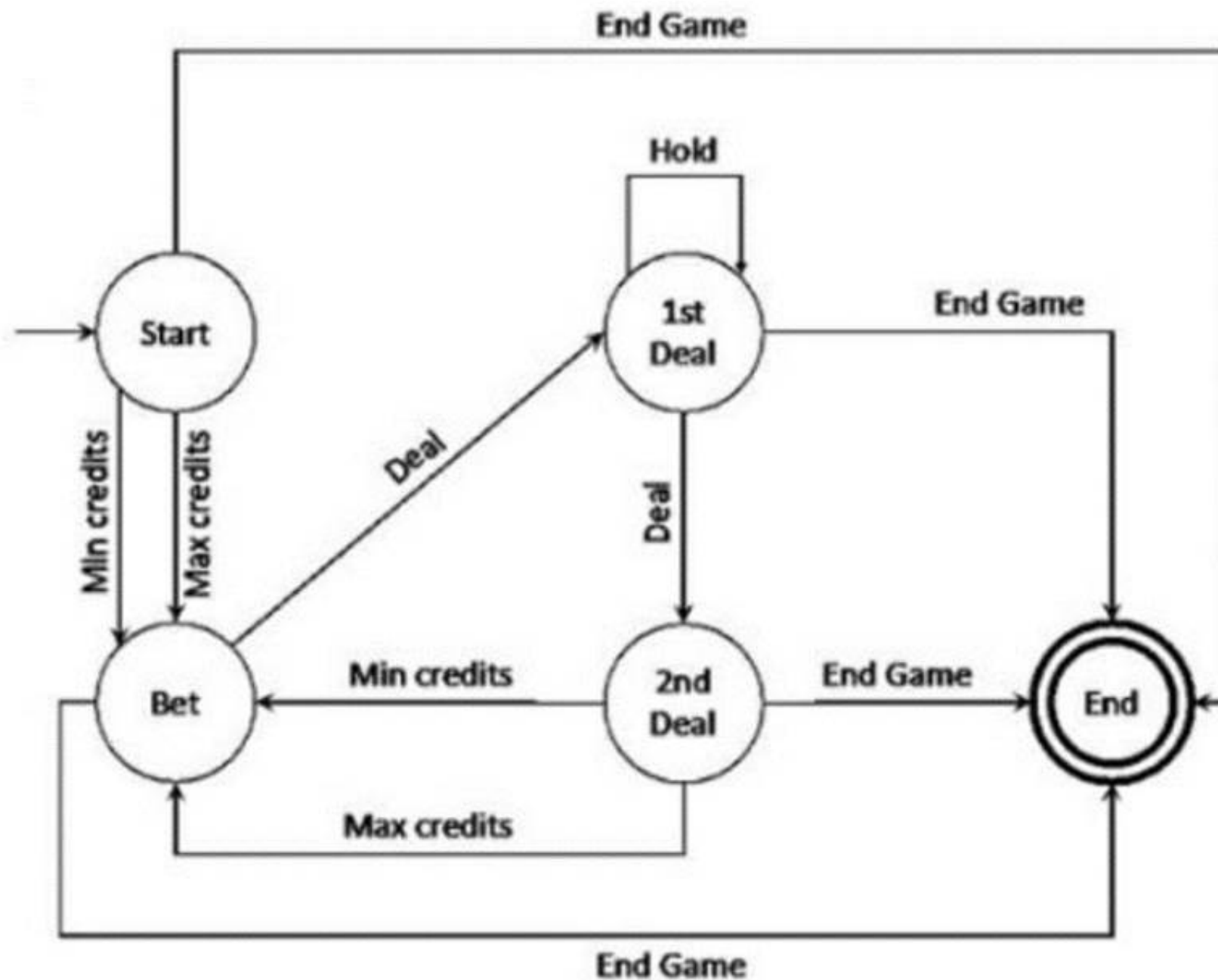
Answer: C

Explanation:

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

NEW QUESTION 86

Consider the following simplified version of a state transition diagram that specifies the behavior of a video poker game:



What Is the minimum number of test cases needed to cover every unique sequence of up to 3 states/2 transitions starting in the "Start" state and ending in the "End" state?

