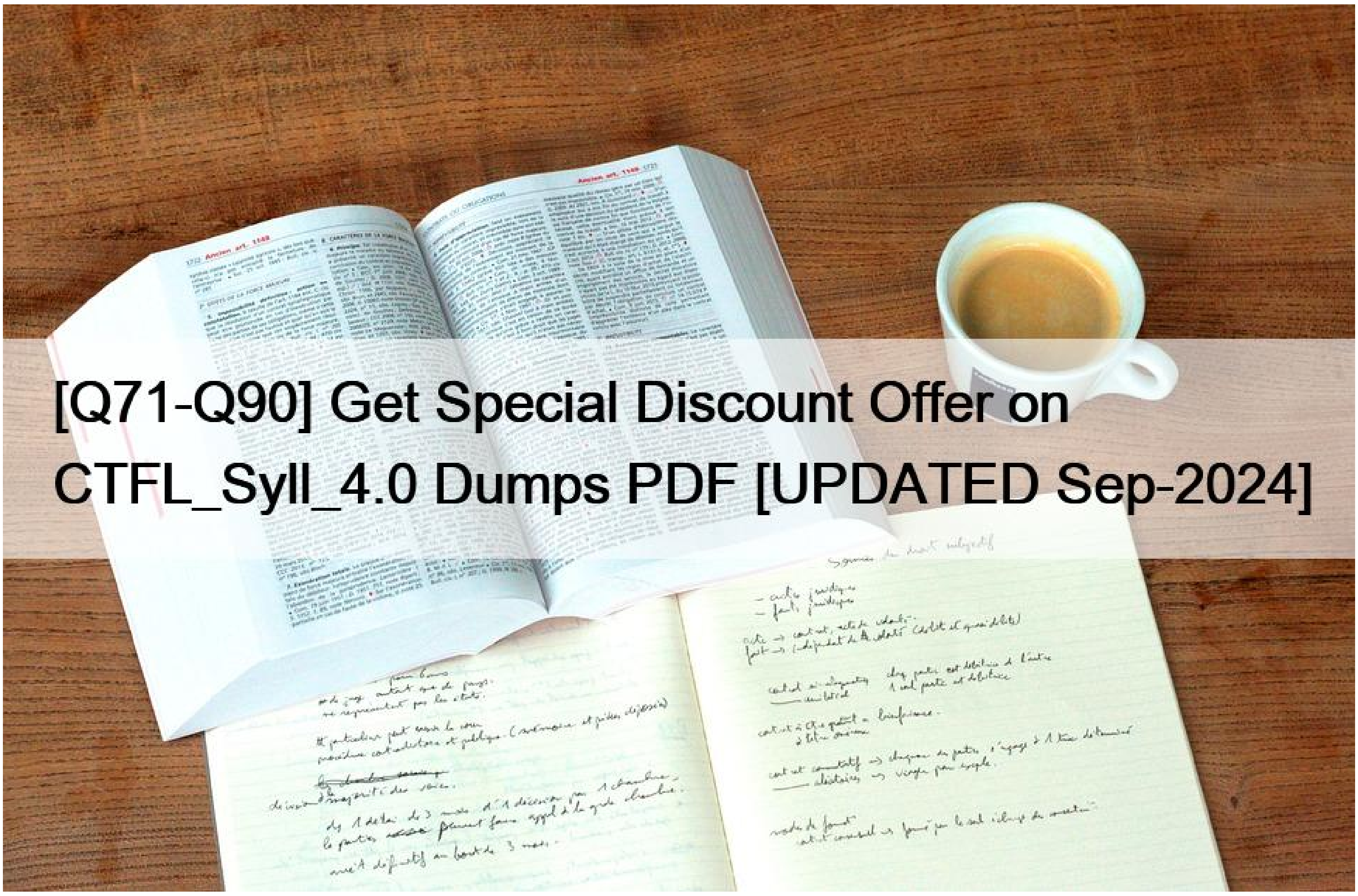
[Q71-Q90 Get Special Discount Offer on CTFL_Syll_4.0 Dumps PDF [UPDATED Sep-2024]



Get Special Discount Offer on CTFL_Syll_4.0 Dumps PDF [UPDATED Sep-2024]
PDF Download ISQI Test To Gain Brilliante Result!

Q71. Confirmation testing is performed after:
* a defect is fixed and after other tests do not find any side-effect introduced in the software as a result of such fix
* a failed test, and aims to run that test again to confirm that the same behavior still occurs and thus appears to be reproducible
* the execution of an automated regression test suite to confirm the absence of false positives in the test results
* a defect is fixed, and if such testing is successful then the regression tests that are relevant for such fix can be executed
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.

