



**BCS**

## **Exam Questions CTFL4**

ISTQB Certified Tester Foundation Level CTFL 4.0 Exam

## About Exambible

*[Your Partner of IT Exam](#)*

## Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

Our guarantee that you will pass the exam.

### \* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

### 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

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

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

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

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

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

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

**Answer:** B

#### Explanation:

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

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

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

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

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

### NEW QUESTION 3

A calculator software is used to calculate the result for 5+6. The user noticed that the result given is 6.

This is an example of;

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

**Answer:** D

#### Explanation:

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

The other options are incorrect because:

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

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

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

measurement error in the observation, or a deviation error in the condition.

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

#### NEW QUESTION 4

Which of the following is not an example of a typical content of a test completion report for a test project?

- A. The additional effort spent on test execution compared to what was planned
- B. The unexpected test environment downtime that resulted in slower test execution
- C. The residual risk level if a risk-based test approach was adopted
- D. The test procedures of all test cases that have been executed

**Answer: D**

#### Explanation:

This answer is correct because the test procedures of all test cases that have been executed are not a typical content of a test completion report for a test project. A test completion report is a document that summarizes the test activities and results at the end of a test project. It usually includes information such as the test objectives, scope, approach, resources, schedule, results, deviations, issues, risks, lessons learned, and recommendations for improvement. The test procedures of all test cases that have been executed are part of the test documentation, but they are not relevant for the test completion report, as they do not provide a high-level overview of the test project outcomes and performance. References: ISTQB Foundation Level Syllabus v4.0, Section 2.5.3.2

#### NEW QUESTION 5

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

- A. Static testing is better suited than dynamic testing for highlighting issues that could indicate inappropriate code modularisation.
- B. Dynamic testing can only be applied to executable work products, while static testing can only be applied to non-executable work products.
- C. Both dynamic testing and static testing cause failures, but failures caused by static testing are usually easier and cheaper to analyse.
- D. Security vulnerabilities can only be detected when the software is being executed, and thus they can only be detected through dynamic testing, not through static testing

**Answer: A**

#### Explanation:

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

#### NEW QUESTION 6

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

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

**Answer: A**

#### Explanation:

Test-driven development is not an example of a typical generic skill required for testing, but rather an example of a specific technical skill or a development practice that may or may not be relevant for testing, depending on the context and the objectives of the testing activities. Test-driven development is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design, and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a skill that is generally expected or needed for testers, especially for testers who are not involved in unit testing or who do not have access to the source code. The other options are examples of typical generic skills required for testing, which are skills that are applicable and beneficial for testing in any context or situation, regardless of the specific testing techniques, tools, or methods used. The typical generic skills required for testing include:

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

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

? Possess the necessary social skills that support effective teamwork: These are skills that involve the ability to interact, cooperate, and coordinate with other people involved in or affected by the testing activities, such as the test manager, the test team, the project manager, the developers, the customers, the users, etc. These skills can include communication, negotiation, leadership, motivation, feedback, conflict resolution, etc. These skills can enhance the quality, the productivity, and the satisfaction of the testing process, as well as to foster a positive and constructive testing culture. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle

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

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

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

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

#### NEW QUESTION 7

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

[1]. It may not be possible to generate the expected workloads to run performance tests, due to the poor hardware equipment of the machines (load injectors) that should generate these workloads.

[ii]. A user's session on a web application is not invalidated after a certain period of inactivity (configured by the system administrator) of the user,  
[iii]. The test team will not have an adequate requirements specification (since many requirements will still be missing) by the time test design and analysis activities should begin according to the test plan.  
[IV]. Following a failure, the system is unable to continue to maintain its pre-failure operation and some data becomes corrupted.  
Which of the following statements is TRUE?

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

**Answer:** A

**Explanation:**

In software testing, risks are categorized into product risks and project risks. Product risks are associated with the potential of a product to fail in meeting its quality criteria. Project risks are related to potential issues that could affect the project's ability to deliver a product.

? [i] is a project risk because it concerns the availability and adequacy of hardware resources for performance testing.

? [ii] is a product risk because it pertains to a security and functionality issue within the web application.

? [iii] is a project risk because it involves the availability of necessary requirements documentation for the testing process.

? [iv] is a product risk because it relates to the system's functionality and data integrity after a failure.

Thus, statement A correctly classifies [ii] and [iv] as product risks and [i] and [iii] as project risks.

**NEW QUESTION 8**

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 9**

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 10**

A financial institution is to implement a system that calculates the interest rates paid on investment accounts based on the sum invested.

