

Exam Questions CTFL4

ISTQB Certified Tester Foundation Level CTFL 4.0 Exam

<https://www.2passeasy.com/dumps/CTFL4/>



NEW QUESTION 1

Which statement is true regarding confirmation testing and regression testing?

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

Answer: C

Explanation:

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

NEW QUESTION 2

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

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

Answer: C

Explanation:

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

NEW QUESTION 3

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 4

Which of the following statements about statement coverage is TRUE?

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

Answer: D

Explanation:

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

NEW QUESTION 5

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 6

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 7

The four test levels used in ISTQB syllabus are:

- * 1. Component (unit) testing
- * 2. Integration testing
- * 3. System testing
- * 4. Acceptance testing

An organization wants to do away with integration testing but otherwise follow V-model. Which of the following statements is correct?

- A. It is allowed as organizations can decide on men test levels to do depending on the context of the system under test
- B. It is allowed because integration testing is not an important test level arc! can be dispensed with.
- C. It is not allowed because integration testing is a very important test level and ignoring i: means definite poor product quality
- D. It is not allowed as organizations can't change the test levels as these are chosen on the basis of the SDLC (software development life cycle) model

Answer: D

Explanation:

The V-model is a software development life cycle model that defines four test levels that correspond to four development phases: component (unit) testing with component design, integration testing with architectural design, system testing with system requirements, and acceptance testing with user requirements. The V-model emphasizes the importance of verifying and validating each phase of development with a corresponding level of testing, and ensuring that the test objectives, test basis, and test artifacts are aligned and consistent across the test levels. Therefore, an organization that wants to follow the V-model cannot do away with integration testing, as it would break the symmetry and completeness of the V-model, and compromise the quality and reliability of the software or system under test. Integration testing is a test level that aims to test the interactions and interfaces between components or subsystems, and to detect any defects or inconsistencies that may arise from the integration of different parts of the software or system. Integration testing is essential for ensuring the functionality, performance, and compatibility of the software or system as a whole, and for identifying and resolving any integration issues early in the development process. Skipping integration testing would increase the risk of finding serious defects later in the test process, or worse, in the production environment, which would be more costly and difficult to fix, and could damage the reputation and credibility of the organization. Therefore, the correct answer is D.

The other options are incorrect because:

? A. It is not allowed as organizations can decide on the test levels to do depending on the context of the system under test. While it is true that the choice and scope of test levels may vary depending on the context of the system under test, such as the size, complexity, criticality, and risk level of the system, the organization cannot simply ignore or skip a test level that is defined and required by the chosen software development life cycle model. The organization must follow the principles and guidelines of the software development life cycle model, and ensure that the test levels are consistent and coherent with the development phases. If the organization wants to have more flexibility and adaptability in choosing the test levels, it should consider using a different software development life cycle model, such as an agile or iterative model, that allows for more dynamic and incremental testing approaches.

? B. It is not allowed because integration testing is not an important test level and can be dispensed with. This statement is false and misleading, as integration testing is a very important test level that cannot be dispensed with. Integration testing is vital for testing the interactions and interfaces between components or subsystems, and for ensuring the functionality, performance, and compatibility of the software or system as a whole. Integration testing can reveal defects or inconsistencies that may not be detected by component (unit) testing alone, such as interface errors, data flow errors, integration logic errors, or performance degradation. Integration testing can also help to verify and validate the architectural design and the integration strategy of the software or system, and to ensure that the software or system meets the specified and expected quality attributes, such as reliability, usability, security, and maintainability. Integration testing can also provide feedback and confidence to the developers and stakeholders about the progress and quality of the software or system development. Therefore, integration testing is a crucial and indispensable test level that should not be skipped or omitted.

? C. It is not allowed because integration testing is a very important test level and ignoring it means definite poor product quality. This statement is partially true, as integration testing is a very important test level that should not be ignored, and skipping it could result in poor product quality. However, this statement is too strong and absolute, as it implies that integration testing is the only factor that determines the product quality, and that ignoring it would guarantee a poor product quality. This is not necessarily the case, as there may be other factors that affect the product quality, such as the quality of the requirements, design, code, and other test levels, the effectiveness and efficiency of the test techniques and tools, the competence and experience of the developers and testers, the availability and adequacy of the resources and environment, the management and communication of the project, and the expectations and satisfaction of the customers and users. Therefore, while integration testing is a very important test level that should not be skipped, it is not the only test level that matters, and skipping it does not necessarily mean definite poor product quality, but rather a higher risk and likelihood of poor product quality.

