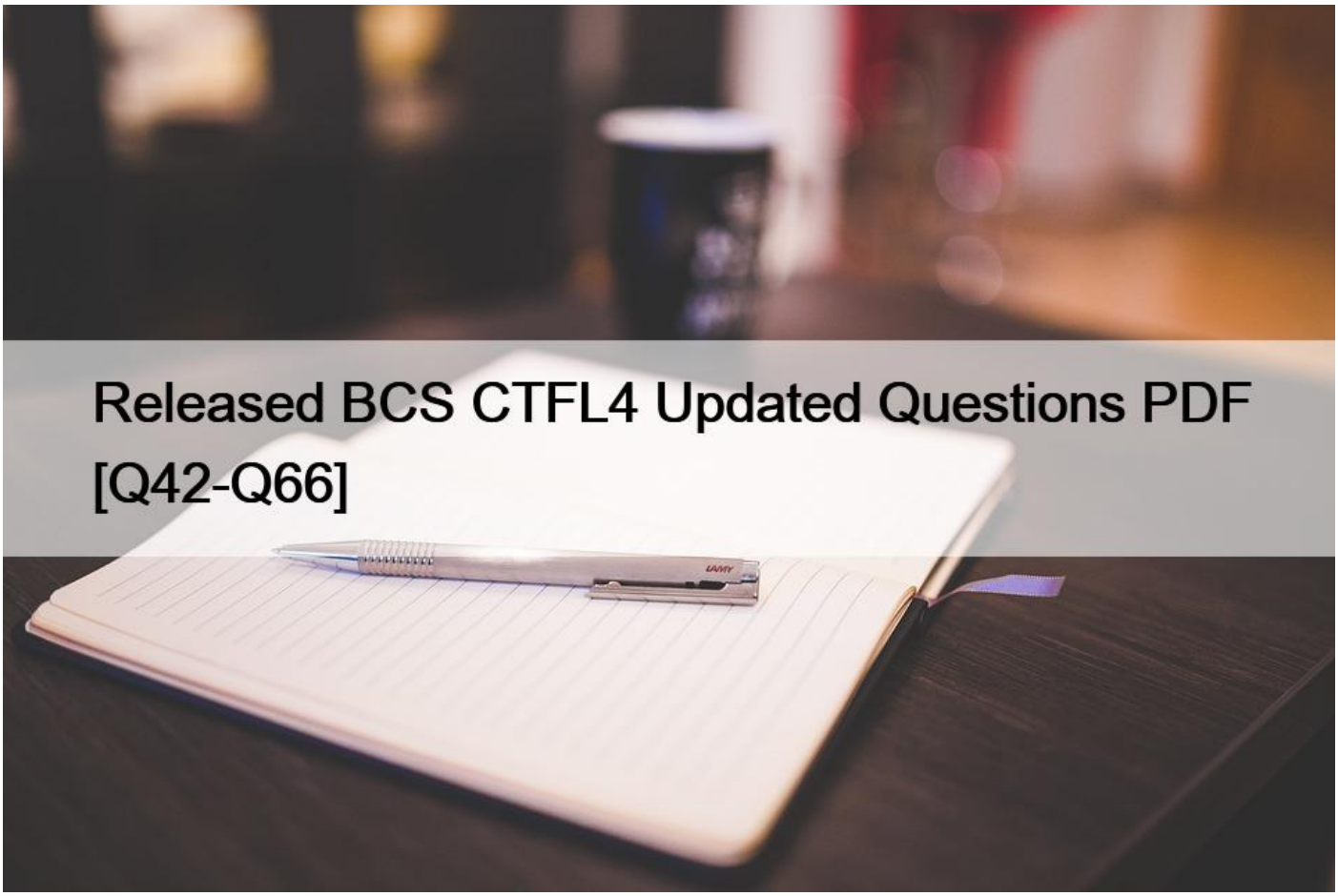
Released BCS CTFL4 Updated Questions PDF [Q42-Q66



Released BCS CTFL4 Updated Questions PDF
CTFL4 Dumps and Practice Test (150 Exam Questions)

Q42. Which of the following statements is true?
* A defect does not always produce a failure, while a bug always produces a failure
* A defect may cause a failure which, when occurring, always causes an error
* Failures can be caused by defects, but also by environmental conditions
* Bugs are defects found during component testing, while failures are defects found at higher test levels

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. Reference: 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

Q43. Which of the following statements about white-box test techniques is true?
* 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
* 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
* 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
* 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
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. Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.2

Q44. 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?
* Get familiar with the functionality and options of the tool
* Check haw the tool fits to the existing test processes
* Train all testers on using the tool
* Decide upon standards for tool implementation
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:

To get familiar with the functionality and options of the tool

To check how the tool fits to the existing test processes and environment To assess the benefits and challenges of using the tool To decide upon standards and guidelines for tool implementation and usage To estimate the costs and resources required for tool deployment and maintenance 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.