You are responsible for testing the system and decide to use equivalence partitioning and boundary value analysis to design test cases. The requirements describe the following expectations:

Investment range| Interest rate  
R500 to R10 000| 10%

R10 001 to R50 000| 11%  
R50 001 to R100 000| 12%  
R100 001 to R500 000| 13%

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

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

**Answer:** B

**Explanation:**

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

? R500 to R10,000 (10%)

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

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

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

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



#### NEW QUESTION 10

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 12

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 activities<sup>1</sup>. 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-months<sup>2</sup>.

? 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 15

A document describes the test procedures that have been derived for the identified test sets Among other things, the order in which the test cases in the

corresponding test set are to be executed according to the dependencies described by preconditions and postconditions is specified This document is a typical work product produced as part of:

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

**Answer: C**

**Explanation:**

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

**NEW QUESTION 18**

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 23**

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

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

**Answer: D**

**Explanation:**

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

defects. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level

Syllabus v4.0, Section 2.3.2.2

**NEW QUESTION 26**

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

- A. Can't be determined
- B. Covering all transitions at least once
- C. Covering only start and end states
- D. Covering all states at least once

**Answer: B**

**Explanation:**

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

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

Option C is the lowest coverage criterion, because it only tests the initial and final states of the system or component, without checking the intermediate states or transitions.

Option A is incorrect, because the coverage criteria for state transition based test cases can be determined and compared based on the

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

**NEW QUESTION 27**

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 29**

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 32**

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 37**

A test status report SHOULD:



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

**Answer:** A

**Explanation:**

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

Option B describes an activity related to test completion rather than ongoing status reporting. Option C is incorrect because the structure and contents of the report may vary based on the audience's needs. Option D, while important, is not the primary purpose of a test status report, which focuses more on the current status and impediments.

**NEW QUESTION 40**

A program is used to control a manufacturing line (turn machines on and off. start and stop conveyer belts, add raw materials to the flow. etc.). Not all actions are possible at all times. For example, there are certain manufacturing stages that cannot be stopped - unless there is an emergency. A tester attempts to evaluate if all such cases (where a specific action is not allowed) are covered by the tests.

Which coverage metric will provide the needed information for this analysis?

- A. Code coverage
- B. Data flow coverage
- C. Statement coverage
- D. Branch Coverage

**Answer:** D

**Explanation:**

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

Branch coverage is suitable for testing the cases where a specific action is not allowed, because it can verify that the test cases cover all the possible outcomes of the conditions that determine the action. For example, if the program has a condition that checks if the manufacturing stage can be stopped, then branch coverage can ensure that the test cases cover both the cases where the stage can be stopped and where it cannot be stopped. This way, branch coverage can help identify any missing or incorrect branches that may lead to undesired or unsafe actions.

The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types of coverage metrics, such as statement coverage, branch coverage, data flow coverage, etc. Code coverage does not specify which type of coverage metric is used for the analysis. Data flow coverage is a type of structural coverage metric that measures the percentage of data flow paths that are executed by the test cases. A data flow path is a sequence of statements that define, use, or kill a variable. Data flow coverage is useful for testing the correctness and completeness of the data manipulation in the software, but not for testing the conditions that determine the actions. Statement coverage is a type of structural coverage metric that measures the percentage of statements or lines of code that are executed by the test cases. Statement coverage ensures that each statement is executed at least once by the test cases, but it does not reveal the behavior of the software under different scenarios. Statement coverage is a weaker criterion than branch coverage, because it does not account for the branches or decision outcomes in the code. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

**NEW QUESTION 42**

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

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

**Answer:** A

**Explanation:**

Reference:ISTQB CTFL Syllabus V4.0, Section 5.3.2

**NEW QUESTION 45**

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 46

Test automation allows you to:

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

**Answer: B**

#### Explanation:

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

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

? ISTQB® Glossary of Testing Terms v4.0, Test Automation2

#### NEW QUESTION 51

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 54

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

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

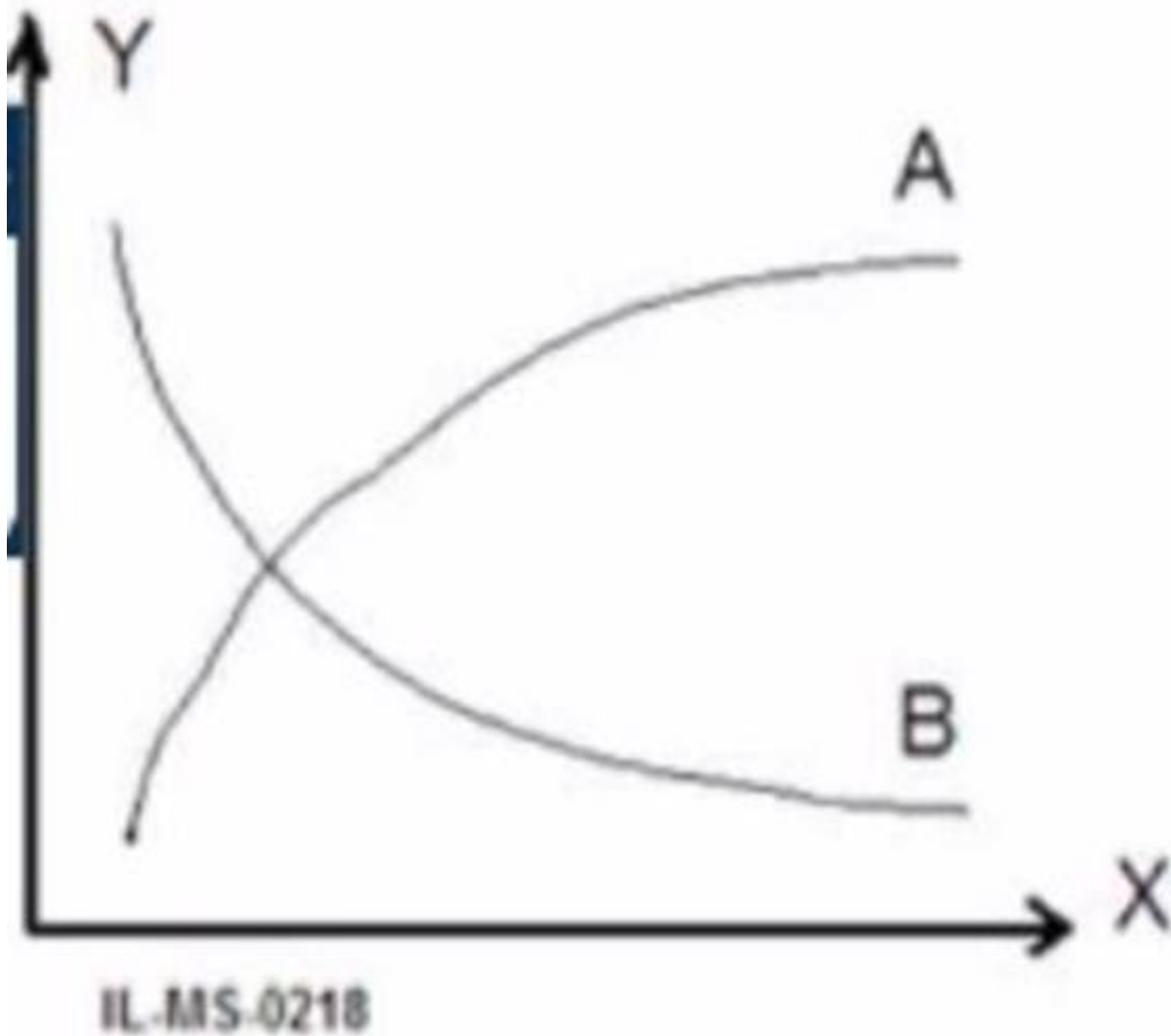
**Answer: C**

#### Explanation:

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

#### NEW QUESTION 56

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



A)

X - Time

Y - Cost

A - Cost of test (per week)

B - Cost of finding a single bug (per week)

B)

X - Time

Y - Number of defects

A - Number of open defects

B - Number of closed defects

C)

X - Time

Y - Percent

A - % of functional tests in the test suite

B - % of non-functional tests in the test suite

D)

X - Time

Y - Count

A - Total number of executed tests

B - Number of open bugs

A. Option A

B. Option B

C. Option C

D. Option D

**Answer: D**

**Explanation:**

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

? X-axis represents Time

? Y-axis represents Count

? Line A represents Number of open bugs

? Line B represents Total number of executed tests

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

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

**NEW QUESTION 57**

Which of the following statements about checklist-based testing is TRUE?

- A. Checklist-based testing is a technique for managing the review meeting that can be applied in those reviews where the use of checklists is mandatory, as is often the case in formal reviews.
- B. Checklist-based testing is a review technique that can be used in a formal review process where reviewers, during individual review, try to detect issues within the work product based on a checklist
- C. In checklist-based testing, using checklists at a high level of detail is more likely to produce test cases that are easier to reproduce than those using checklists at a low level of detail
- D. Checklists used in checklist-based testing should be reviewed periodically for updates as, over time, test cases designed using the same checklist may become less effective at finding defects.