Q72. The acceptance criteria associated with a user story:
* are often written in a rule-oriented format using the template referred to as &#8220;Given/When/Then&#8221;
* are often documented following in rule-oriented format using the following template: &#8220;As a [role], I want [feature], so that I can [benefit]&#8221;
* can be written in different formats and represent an aspect of a user story referred to as confirmation&#8217; of the so called &#8220;3 C&#8217;s&#8221;
* must be written in one of the two following formats: scenario-oriented or rule-oriented
Explanation

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 &#8220;3 C&#8217;s&#8221; 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: &#8220;As a [role], I want [feature], so that I can

[benefit]&#8221;. 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.

References: ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 3.2.2, page 35-36; ISTQB Glossary v4.02, page 37.

Q73. In a two-hour uninterrupted test session, performed as part of an iteration on an Agile project, a heuristic checklist was used to help the tester focus on some specific usability issues of a web application.

The unscripted tests produced by the tester&#8217;s experience during such session belong to which one of the following testing quadrants?
* Q1
* Q2
* Q3
* Q4
Explanation

The unscripted tests produced by the tester&#8217;s experience during the two-hour test session belong to the testing quadrant Q3. The testing quadrants are a classification of testing types based on two dimensions: the test objectives (whether the testing is focused on supporting the team or critiquing the product) and the test basis (whether the testing is based on the technology or the business). The testing quadrants are labeled as Q1, Q2, Q3, and Q4, and each quadrant represents a different testing perspective, such as unit testing, acceptance testing, usability testing, or performance testing. The testing quadrant Q3 corresponds to the testing types that have the objective of critiquing the product from the business perspective, such as exploratory testing, usability testing, user acceptance testing, alpha testing, beta testing, etc. The unscripted tests performed by the tester in the given scenario are examples of exploratory testing and usability testing, as they are based on the tester&#8217;s experience, intuition, and learning of the web application, and they focus on some specific usabilityissues, such as the user interface, the user satisfaction, the user feedback, etc. The other options are incorrect, because:

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

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

The testing quadrant Q4 corresponds to the testing types that have the objective of critiquing the product from the technology perspective, such as performance testing, security testing, reliability testing, compatibility testing, etc. These testing types are usually performed by testers or specialists who have access to the tools, the metrics, the standards, or the benchmarks of the software system, and they aim to evaluate the non-functional aspects of the software system, such as the efficiency, the security, the reliability, or the compatibility of the software system under different conditions or environments.

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

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles ISTQB Glossary of Testing Terms v4.0, Testing Quadrant, Exploratory Testing, Usability Testing, Unit Testing, Component Testing, Integration Testing, System Testing, Functional Testing, Acceptance Testing, Story Testing, Scenario Testing, Performance Testing, Security Testing, Reliability Testing, Compatibility Testing

Q74. A Test Manager conducts risk assessment for a project. One of the identified risks is: The sub-contractor may fail to meet his commitment&#8221;. 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&#8217;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 &#8220;object&#8221; 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.

References =

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

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

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

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

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

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

Q75. After being in operation for many years, a document management system must be decommissioned as it has reached its end of life. This system will not be replaced by any other new system. A legal obligation provides that all documents within the system must be kept for at least 20 years in a state archive.

Which of the following statements about maintenance testing for decommissioning of this system is true?
* No maintenance testing is required as this system will not be replaced
* Data migration testing is required as part of maintenance testing
* Confirmation testing is required as part of maintenance testing
* Regression testing is required as part of maintenance testing

When a document management system is decommissioned, maintenance testing must include data migration testing to ensure that all documents are correctly transferred to a state archive, meeting legal requirements for long-term storage. This process verifies that data integrity is maintained during migration.

References:

* ISTQB CTFL Syllabus 4.0, Chapter 2.3, page 29: Maintenance Testing and Data Migration

Q76. Which of the following statements is not correct?
* Looking for defects in a system may require Ignoring system details
* Identifying defects may be perceived as criticism against product
* Looking for defects in system requires professional pessimism and curiosity
* Testing is often seen as a destructive activity instead of constructive activity

Looking for defects in a system does not require ignoring system details, but rather paying attention to them and understanding how they affect the system&#8217;s quality, functionality, and usability. Ignoring system details could lead to missing important defects or testing irrelevant aspects of the system.

Identifying defects may be perceived as criticism against product, especially by the developers or stakeholders who are invested in the product&#8217;s success. However, identifying defects is not meant to be a personal attack, but rather a constructive feedback that helps to improve the product and ensure its alignment with the requirements and expectations of the users and clients.

Looking for defects in system requires professional pessimism and curiosity, as testers need to anticipate and explore the possible ways that the system could fail, malfunction, or behave unexpectedly. Professional pessimism means being skeptical and critical of the system&#8217;s quality and reliability, while curiosity means being eager and interested in finding out the root causes and consequences of the defects.

Testing is often seen as a destructive activity instead of constructive activity, as it involves finding and reporting the flaws and weaknesses of the system, rather than creating or enhancing it. However, testing is actually a constructive activity, as it contributes to the system&#8217;s improvement, verification, validation, and optimization, and ultimately to the delivery of a high-quality product that meets the needs and expectations of the users and clients.