Reference:

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

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

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

Q45. Which of the following statements is true?
* 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
* Sequential development models impose the use of systematic test techniques and do not allow the use of experience-based test techniques
* 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
* 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

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. Reference: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.2, Section 3.2.1.2

Q46. Which one of the following statements correctly describes the term ‘debugging’?
* There is no difference between debugging and testing.
* Debugging is a confirmation activity that checks whether fixes resolved defects.
* Debugging is the development activity that finds, analyses and fixes defects.
* Debugging is of no relevance in Agile development.

Debugging is the development activity that finds, analyses and fixes defects. Unlike testing, which aims to identify defects in the software, debugging is the process that developers use to locate and correct the errors found during testing. This involves diagnosing the root causes of these defects and making necessary code changes to resolve them. Debugging is a critical part of the development cycle and ensures that the software functions correctly after defects are fixed.

Reference: ISTQB CTFL Syllabus V4.0, Section 1.1.2

Q47. The statement: “Test activities should start in the early stages of the lifecycle, adhering to the testing principle of early testing” is relevant to which of the recognized software development models?
* Sequential development model.
* Iterative development model.
* Incremental development model.
* All the above

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

Q48. During which main group of test activity are the following tasks performed?

* Checking test results and logs against specified coverage criteria.

* Assessing the level of component or system quality based on test results and logs.

* Determining whether more tests are needed.

Select the correct answer:
* Test planning.
* Test analysis.
* Test design.
* Test monitoring and control.

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

Q49. Which of the following statements about the value of maintaining traceability between the test basis and test work products is not true?
* Traceability can be useful for assessing the impact of a change to a test basis item on the corresponding tests
* Traceability can be useful for determining how many test basis items are covered by the corresponding tests
* Traceability can be useful for determining the most suitable test techniques to be used in a testing project
* Traceability can be useful to support the needs required by the auditing of testing

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

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

Traceability can be useful for determining how many test basis items are covered by the corresponding tests: This statement is true, because traceability can help to measure the test coverage, which is the degree to which the test basis is exercised by the test cases. Traceability can help to identify which test basis items are covered, partially covered, or not covered by the tests, and to evaluate the adequacy, the completeness, and the effectiveness of the testing process. Traceability can also help to identify the gaps, the overlaps, or the redundancies in the test coverage, and to prioritize, optimize, or improve the test cases.

Traceability can be useful to support the needs required by the auditing of testing: This statement is true, because traceability can help to provide evidence, documentation, and justification for the testing activities, results, and outcomes. Traceability can help to demonstrate that the testing process follows the standards, the regulations, the policies, and the best practices that are applicable to the software system, the project, or the organization. Traceability can also help to verify that the testing process meets the expectations, the needs, and the satisfaction of the users and the stakeholders. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches1 ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1 ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.2, Test Monitoring and Control1 ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design1 ISTQB Glossary of Testing Terms v4.0, Traceability, Test Basis, Test Artifact, Test Objective, Test Level, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time2

Q50. 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
* Test execution
* Test design.
* Test analysis.
* Test planning

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.

Q51. Match each objective to the correct test level

Objective:

A) Verifying whether the functional and non-functlonal behaviors of the system are as designed and specified.

B) Verifying whether the functional and non-functlonal 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.
* A3, B2. C4. D1
* A2, B3, Cl, D4
* A3. B2. C1. D4
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).