**Answer:** D

**Explanation:**

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

**NEW QUESTION 62**

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

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

Which of the following statements is TRUE?

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

**Answer:** B

**Explanation:**

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

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

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

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

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

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

**NEW QUESTION 64**

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

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

**Answer:** C

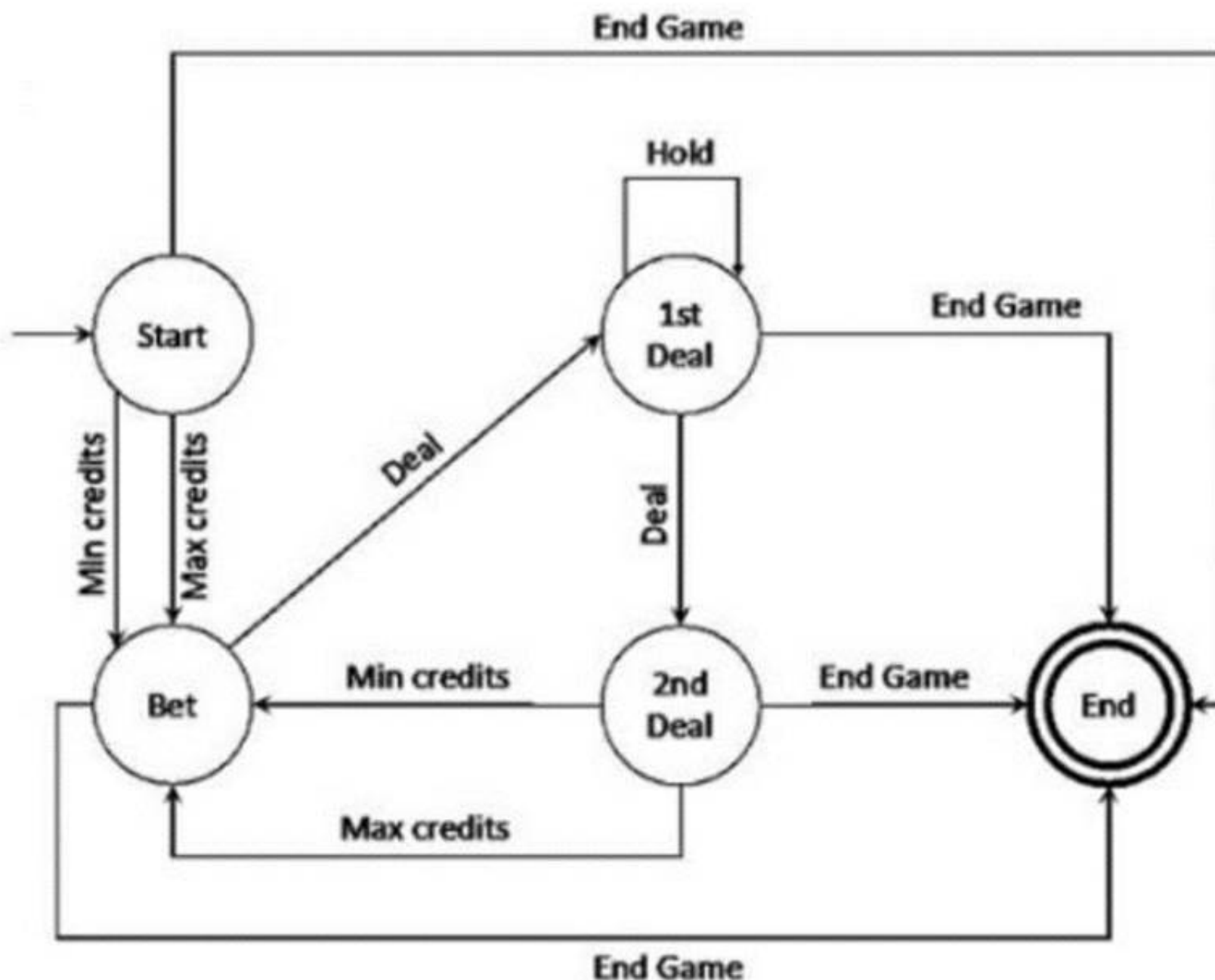
**Explanation:**

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

**NEW QUESTION 68**

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





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

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

**Answer: D**

**Explanation:**

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

? Start -> Bet -> End

? Start -> Deal -> End

? Start -> 1st Deal -> End

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

**NEW QUESTION 70**

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

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

**Answer: C**

**Explanation:**

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

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

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

References:

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

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

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

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

**NEW QUESTION 75**

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 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 81**

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 84**

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 87**

Which of the following statements is incorrect regarding the involvement of testers in the software development lifecycle (SDLC)?