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

Answer: D

Explanation:

The minimum number of test cases needed to cover every unique sequence of up to 3 states/2 transitions starting in the ??Start?? state and ending in the ??End?? state is 4. This is because there are 4 unique sequences of up to 3 states/2 transitions starting in the ??Start?? state and ending in the ??End?? state:

? Start -> Bet -> End

? Start -> Deal -> End

? Start -> 1st Deal -> End

? Start -> 2nd Deal -> End References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents.

NEW QUESTION 89

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 93

Consider an estimation session in which a six-member Agile team (Memb1..... Memb6) uses the planning poker technique to estimate a user story (in story points). The team members will use a set of cards with the following values: 1,2, 3,5, 8,13,21. Below is the outcome of the first round of estimation for this session:

Memb1 = 3
Memb4 = 21

Memb2 = 3
Memb5 = 3

Memb3 = 3
Memb6 = 1

Which of the following answers BEST describes how the estimation session should proceed?

- A. The final estimate of the user story in story points is determined by applying the three- point estimation technique with the following input values most optimistic estimate - 1, most likely estimate - 3, and most pessimistic estimate - 21
- B. Further estimation rounds should be performed until all team members will pick the card having the same value: this value will represent the final estimate of the user story in story points.
- C. The final estimate of the user story in story points is determined by calculating the average value between the most optimistic estimate of 21 story points (Memb4) and the most pessimistic estimate of 1 story point (Memb6)
- D. Memb6 and Memb4 which have produced the most pessimistic and the most optimistic estimates respectively, should explain the reasons of their choices to stimulate a discussion between all members before proceeding to another estimation round

Answer: D

Explanation:

In Agile teams using the planning poker technique for estimating user stories, it is common practice to have further discussions and rounds of estimation if there is a significant discrepancy in the initial estimates. This helps in reaching a consensus and ensures that all team members understand the complexity and requirements of the user story. According to the ISTQB CTFL syllabus, planning poker involves discussions to clarify differences in estimates, especially when there is a wide range of values selected. By having Memb6 and Memb4, who provided the most pessimistic and optimistic estimates, explain their reasoning, it fosters a deeper understanding and encourages the team to converge towards a more accurate and agreed-upon estimate.

References:ISTQB CTFL Syllabus, Section on Agile methodologies and estimation techniques.

NEW QUESTION 94

Which two of the following statements describe the advantages provided by good traceability between the test basis and test work products?

- A. Analyzing the impact of changes.i
- B. A measure of code quality.ii
- C. Accurate test estimation.i
- D. Making testing auditable
- E. Select the correct Answer:
- F. i and ii
- G. i and iv
- H. i and iii
- I. ii and iii

Answer: B

Explanation:

Good traceability between the test basis and test work products provides several advantages: i.Analyzing the impact of changes:Traceability allows for easy identification of which parts of the test work products will be affected by changes in the requirements or design, facilitating impact analysis. iv.Making testing auditable:Traceability ensures that there is a clear connection between the requirements and the test cases, which makes the testing process auditable and provides evidence that all requirements have been tested.

NEW QUESTION 98

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 100

Which of the following is the most correct statement about state 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:

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

NEW QUESTION 104

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 108

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 109

Which of the following issues cannot be identified by static analysis tools?

- A. Very low MTBF (Mean Time Between failure)
- B. Potentially endless loops
- C. Referencing a variable with an undefined value
- D. Security vulnerabilities

Answer: A

Explanation:

Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential bugs¹². However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)³. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures. MTBF can only be estimated by dynamic testing, which involves executing the program under various conditions and collecting data on its failures⁴. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools. The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can be identified by static analysis tools. Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false⁵. Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables⁶. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption. References = What Is Static Analysis? Static Code Analysis Tools - Perforce Software, How Static Code Analysis Works | Perforce, Static Code Analysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs - CareerRide, ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

NEW QUESTION 113

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 116

Who of the following has the best knowledge to decide what tests in a test project should be automated?