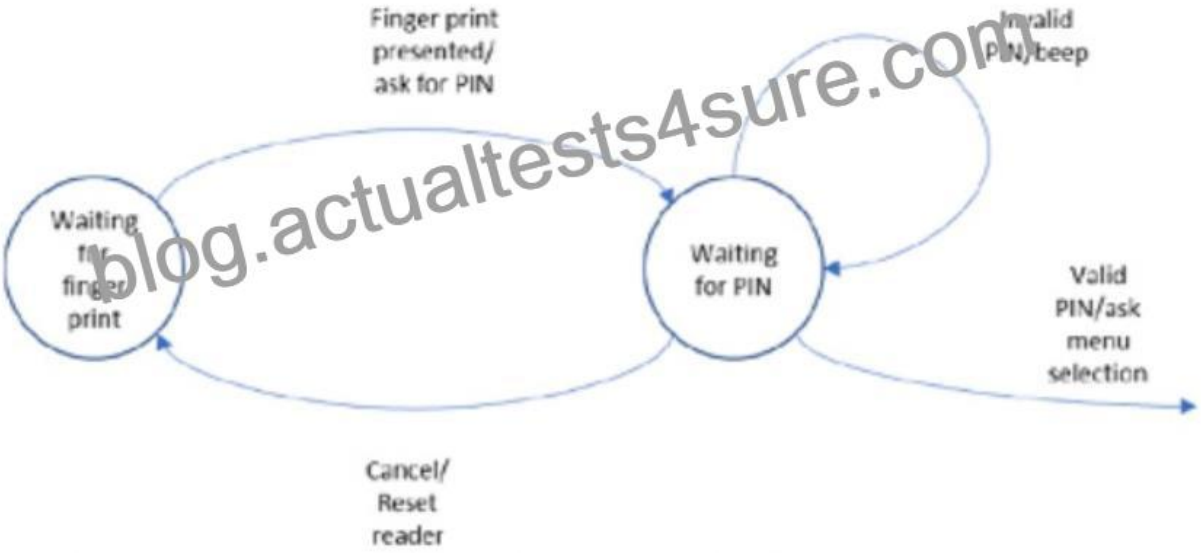
Verifying whether the functional and non-functional behaviors of the components are as designed and specified (C1: Component testing).

Establishing confidence in the quality of the system as a whole (D4: Acceptance testing).

Q52. Which of the following types of tools is BEST suited for determining source code compliance with the guidelines provided by a coding standard?
* Containerisation tool
* Fault seeding tool.
* Static analysis tool.
* Test data preparation tool
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.

Q53. Which statement is true regarding confirmation testing and regression testing?
* 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.
* Confirmation testing is an optional activity whilst regression testing is not negotiable.
* Confirmation testing aims to verify that a defect has been resolved and regression testing ensuring that existing functionality still works after a change.
* Testers’ involvement is essential whilst running retesting and regression testing.
* TESTER Involvement is essential whilst running retesting and regression testing.
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.

Q54. The following state transition diagram describes the functionality involved in a system using fingerprint and password authentication to log onto a system.



How many distinct states of the system are visible in the above diagram?
* 1
* 2
* 3
* 4
The state transition diagram provided shows three distinct states:

Waiting for fingerprint

Waiting for PIN

Valid PIN/ask menu selection

Each state represents a different stage in the system’s operation, with transitions based on user actions and system responses.

Q55. Which of the following statements about static testing and dynamic testing is true?
* Unlike dynamic testing, which can be also performed manually, static testing cannot be performed without specialized tools
* Static testing is usually much less cost-effective than dynamic testing
* Unlike dynamic testing, which focuses on detecting potential defects, static testing focuses on detecting failures which may be due to actual defects
* Both static testing and dynamic testing can be used to highlight issues associated with non-functional characteristics
This answer is correct because static testing and dynamic testing are both types of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc. Static testing is a type of testing that involves the analysis of software work products, such as requirements, design, code, or test cases, without executing them. Dynamic testing is a type of testing that involves the execution of software work products, such as code or test cases, using inputs and verifying outputs. Both static testing and dynamic testing can be applied to different test levels and test types, and can use different test techniques and tools, to evaluate the non-functional characteristics of the software product. Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.1.1, Section 2.2.1.2

Q56. Given the following User Story: “As an online customer, I would like to be able to cancel the purchase of an individual

item from a shopping list so that it only displays the relevant items, in less than 1 second”, which of the following can be considered as applicable acceptance test cases?

I . Click on my online shopping list, select the unwanted Item, delete the unwanted item, the unwanted Item is deleted from the shopping list in less than 1 second.

Ii . Click on my online shopping list, select all the items, delete all the items, the unwanted items are deleted from the shopping list in less than 1 second.

Iii . Tab to the online shopping list and press enter, select the unwanted item, delete the unwanted item, the unwanted item is deleted from the shopping list In less than 1 second.