References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.3, pages 16-18; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 38-39; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 104, page 36.

NEW QUESTION 8

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

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

Answer: C

Explanation:

? Effort estimate does not depend on the budget of the project, but rather on the scope, complexity, and quality of the software product and the testing activities¹. Budget is a constraint that may affect the feasibility and accuracy of the effort estimate, but it is not a factor that determines the effort estimate. Effort estimate is the amount of work required to complete the testing activities, measured in terms of person-hours, person-days, or person-months².

? The other options are correct because: References =

? 1 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 154

? 2 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 155

? 3 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 156

? 4 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 157

? 5 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 158

? 6 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 159

? 7 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16

? [8] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 160

? [9] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 161

NEW QUESTION 9

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 10

Which of the following statements best describe Behavior-Driven Development (BDD)?

- A. A collaborative approach that allows every stakeholder to contribute to how the software component must behave.
- B. Expresses the behavior of an application with test cases written in Given When Then format.
- C. Is used to develop code guided by automated test cases.
- D. A psychological technique in which the team's behavior in agile teams is evaluated.

Answer: A

Explanation:

Behavior-Driven Development (BDD) is a collaborative approach that enhances communication among project stakeholders, including developers, testers, and business analysts. It involves defining how software should behave through examples written in a common language understandable by all stakeholders, often using the Given-When-Then format.

NEW QUESTION 10

You are testing the latest version of an air-traffic control system prior to production deployment using exploratory testing. After following an unusual sequence of input steps, the system crashes. After the crash, you document a defect report with the following information:

- Title: System crashes unexpectedly during input.
 - Brief summary: System crashes when an unusual sequence of inputs is used.
 - Version: V1.001
 - Test: Exploratory testing prior to production deployment
 - Priority: Urgent
 - Risk: High
 - References: Screenshot of crashed application
- What critical information is missing from this report?

- A. Conclusions, recommendations, and approvals.
- B. Change history.
- C. Description of the defect to enable reproduction.
- D. Status of defect

Answer: C

Explanation:

The critical information missing from the defect report is a detailed description of the defect to enable reproduction. A clear and concise description of the steps taken to reproduce the defect is essential for developers to understand the context and to be able to replicate the issue in their environment. Without this information, it can be challenging to diagnose and fix the defect. The ISTQB CTFL syllabus emphasizes the importance of providing all necessary details in a defect report to facilitate effective communication and resolution.

References: ISTQB CTFL Syllabus, Section 5.5, "Defect Management."

NEW QUESTION 12

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

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

Answer: A

Explanation:

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

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

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

? Gather the metrics that are used to guide the test project: This is a test task that usually occurs during test monitoring and control, which is the activity of tracking and reviewing the test progress, status, and quality, and taking corrective actions when necessary. During this activity, the testers can gather the metrics, which are the measurements of the testing process, such as the test coverage, the defect density, the test effort, the test duration, etc., that are used to guide the test project, such as planning, estimating, scheduling, reporting, or improving the testing process. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

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

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

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

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.4, Test Implementation¹

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.5, Test Execution¹

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.6, Test Closure¹

? ISTQB® Glossary of Testing Terms v4.0, Test Environment, Test Condition, Test Data, Test Tool, Test Interface, Failure, Root Cause Analysis, Defect Prevention, Defect Analysis, Testware, Regression Testing, Maintenance Testing, Reuse Testing, Test Coverage, Defect Density, Test Effort, Test Duration²

NEW QUESTION 15

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

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

Answer: B

Explanation:

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

NEW QUESTION 20

What is test oracle?