- A. The developer
- B. The customer
- C. The development manager
- D. The test leader

Answer: D

Explanation:

The test leader is the person who is responsible for planning, monitoring, and controlling the test activities and resources in a test project. The test leader should have the best knowledge of the test objectives, scope, risks, resources, schedule, and quality criteria. The test leader should also be aware of the test automation criteria, such as the execution frequency, the test support, the team education, the roles and responsibilities, and the devs and testers collaboration¹. Based on these factors, the test leader can decide which tests are suitable for automation and which are not, and prioritize them accordingly. The test leader can also coordinate with the test automation engineers, the developers, and the stakeholders to ensure the alignment of the test automation strategy with the test project goals and expectations. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 2, Section 2.3.1, Page 152; ISTQB Glossary of Testing Terms v4.0, Page 403; ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 6, Section 6.1.1, Page 514; Top 8 Test Automation Criteria You Need To Fulfill - QAMIND¹

NEW QUESTION 121

From a testing perspective, configuration management

- A. Allows the expected results to be compared with the actual results.
- B. Allows the tracking of all changes to versions of the testware.
- C. Includes all activities that direct and control an organisation with regard to quality
- D. Focuses on configuring static analysis tools to choose the most suitable breadth and depth of analysis.

Answer: B

Explanation:

Configuration management in the context of testing involves the systematic control of changes to the configuration items, including testware such as test scripts, test data, and test environments. It ensures that all changes are tracked and recorded, enabling the version control and management of testware . Option A is related to test execution rather than configuration management. Option C describes quality management in a broader sense, not specifically configuration management. Option D is specific to the configuration of tools, not the overall management of testware versions.

NEW QUESTION 122

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 123

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 124

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 state transition coverage, equivalence class coverage, and boundary value coverage, are not guaranteed to be 100% by achieving 100% decision coverage. State transition coverage is a structural coverage metric that measures the percentage of transitions between states in a state machine that are executed by a test suite³.

Equivalence class coverage is a functional coverage metric that measures the percentage of equivalence classes (or partitions) of input or output values that are tested by a test suite⁴. Boundary value coverage is another functional coverage metric that measures the percentage of boundary values (or extreme values) of input or output ranges that are tested by a test suite⁴. These metrics are independent of decision coverage, because they are based on different aspects of the system under test, such as its behavior, functionality, or specification. Therefore, achieving 100% decision coverage does not imply achieving 100% of any of these metrics, and vice versa. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Test Coverage in Software Testing - Guru99, Structural Coverage Metrics - MATLAB & Simulink - MathWorks India, Test Design Coverage in Software Testing - GeeksforGeeks.

NEW QUESTION 128

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 129

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 134

Metrics can be collected during and at the end of testing activities to assess which of the following?

- A. Progress against the planned schedule and budget.I
- B. Current quality of the test objec
- C. H
- D. Adequacy of the test approach.i
- E. Effectiveness of the test activities with respect to the objectives.
- F. All the above.Select the correct Answer:
- G. Only i and ii.
- H. Only i and iii.
- I. Only I, ii and iv.
- J. Only v.

Answer: D

Explanation:

Metrics can be collected during and at the end of testing activities to assess various aspects including progress against the planned schedule and budget, the current quality of the test object, the adequacy of the test approach, and the effectiveness of the test activities with respect to the objectives. Collecting these metrics helps in understanding the overall performance and quality of the testing process.

NEW QUESTION 138

Which of the following statements is true?

- A. Functional testing focuses on what the system should do while non-functional testing on the internal structure of the system
- B. Non-functional testing includes testing of both technical and non-technical quality characteristics
- C. Testers who perform functional tests are generally expected to have more technical skills than testers who perform non-functional tests
- D. The test techniques that can be used to design white-box tests are described in the ISO/IEC 25010 standard

Answer: B

Explanation:

Non-functional testing includes testing of both technical and non-technical quality characteristics. Non-functional testing is the process of testing the quality attributes of a system, such as performance, usability, security, reliability, etc. Non-functional testing can be applied at any test level and can use both black-box and white-box test techniques. Non-functional testing can cover both technical aspects, such as response time, throughput, resource consumption, etc., and non-technical aspects, such as user satisfaction, accessibility, compliance, etc. Therefore, option B is the correct answer. References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 1.3.1, page 13; ISTQB® Glossary v4.02, page 40.

NEW QUESTION 139

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 144

A system has a self-diagnostics module that starts executing after the system is reset. The diagnostics are running 12 different tests on the system's memory hardware. The following is one of the requirements set for the diagnostics module:

'The time taking the diagnostics tests to execute shall be less than 2 seconds' Which of the following is a failure related to the specified requirement?