Iv . Click on the checkout button, select the payment method, make payment, confirmation received of payment and shipping date.

V . Click on my shopping list, select the unwanted Item, delete the unwanted item, the unwanted item is deleted from the shopping list.

Select the correct answer:
* I, ii and v
* iv
* i and iii
* v

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

Reference: ISTQB CTFL Syllabus V4.0, Section 5.2.2

Q57. A Test Manager conducts risk assessment for a project. One of the identified risks is: The sub-contractor may fail to meet his commitment”. If this risk materializes. it will lead to delay in completion of testing required for the current cycle.

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

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

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

The other options are not correct because:

A . It is not true that any risk associated with development timeline is a product risk. Some risks could be project risks, which are risks that affect the management or control of the software project, such as budget, resources, schedule, or communication1. For example, a risk of losing a key project stakeholder is a project risk, not a product risk.

B . It is not true that the risk is no longer a risk for the Test Manager since an independent party is managing it. The Test Manager is still responsible for ensuring that the testing activities are completed according to the test plan and the quality objectives4. The Test Manager should monitor and control the sub-contractor’s performance and communicate with him regularly to identify and mitigate any potential issues or deviations5.

C . It is not clear what is meant by “object” in this option, but it could be interpreted as the software system under test or the test object6. In any case, the risk is not an object risk, as it does not affect the successful completion of the object, but rather the successful completion of the testing of the object. An object risk could be a risk that affects the functionality, reliability, usability, efficiency, maintainability, or portability of the software system under test2. For example, a risk of the software system having a high complexity or a low testability is an object risk, not a product risk.

Reference =

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

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

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

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

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

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

Q58. A typical objective of testing is to ensure that:
* testing is used to drive the development of a software
* a software has been tested using a combination of test techniques
* there are no defects in a software that is about to be released
* a software has been properly covered

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. Reference: ISTQB Foundation Level Syllabus v4.0, Section 2.3.1.1, Section 2.3.2

Q59. 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;
* Mistake
* Fault
* Error
* Failure

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.

Reference = 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.

Q60. Which TWO of the following are benefits of continuous integration?

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

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

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

iv . Gives the development team regular feedback on whether the code is working.

Select the correct answer:
* i and iv
* i and ii
* i and iii
* iii and iv

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.

Q61. A document descnbes 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:
* Test design.
* Test analysis
* Test Implementation.
* Test monitoring and control

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.

Q62. The acceptance criteria associated with a user story:
* are often written in a rule-oriented format using the template referred to as “Given/When/Then”
* are often documented following in rule-oriented format using the following template: “As a [role], I want [feature], so that I can [benefit]”
* can be written in different formats and represent an aspect of a user story referred to as confirmation’ of the so called “3 C’s”
* must be written in one of the two following formats: scenario-oriented or rule-oriented

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

Q63. Which of the following statements about estimation of the test effort is WRONG?
* Once the test effort is estimated, resources can be identified and a schedule can be drawn up.
* Effort estimate can be inaccurate because the quality of the product under tests is not known.
* Effort estimate depends on the budget of the project.
* Experience based estimation is one of the estimation techniques.

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

The other options are correct because:

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

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

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

Reference =

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

Q64. Which of the following coverage criteria results in the highest coverage for state transition based test cases?
* Can’t be determined
* Covering all transitions at least once
* Covering only start and end states
* Covering all states at least once

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. Reference = CTFL 4.0 Syllabus, Section 4.2.3, page 49-50.

Q65. Atypical generic skill required for the role of tester is the ability to
* Take on the role of developer to meet challenging project deadlines
* Assume leadership aimed at imposing decisions on the rest of the team.
* Use tools to make the execution of repetitive testing tasks more efficient.
* Determine the corrective actions to get a test project on track in case of deviations from the test plan

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.

Q66. Which of the following best describes the way in which statement coverage is measured?
* Measured as the number of decision outcomes executed by the tests, divided by the total number of decision outcomes in the test object.
* It is not possible to accurately measure statement coverage.
* Measured as the number of statements executed by the tests, divided by the total number of executable statements in the code.
* Measured as the number of lines of code executed by the test, divided by the total number of lines of code in the test object.

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

CTFL4 Exam Dumps Pass with Updated 2024 Certified Exam Questions:
https://www.actualtests4sure.com/CTFL4-test-questions.html]