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

Answer: C

Explanation:

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

NEW QUESTION 22

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

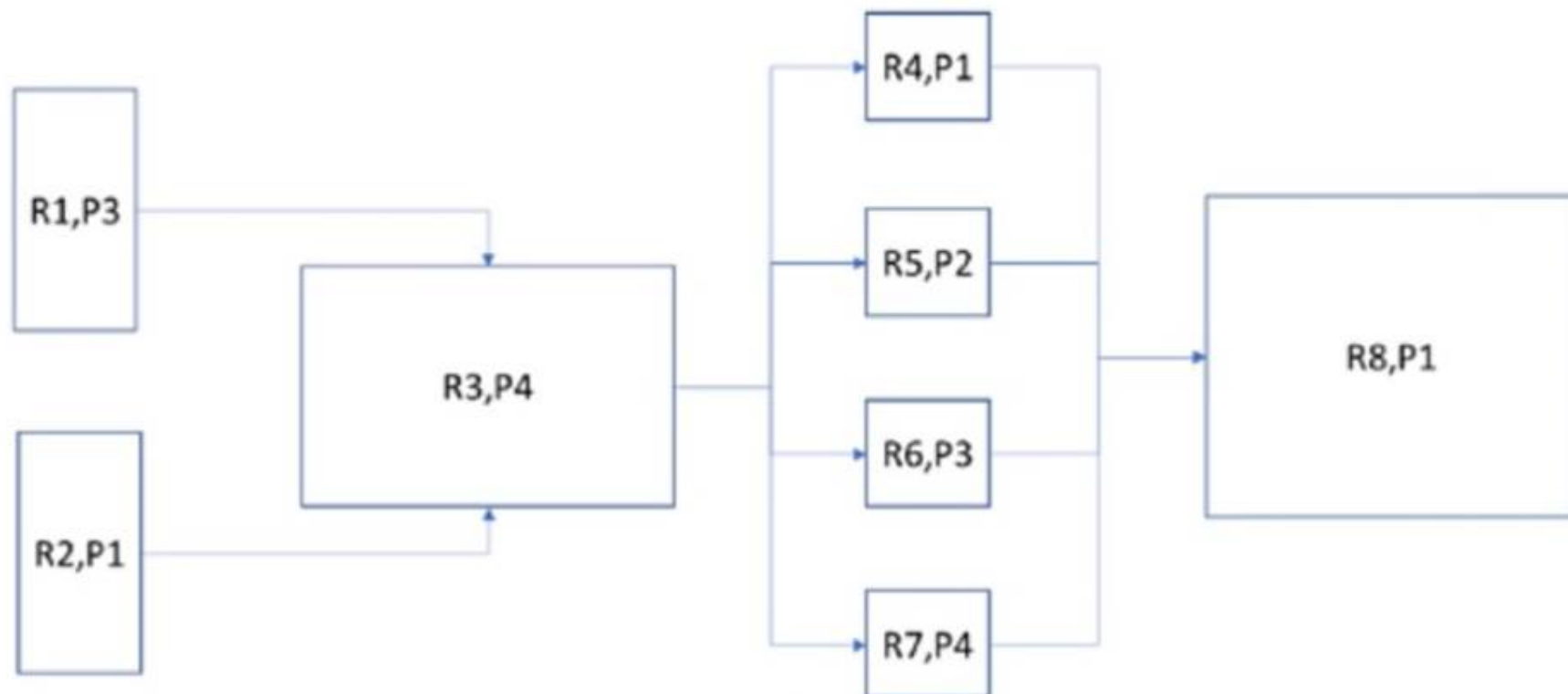
Answer: C

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 27

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



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

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

Answer: D

Explanation:

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

NEW QUESTION 30

Which of the following statements is TRUE'?

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

Answer: D

Explanation:

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

NEW QUESTION 34

Which of the following statements refers to a good testing practice that applies to all software development lifecycles?

- A. Each test level should have its own specific test objectives that should be consistent with the software development lifecycle phase or type of activities it addresses.
- B. Test analysis and design for any test levels should begin as soon as coding is complete, and all system components are available for testing
- C. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- D. All the tests should be automated and run as part of the continuous integration process with every software change

Answer: A

Explanation:

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

NEW QUESTION 37

For each test case to be executed, the following table specifies its dependencies and the required configuration of the test environment for running such test case:

| Test Case | Dependencies | Configuration |
|-----------|--------------|---------------|
| TC1 | | CONF2 |
| TC2 | TC4 | CONF2 |
| TC3 | TC4 | CONF1 |
| TC4 | | CONF1 |
| TC5 | TC1 | CONF2 |

Assume that CONF1 is the initial configuration of the test environment Based on this assumption, which of the following is a test execution schedule that is compatible with the specified dependencies and allows minimising the number of switches between the different configurations of the test environment^

- A. TC4, TC3, TC2, TC1, TC5.
- B. TC1, TC5, TC4, TC3, TC2
- C. TC4, TC3, TC2, TC5, TC1.
- D. TC4, TC1, TC5, TC2, TC3

Answer: A

Explanation:

To determine the optimal test execution schedule that minimizes the number of configuration switches and respects the dependencies, we start with the initial configuration, CONF1.