- A. The diagnostic tests fail to start after a system reset
- B. The diagnostic tests take too much time to execute
- C. The diagnostic tests that measure the speed of the memory, fail
- D. The diagnostic tests fail due to incorrect implementation of the test code

Answer: B

Explanation:

A failure is an event in which a component or system does not perform a required function within specified limits¹. A requirement is a condition or capability needed by a user to solve a problem or achieve an objective². In this case, the requirement is that the diagnostics tests should execute in less than 2 seconds. Therefore, any event that violates this requirement is a failure. The only option that clearly violates this requirement is B. The diagnostic tests take too much time to execute. If the diagnostic tests take more than 2 seconds to complete, then they do not meet the specified limit and thus fail. The other options are not necessarily failures related to the specified requirement. Option A. The diagnostic tests fail to start after a system reset is a failure, but not related to the time limit. It is related to the functionality of the self-diagnostics module. Option C. The diagnostic tests that measure the speed of the memory, fail is also a failure, but not related to the time limit. It is related to the accuracy of the memory tests. Option D. The diagnostic tests fail due to incorrect implementation of the test code is also a failure, but not related to the time limit. It is related to the quality of the test code. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Requirements Engineering Fundamentals.

NEW QUESTION 149

Which of the following statements about retrospectives is TRUE?

- A. Only developers and testers should be involved in retrospectives, as involving people in other roles is very likely to prevent developers and testers from having open and constructive discussions that really help identify process improvements.
- B. Retrospectives can be very effective in identifying process improvements and can also be very efficient and cost-effective especially since, unlike reviews, they do not require any follow-up activities
- C. On Agile projects, well-conducted retrospectives at the end of each iteration can help the team reduce and sometimes even eliminate the need for daily stand-up meetings.
- D. During retrospectives, in addition to identifying relevant process improvements, participants should also consider how to implement these improvements and retain them based on the context of the project, such as the software development lifecycle.

Answer: D

Explanation:

Retrospectives are a crucial part of Agile practices, aiming to identify process improvements and determine how to implement them effectively. They should involve participants discussing not only what improvements could be made but also how to integrate and sustain those improvements within the project context, including the software development lifecycle. This makes statement D accurate according to the ISTQB CTFL syllabus.

NEW QUESTION 150

Can "cost" be regarded as Exit criteria?

- A. Ye
- B. Spending too much money on test ng will result in an unprofitable product, andhaving cost as an exit criterion helps avoid this
- C. N
- D. The financial value of product quality cannot be estimated, so it is incorrect to use cost as an exit criterion
- E. Ye
- F. Going by cost as an exit criterion constrains the testing project which will hello achieve the desired quality level defined for the project
- G. No The cost of testing cannot be measured effectively, so it is incorrect to use cost as an exit criterion

Answer: A

Explanation:

Cost can be regarded as an exit criterion for testing, because it is a factor that affects the profitability and feasibility of the software product. Testing is an investment that aims to improve the quality and reliability of the software product, but it also consumes resources, such as time, money, and human effort. Therefore, testing should be planned and executed in a way that balances the cost and benefit of testing activities. Having cost as an exit criterion helps to avoid spending too much money on testing, which mayresult in an unprofitable product or a loss of competitive advantage. Cost can also help to prioritize and focus the testing efforts on the most critical and valuable features and functions of the software product. However, cost should not be the only exit criterion for testing, as it may not reflect the true quality and risk level of the software product. Other exit criteria, such as defect rate, test coverage, user satisfaction, etc., should also be considered and defined in the test plan.

The other options are incorrect, because they either deny the importance of cost as an exit criterion, or they make false or unrealistic assumptions about the cost of testing. Option B is incorrect, because the financial value of product quality can be estimated, for example, by using cost-benefit analysis, return on investment, or cost of quality models. Option C is incorrect, because going by cost as an exit criterion does not necessarily constrain the testing project or help achieve the desired quality level. Cost is a relative and variable factor that depends on the scope, complexity, and context of the software product and the testing project. Option D is incorrect, because the cost of testing can be measured effectively, for example, by using metrics, such as test effort, test resources, test tools, test environment, etc.

NEW QUESTION 155

Which of the following statements describes regression testing?

- A. Retesting of a fixed defectI
- B. Testing of an already tested programII
- C. Testing of new functionality in a programI
- D. Regression testing applies only to functional testingV Tests that do not nave to be repeatable, because They are only used once
- E. II, IV, V
- F. I, III, IV
- G. II
- H. I, IV

Answer: C

Explanation:

Regression testing is the re-running of functional and non-functional tests to ensure that previously developed and tested software still performs as expected after a change1 It does not involve retesting of a fixed defect, testing of new functionality, or applying only to functional testing. Tests that are used for regression testing should be repeatable, because they are used to verify the stability of the software after each change2 References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 4, Section 4.2.2, Page 291; ISTQB Glossary of Testing Terms v4.0, Page 292

NEW QUESTION 160

Which one of the following statements correctly describes the term 'debugging'?