- A. Testers should contribute to all activities in the SDLC and participate in design discussions.
- B. Testers should be involved from the beginning of the SDLC to increase understanding of design decisions and detect defects early.
- C. Testers should only be involved during the testing phase.
- D. Testers' involvement is essential; developers find it difficult to be objective.

**Answer:** C

**Explanation:**

Involving testers only during the testing phase is incorrect as per the ISTQB CTFL syllabus. Effective involvement of testers is crucial throughout the entire software development lifecycle (SDLC). This includes early stages such as requirement analysis and design, which allows testers to understand the design decisions and detect defects early. Early involvement helps in better understanding the project and ensures that quality is built into the product from the beginning. Furthermore, the ISTQB syllabus emphasizes the importance of testers contributing to all activities in the SDLC, including design discussions, to enhance defect detection and prevention.

References: ISTQB CTFL Syllabus, Section 2.1.1, "The Influence of Development Models on Testing" and Section 1.1.1, "Test Objectives."

**NEW QUESTION 91**

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 92**

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

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

**Answer:** D

**Explanation:**

Reference: ISTQB CTFL Syllabus V4.0, Section 5.4

**NEW QUESTION 96**

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

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

**Answer:** B

**Explanation:**

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

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

**NEW QUESTION 98**

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

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

**Answer:** A

**Explanation:**

Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential bugs<sup>12</sup>. However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)<sup>3</sup>. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures. MTBF can only be estimated by dynamic testing, which involves executing the program under various conditions and collecting data on its failures<sup>4</sup>. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools. The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can be identified by static analysis tools. Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false<sup>5</sup>. Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables<sup>6</sup>. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption. References = What Is Static Analysis? Static Code Analysis Tools - Perforce Software, How Static Code Analysis Works | Perforce, Static Code Analysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs - CareerRide, ISTQB\_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

**NEW QUESTION 103**

For each of the test cases to be executed, the following table specifies the priority order and dependencies on other test cases

Test Case	Priority	Logical Dependencies
TC1	Low	TC5
TC2	High	TC3
TC3	High	TC4
TC4	High	-
TC5	Low	TC2
TC6	Medium	-

Which of the following test execution schedules is compatible with the logical dependencies and allows executing the test cases in priority order?

- A. TC4, TC3, TC2, TC6, TC5, TC1
- B. TC4, TC6, TC3, TC2, TC5, TC1
- C. TC3, TC5, TC6, TC1, TC4, TC3
- D. TC4, TC3, TC2, TC6, TC1, TC5

**Answer:** D

**Explanation:**

This answer is correct because it follows the logical dependencies and allows executing the test cases in priority order. TC4, TC3, and TC2 are executed first because they have the highest priority. TC6 is executed next because it has a logical dependency on TC2. TC1 is executed next because it has a logical dependency on TC5. Finally, TC5 is executed last because it has the lowest priority. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents

**NEW QUESTION 104**

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

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

**Answer:** D

**Explanation:**

The test leader is the person who is responsible for planning, monitoring, and controlling the test activities and resources in a test project. The test leader should have the best knowledge of the test objectives, scope, risks, resources, schedule, and quality criteria. The test leader should also be aware of the test automation criteria, such as the execution frequency, the test support, the team education, the roles and responsibilities, and the devs and testers collaboration<sup>1</sup>. Based on these factors, the test leader can decide which tests are suitable for automation and which are not, and prioritize them accordingly. The test leader can also coordinate with the test automation engineers, the developers, and the stakeholders to ensure the alignment of the test automation strategy with the test project goals and expectations. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 2, Section 2.3.1, Page 152; ISTQB Glossary of



#### NEW QUESTION 108

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 111

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 113

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 115

Which of the following statements about how different types of test tools support testers is true?