? TC4: It has no dependencies and runs on CONF1 (initial configuration).

? TC3: Depends on TC4 and runs on CONF1. Since TC4 is already executed, we can proceed with TC3.

? TC2: Depends on TC4 and runs on CONF2. We switch to CONF2 after TC3.

? TC1: No dependencies and runs on CONF2. Since we are already in CONF2, we can execute TC1 next.

? TC5: Depends on TC1 and runs on CONF2. Since TC1 is already executed, we can proceed with TC5 without additional configuration switches.

By following this sequence (TC4, TC3, TC2, TC1, TC5), we respect the dependencies and minimize the number of configuration switches

NEW QUESTION 41

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 44

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

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

Answer: B

Explanation:

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

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

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

NEW QUESTION 46

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

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

Answer: C

Explanation:

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

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

NEW QUESTION 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

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

- A. Understanding, implementing, and updating the test strategy.

- B. Ensuring the proper use of testing tools.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 58

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 59

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 60

The fact that defects are usually not evenly distributed among the various modules that make up a software application, but rather their distribution tend to reflect the Pareto principle:

- A. is a false myth
- B. is expressed by the testing principle referred to as 'Tests wear out'
- C. is expressed by the testing principle referred to as 'Defects cluster together'
- D. is expressed by the testing principle referred to as 'Bug prediction'

Answer: C

Explanation:

The fact that defects are usually not evenly distributed among the various modules that make up a software application, but rather their distribution tend to reflect the Pareto principle, is expressed by the testing principle referred to as ??Defects cluster together??. This principle states that a small number of modules contain most of the defects detected, or that a small number of causes are responsible for most of the defects. This principle can be used to guide the test analysis and design activities, by prioritizing the testing of the most critical or risky modules, or by applying more rigorous test techniques to them. Therefore, option C is the correct answer.

References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 1.2.1, page 11; ISTQB® Glossary v4.02, page 16.

NEW QUESTION 65

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 70

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

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

Answer: D

Explanation:

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

NEW QUESTION 75

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 auditabl
- 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 76

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

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

Answer: A

Explanation:

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

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

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

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

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

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

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

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

NEW QUESTION 79

Consider the following examples of risks identified in different software development projects:

[I]. The contrast color ratio for both normal text and large text of a website does not comply with the applicable accessibility guidelines, making it difficult for many users to read the content on the pages

[II]. A development vendor fails to deliver their software system on time, causing significant delays to system integration testing activities that have been planned as part of a development project for a system of systems

[III]. People in the test team do not have sufficient skills to automate tests at the test levels required by the test automation strategy which does not allow production of an effective regression test suite

[IV]. In a web application, data from untrusted sources is not subject to proper input validation, making the application vulnerable to several security attacks. Which of the following statements is true?

- A. [I] and [III] are product risks; [II] and [IV] are project risks
- B. [I] and [IV] are product risk
- C. [II] and [III] are project risks
- D. [II], [III] and [IV] are product risks; [I] is a project risk
- E. [IV] is a product risk; [I], [II] and [III] are project risks

Answer: B

Explanation:

This answer is correct because product risks are risks that affect the quality of the software product, such as defects, failures, or non-compliance with requirements or standards. Project risks are risks that affect the project's schedule, budget, resources, or scope, such as delays, cost overruns, skill gaps, or scope changes. In this case, [I] and [IV] are product risks, as they relate to the accessibility and security of the software product, which are quality attributes. [II] and [III] are project risks, as they relate to the delivery time and the test automation skills of the test team, which are project factors. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.1.1.1

NEW QUESTION 84

Select which of the following statements describe the key principles of software testing?