- A. There is no difference between debugging and testing.
- B. Debugging is a confirmation activity that checks whether fixes resolved defects.
- C. Debugging is the development activity that finds, analyses and fixes defects.
- D. Debugging is of no relevance in Agile development.

Answer: C

Explanation:

Reference:ISTQB CTFL Syllabus V4.0, Section 1.1.2

NEW QUESTION 165

An alphanumeric password must be between 4 and 7 characters long and must contain at least one numeric character, one capital (uppercase) letter and one lowercase letter of the alphabet.

Which one of the following sets of test cases represents the correct outcome of a two-value boundary value analysis applied to the password length? (Note: test cases are separated by a semicolon)

- A. 1xA;aB11;Pq1ZZab;7iDD0a1x
- B. aB11;99rSp:5NnN10;7iDD0a1x
- C. 1xB: aB11: 99rSp: 5NnN10; 4NnN10T; 44ghWn19
- D. 1RhT;rSp53;3N3e10;8sBdbY

Answer: D

Explanation:

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

password, such as containing at least one numeric character, one capital letter, and one lowercase letter. The test cases in option D are:

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

? rSp53: a 5-character password that is valid

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

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

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

? aB11: a 4-character password that is valid

? Pq1ZZab: a 7-character password that is valid

? 7iDD0a1x: an 8-character password that is invalid References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

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

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

NEW QUESTION 166

As a tester, as part of a V-model project, you are currently executing some tests aimed at verifying if a mobile app asks the user to grant the proper access permissions during the installation process and after the installation process. The requirements specification states that in both cases the app shall ask the user to grant access permissions only to the camera and photos stored on the device. However, you observe that the app also asks the user to grant access permission to all contacts on the device. Consider the following items:

[I]. Test environment [ii]. Expected result [iii]. Actual result. [IV] Test level.

[V]. Root cause.

Based on only the given information, which of the items listed above, are you able to CORRECTLY specify in a defect report?

A. [I] and [IV]

B. [ii] and [III].

C. [ii], [iii] and [v]

D. [ii], [IV] and [V].

Answer: B

Explanation:

When writing a defect report, the tester can specify the expected result and the actual result based on the observation. The expected result is what the requirements specify, and the actual result is what was observed during testing. These elements are crucial for clearly communicating the nature of the defect to developers and other stakeholders. The other items such as test environment, test level, and root cause may not be clear or necessary at this stage of defect reporting.

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

NEW QUESTION 167

Which TWO of the following are benefits of continuous integration?

A. Allows earlier detection and easier root cause analysis of integration problems and conflicting changes.I

B. Removes the need for manual test analysis, design and execution.H

C. Removes the dependency on automated regression packs when integrating larger systems, or components.i

D. Gives the development team regular feedback on whether the code is workin

E. Select the correct Answer:

F. i and iv

G. i and ii

H. i and iii

I. iii and iv

Answer: A

Explanation:

The benefits of continuous integration include: i. Allows earlier detection and easier root cause analysis of integration problems and conflicting changes. iv. Gives the development team regular feedback on whether the code is working. These benefits help in maintaining the stability and quality of the codebase by integrating and testing changes frequently and providing quick feedback to developers.

NEW QUESTION 169

Which of the following statements best describes how configuration management supports testing?

A. Configuration management helps reduce testing effort by identifying a manageable number of test environment configurations in which to test the software, out of all possible configurations of the environment in which the software will be released

B. Configuration management is an administrative discipline that includes change control, which is the process of controlling the changes to identified items referred to as Configuration Items'

C. Configuration management is an approach to interoperability testing where tests are executed in the cloud, as the cloud can provide cost-effective access to multiple configurations of the test environments

D. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation

Answer: D

Explanation:

This answer is correct because configuration management is a process of establishing and maintaining consistency of a product??s performance, functional, and physical attributes with

its requirements, design, and operational information throughout its life. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation. This supports testing by providing traceability, consistency, and control over the test artifacts and the software under test. References: : ISTQB Glossary of Testing Terms v4.0, : ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.2

NEW QUESTION 172

.....

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](#)