- A. The support offered by a test data preparation tool is often leveraged by testers to run automated regression test suites
- B. The support offered by a performance testing tool is often leveraged by testers to run load tests
- C. The support offered by a bug prediction tool is often used by testers to track the bugs they found
- D. The support offered by a continuous integration tool is often leveraged by testers to automatically generate test cases from a model

**Answer: B**

#### Explanation:

The support offered by a performance testing tool is often leveraged by testers to run load tests, which are tests that simulate a large number of concurrent users or transactions on the system under test, in order to measure its performance, reliability, and scalability. Performance testing tools can help testers to generate realistic workloads, monitor system behavior, collect and analyze performance metrics, and identify performance bottlenecks. The other statements are false, because:

? A test data preparation tool is a tool that helps testers to create, manage, and manipulate test data, which are the inputs and outputs of test cases. Test data preparation tools are not directly related to running automated regression test suites, which are test suites that verify that the system still works as expected after changes or modifications. Regression test suites are usually executed by test execution tools, which are tools that can automatically run test cases and compare actual results with expected results.

? A bug prediction tool is a tool that uses machine learning or statistical techniques to predict the likelihood of defects in a software system, based on various factors such as code complexity, code churn, code coverage, code smells, etc. Bug prediction tools are not used by testers to track the bugs they found, which are the actual defects that have been detected and reported during testing. Bugs are usually tracked by defect management tools, which are tools that help testers to record, monitor, analyze, and resolve defects.

? A continuous integration tool is a tool that enables the integration of code changes from multiple developers into a shared repository, and the execution of automated builds and tests, in order to ensure the quality and consistency of the software system. Continuous integration tools are not used by testers to

automatically generate test cases from a model, which are test cases that are derived from a representation of the system under test, such as a state diagram, a decision table, a use case, etc. Test cases can be automatically generated by test design tools, which are tools that support the implementation and maintenance of test cases, based on test design specifications or test models. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:  
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 3.4.1, Types of Test Tools  
? ISTQB® Glossary of Testing Terms v4.0, Performance Testing Tool, Test Data Preparation Tool, Bug Prediction Tool, Continuous Integration Tool, Test Execution Tool, Defect Management Tool, Test Design Tool

#### NEW QUESTION 118

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

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

**Answer:** D

#### Explanation:

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

#### NEW QUESTION 123

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

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

**Answer:** A

#### Explanation:

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

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

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

#### NEW QUESTION 124

Can "cost" be regarded as Exit criteria?

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

**Answer:** A

#### Explanation:

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

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

#### NEW QUESTION 129

Which of the following statements about error guessing is true?

- A. Error guessing is a system that adopts artificial intelligence to predict whether software components are likely to contain defects or not
- B. Experienced testers, when applying error guessing, rely on the use of a high-level list of what needs to be tested as a guide to find defects
- C. Error guessing refers to the ability of a system or component to continue normal operation despite the presence of erroneous inputs
- D. Experienced testers, when applying error guessing technique, can anticipate where errors, defects and failures have occurred and target their tests at those issues

**Answer:** D

**Explanation:**

This answer is correct because error guessing is a test design technique where the experience and intuition of the tester are used to anticipate where errors, defects and failures have occurred or are likely to occur, and to design test cases to expose them. Error guessing can be based on factors such as the complexity of the system or component, the known or suspected weaknesses of the system or component, the previous history of defects, or the common types of errors in the domain or technology. Error guessing can be used as a complementary technique to other more systematic or formal techniques, or when there is insufficient information or time to apply them. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.5

**NEW QUESTION 134**

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 138**

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 140**

Which TWO of the following are benefits of continuous integration?

- A. Allows earlier detection and easier root cause analysis of integration problems and conflicting changes.I
- B. Removes the need for manual test analysis, design and execution.H
- C. Removes the dependency on automated regression packs when integrating larger systems, or components.i
- D. Gives the development team regular feedback on whether the code is workin
- E. Select the correct Answer:
- F. i and iv
- G. i and ii
- H. i and iii
- I. iii and iv

**Answer:** A

**Explanation:**

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

**NEW QUESTION 142**

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

- A. Configuration management helps reduce testing effort by identifying a manageable number of test environment configurations in which to test the software, out of all possible configurations of the environment in which the software will be released
- B. Configuration management is an administrative discipline that includes change control, which is the process of controlling the changes to identified items referred to as Configuration Items'
- C. Configuration management is an approach to interoperability testing where tests are executed in the cloud, as the cloud can provide cost-effective access to

multiple configurations of the test environments

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

**Answer:** D

**Explanation:**

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

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

**NEW QUESTION 147**

.....



## Relate Links

**100% Pass Your CTFL4 Exam with ExamBible Prep Materials**

<https://www.exambible.com/CTFL4-exam/>

## Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>