- A. Testing shows the presence of defects, not their absence.i
- B. Testing everything is possible.ii
- C. Early testing is more expensive and is a waste of time.i
- D. Defects cluster together.
- E. Testing is context dependent.v
- F. Beware of the pesticide paradox.vi
- G. Absence of errors is a fallacy
- H. Select the correct Answer:
- I. i, iv, v, vi and vii
- J. I, ii,
- K. vi and vii
- L. ii
- M. iv,
- N. vi and vii
- O. ii, iii, iv, v and vi

Answer: A

Explanation:

The key principles of software testing include: i. Testing shows the presence of defects, not their absence. iv. Defects cluster together. v. Testing is context dependent. vi. Beware of the pesticide paradox. vii. Absence of errors is a fallacy. These principles highlight the importance of recognizing the limitations and context of testing, as well as the potential for repeated tests to become less effective.

NEW QUESTION 88

Mark the correct sentences:

- * Defects are a result of environmental conditions and are also referred to as "Failures"
- * A human mistake may produce a defect
- * A system may totally fail to operate correctly when a failure exists in it
- * When a defect exists in a system it may result in a failure
- * Defects occur only as a result of technology changes

- A. II, IV
- B. I, II
- C. IV, V
- D. II, III, IV

Answer: A

Explanation:

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

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

? The other three statements are incorrect, namely: References:

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

NEW QUESTION 90

Confirmation testing is performed after:

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

Answer: D

Explanation:

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

a result of changes or fixes. Therefore, option D is the correct answer.

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

NEW QUESTION 92

Which one of the following is a typical entry criteria for testing?

- A. Planned tests have been executed.
- B. Availability of testable requirements.
- C. The number of unresolved defects is within an agreed limit.
- D. The number of estimated remaining defects is sufficiently low.

Answer: B

Explanation:

A typical entry criterion for testing is the availability of testable requirements. Testable requirements provide a basis for designing and executing test cases. Without clear and testable requirements, it is challenging to determine what needs to be tested and to create effective test cases. Entry criteria ensure that the necessary preconditions are met before testing begins, which helps in conducting efficient and effective testing. References: ISTQB CTFL Syllabus, Section 5.1.3, "Entry and Exit Criteria."

NEW QUESTION 94

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 96

Select the roles required in a formal review:

- A. Author, Management, Facilitator, Review Leader, Reviewers, Scribe
- B. Author, Tester
- C. Facilitator
- D. Review Leader
- E. Reviewer
- F. Scribe
- G. Author, Business analyst
- H. Facilitator, Review Leader
- I. Reviewer
- J. Scribe
- K. Author
- L. Developer, Facilitator
- M. Review Leader
- N. Reviewer
- O. Scribe

Answer: A

Explanation:

In a formal review, the roles involved typically include the author, management, facilitator (also known as moderator), review leader, reviewers, and scribe. Each role has specific responsibilities to ensure the effectiveness and efficiency of the review process:

- ? The author creates and refines the work product being reviewed.
- ? Management allocates resources and supports the review process.
- ? The facilitator manages the review meeting, ensuring it proceeds smoothly.
- ? The review leader plans the review and ensures it meets its objectives.
- ? Reviewers examine the work product to identify defects.
- ? The scribe records issues raised during the review meeting.

NEW QUESTION 101

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 102

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 106

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 107

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 109

You are an experienced tester on a project with incomplete requirements and under pressure to deploy. What type of testing should you do?

- A. Decision-based testing.
- B. Checklist-based testing.
- C. Error guessing.
- D. Exploratory testing.

Answer: D

Explanation:

When working on a project with incomplete requirements and under pressure to deploy, exploratory testing is particularly suitable. This type of testing allows testers to use their expertise and intuition to explore the system's functionality and identify defects without needing detailed specifications. Exploratory testing is flexible and can quickly adapt to changes and gaps in the requirements.

NEW QUESTION 111

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

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

| Conditions | | | | |
|------------------------------|--------|----------|--------|----------|
| Preferred Guest Card holder | YES | YES | NO | NO |
| Room Type | Silver | Platinum | Silver | Platinum |
| Actions | | | | |
| Offer upgrade to Gold Luxury | YES | NO | NO | NO |
| Offer upgrade to Silver | N/A | YES | N/A | NO |

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

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

Answer: C

Explanation:

According to the decision table in the image, a Preferred Guest Card holder with a Silver room is eligible for an upgrade to Gold Luxury (YES), while a non-Preferred Guest Card holder, regardless of room type, is not eligible for any upgrade (NO).

Therefore, Customer A (a Preferred Guest Card holder with a Silver room) would be offered an upgrade to Gold Luxury, and Customer B (a non-Preferred Guest Card holder with a Platinum room) would not be offered any upgrade. References = The answer is derived directly from the decision table provided in the image; specific ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents are not referenced.

NEW QUESTION 115

Which of the following statements about exploratory testing is true?

- A. Exploratory testing is an experience-based test technique in which testers explore the requirements specification to detect non testable requirements
- B. When exploratory testing is conducted following a session-based approach, the issues detected by the testers can be documented in session sheets
- C. Exploratory testing is an experience-based test technique used by testers during informal code reviews to find defects by exploring the source code
- D. In exploratory testing, testers usually produce scripted tests and establish bidirectional traceability between these tests and the items of the test basis

Answer: B

Explanation:

Exploratory testing is an experience-based test technique in which testers dynamically design and execute tests based on their knowledge, intuition, and learning of the software system, without following predefined test scripts or test cases. Exploratory testing can be conducted following a session-based approach, which is a structured way of managing and measuring exploratory testing. In a session-based approach, the testers perform uninterrupted test sessions, usually lasting between 60 and 120 minutes, with a specific charter or goal, and document the issues detected, the test coverage achieved, and the time spent in session sheets. Session sheets are records of the test activities, results, and observations during a test session, which can be used for reporting, debriefing, and learning purposes. The other statements are false, because:

? Exploratory testing is not a test technique in which testers explore the requirements specification to detect non testable requirements, but rather a test technique in which testers explore the software system to detect functional and non-functional defects, as well as to learn new information, risks, or opportunities. Non testable requirements are requirements that are ambiguous, incomplete, inconsistent, or not verifiable, which can affect the quality and effectiveness of the testing process. Non testable requirements can be detected by applying static testing techniques, such as reviews or inspections, to the requirements specification, before the software system is developed or tested.

? Exploratory testing is not a test technique used by testers during informal code reviews to find defects by exploring the source code, but rather a test technique used by testers during dynamic testing to find defects by exploring the behavior and performance of the software system, without examining the source code. Informal code reviews are static testing techniques, in which the source code is analyzed by one or more reviewers, without following a formal process or using a checklist, to identify defects, violations, or improvements. Informal code reviews are usually performed by developers or peers, not by testers.

? In exploratory testing, testers usually do not produce scripted tests and establish bidirectional traceability between these tests and the items of the test basis, but rather produce unscripted tests and adapt them based on the feedback and the findings of the testing process. Scripted tests are tests that are designed and documented in advance, with predefined inputs, outputs, and expected results, and are executed according to a test plan or a test procedure.

Bidirectional traceability is the ability to trace both forward and backward the relationships between the items of the test basis, such as the requirements, the design, the risks, etc., and the test artifacts, such as the test cases, the test results, the defects, etc. Scripted tests and bidirectional traceability are usually associated with more formal and structured testing approaches, such as specification-based or structure-based test techniques, not with exploratory

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

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.3,

Experience-based Test Design Techniques¹

? ISTQB® Glossary of Testing Terms v4.0, Exploratory Testing, Session-based Testing, Session Sheet, Non Testable Requirement, Static Testing, Informal Review, Dynamic Testing, Scripted Testing, Bidirectional Traceability²

NEW QUESTION 117

Which of the following statements about the shift-left approach is true?

- A. Shift-left in testing can be implemented only in Agile/DevOps frameworks, as it relies completely on automated testing activities performed within a continuous integration process
- B. Performance testing performed during component testing, is a form of shift-left in testing that avoids planning and executing costly end-to-end testing at the system test level in a production-like environment
- C. Shift-left in testing can be implemented in several ways to find functional defects early in the lifecycle, but it cannot be relied upon to find defects associated with non-functional characteristics
- D. Continuous integration supports shift-left in testing as it can reduce the time between the introduction of a defect and its detection, thereby reducing the cost to fix it

Answer: D

Explanation:

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

NEW QUESTION 118

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 121

Exploratory testing is an experience-based test technique

- A. Where a developer and a tester work together on the same workstation while the developer actively writes code, the tester explores the code to find defects.
- B. That can be organised into sessions guided by test charters outlining test objectives that will guide the testers' exploration
- C. Where a team of testers explores all possible test techniques in order to determine the most suitable combination of these techniques to apply for a test project.
- D. That aims at finding defects by designing tests that exercise all possible combinations of input values and preconditions

Answer: B

Explanation:

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

NEW QUESTION 126

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 128

Which of the following statements about the shift-left approach is FALSE?

- A. The shift-left approach can only be implemented with test automation
- B. The shift-left approach in testing is compatible with DevOps practices.

- C. The shift-left approach can involve security vulnerabilities
- D. The shift-left approach can be supported by static analysis tools.

Answer: A

Explanation:

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

NEW QUESTION 131

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 135

Which of the following statements is true?

- A. In Agile software development, work product documentation tends to be lightweight and manual tests tend to be often unscripted as they are often produced using experience-based test techniques
- B. Sequential development models impose the use of systematic test techniques and do not allow the use of experience-based test techniques
- C. In Agile software development, the first iterations are exclusively dedicated to testing activities, as testing will be used to drive development, which will be performed in the subsequent iterations
- D. Both in Agile software development and in sequential development models, such as the V-model, test levels tend to overlap since they do not usually have defined entry and exit criteria

Answer: A

Explanation:

This answer is correct because in Agile software development, work product documentation, such as user stories, acceptance criteria, or test cases, tends to be lightweight and concise, as the focus is on working software and frequent communication rather than comprehensive documentation. Manual tests tend to be often unscripted, as they are often produced using experience-based test techniques, such as error guessing or exploratory testing, which rely on the tester's skills, knowledge, and creativity to find defects and provide feedback. References: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.2, Section 3.2.1.2

NEW QUESTION 138

Consider the following user story about an e-commerce website's registration feature that only allows registered users to make purchases ; As a new user, I want to register to the website, so that I can start shopping online"

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

- [a] The registration form consists of the following fields: username, email address, first name, last name, date of birth, password and repeat password.
- [b] To submit the registration request, the new user must fill in all the fields of the registration form with valid values and must agree to the terms and conditions.
- [c] To be valid, the email address must not be provided by free online mail services that allow to create disposable email addresses. A dedicated error message must be presented to inform the new user when an invalid address is entered.
- [d] To be valid, the first name and last name must contain only alphabetic characters and must be between 2 and 80 characters long A dedicated error message must be presented to inform the new user when an invalid first name and/or the last name is entered
- [e] After submitting the registration request, the new user must receive an e-mail containing the confirmation link to the e-mail address specified in the registration form

Based only on the given information, which of the following ATDD tests is MOST LIKELY to be written first?

- A. The new user enters valid values in the fields of the registration form, except for the email address, where he/she enters an e-mail address provided by a free online mail service that allow to create disposable email address
- B. Then he/she is informed by the website about this issue.
- C. The new user enters valid values in the fields of the registration form, except for the first name, where he/she enters a first name with 10 characters that contains a number
- D. Then he/she is informed by the website about this issue.
- E. The user accesses the website with a username and password, and successfully places a purchase order for five items, paying by Mastercard credit card
- F. The new user enters valid values in all the fields of the registration form, confirms to accept all the terms and conditions, submits the registration request and then receives an e-mail containing the confirmation link to the e-mail address specified in the registration form

Answer: D

Explanation:

Acceptance Test-Driven Development (ATDD) tests focus on verifying whether the system meets the specified acceptance criteria. The most critical path to test first would be the scenario where everything is done correctly (happy path), ensuring the basic functionality works as expected.

? The new user provides all valid data.

? This ensures the registration form works and the user receives a confirmation email.

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

Reference: ISTQB CTFL Syllabus V4.0, Chapter 4.5.3, Acceptance Test-Driven Development (ATDD).

NEW QUESTION 140

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual CTFL4 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the CTFL4 Product From:

<https://www.2passeasy.com/dumps/CTFL4/>

Money Back Guarantee

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 FirstTry
- * CTFL4 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year