Q77. Which of the following is an example of scenario-oriented acceptance criteria?
* Verify that a registered user can create add a new project with name having more than 100 characters
* An unregistered user shouldn&#8217;t be shown any report.
* The user should be able to provide three inputs to test the product &#8211; the Al model to be tested, the data used and an optional text file
* A user is already logged in then on navigating to the Al model testing page the user should be directly shown the report of last test run.
Scenario-oriented acceptance criteria describe how a system should behave in a specific situation or scenario.

These criteria are typically written from the end-user&#8217;s perspective and focus on user interactions and system responses. Option D fits this description as it outlines a specific scenario where a user is already logged in and describes the expected behavior when the user navigates to a particular page, which is to show the report of the last test run. This type of criterion ensures that the system meets user expectations in that scenario.

Q78. Which of the following s the most correct statement about state testing techniques?
* Static techniques can be used before all code is ready for execution
* Static techniques find more detects then dynamic techniques.
* Static techniques can be used by inexperienced users.
* Static techniques are always cheaper than dynamic techniques.
State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques require the system to be executed with test cases, whereas static testing techniques do not. Static testing techniques can be applied before the code is ready for execution, such as reviews, inspections, walkthroughs, and static analysis. Static testing techniques can help find defects early in the development process, improve the quality of the code, and reduce the cost and effort of dynamic testing. Reference = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 4, Section 4.2.1, Page 281; ISTQB Glossary of Testing Terms v4.0, Page 292

Q79. Which of the following statements about how different types of test tools support testers is true?
* The support offered by a test data preparation tool is often leveraged by testers to run automated regression test suites
* The support offered by a performance testing tool is often leveraged by testers to run load tests
* The support offered by a bug prediction tool is often used by testers to track the bugs they found
* The support offered by a continuous integration tool is often leveraged by testers to automatically generate test cases from a model
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

Q80. Which of the following lists factors That contribute to PROJECT risks?
* skill and staff shortages; problems in defining the right requirements, contractual issues.
* skill and staff shortages; software does not perform its intended functions; problems in defining the right requirements.
* problems in defining the right requirements; contractual issues; poor software quality characteristics.
* poor software quality characteristics; software does not perform its intended functions.

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.

Reference = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 1: Fundamentals of Testing, Section 1.5:

Risks and Testing, Pages 14-16.

Q81. In which of the following test documents would you expect to find test exit criteria described9
* Test design specification
* Project plan
* Requirements specification
* Test plan
Test exit criteria are the conditions that must be fulfilled before concluding a particular testing phase. These criteria act as a checkpoint to assess whether we have achieved the testing objectives and are done with testing1. Test exit criteria are typically defined in the test plan document, which is one of the outputs of the test planning phase. The test plan document describes the scope, approach, resources, and schedule of the testing activities. It also identifies the test items, the features to be tested, the testing tasks, the risks, and the test deliverables2. According to the ISTQB® Certified Tester Foundation Level Syllabus v4.0, the test plan document should include the following information related to the test exit criteria3:

The criteria for evaluating test completion, such as the percentage of test cases executed, the percentage of test coverage achieved, the number and severity of defects found and fixed, the quality and reliability of the software product, and the stakeholder satisfaction.

The criteria for evaluating test process improvement, such as the adherence to the test strategy, the efficiency and effectiveness of the testing activities, the lessons learned and best practices identified, and the recommendations for future improvements.

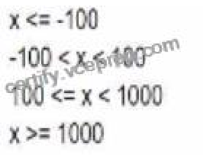
Therefore, the test plan document is the most appropriate test document to find the test exit criteria described. The other options, such as test design specification, project plan, and requirements specification, are not directly related to the test exit criteria. The test design specification describes the test cases and test procedures for a specific test level or test type3. The project plan describes the overall objectives, scope, assumptions, risks, and deliverables of the software project4. The requirements specification describes the functional and non-functional requirements of the software product5. None of these documents specify the conditions for ending the testing process or evaluating the testing outcomes. Reference = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Entry and Exit Criteria in Software Testing | Baeldung on Computer Science, Entry And Exit Criteria In Software Testing &#8211; Rishabh Software, Entry and Exit Criteria in Software Testing Life Cycle &#8211; STLC [2022 Updated] &#8211; Testsigma Blog, ISTQB® releases Certified Tester Foundation Level v4.0 (CTFL).

Q82. Which of the following statements about TDD, BDD and ATDD is true?
* Refactoring is a practice that is an integral part of TDD and is applied both to tests and to code written to satisfy those tests
* ATDD is a black-box test design technique that is applicable exclusively at acceptance test level
* BDD is a developer practice where business stakeholders are not usually involved as the tests are directly written at unit/component test level
* ATDD is the practice of running the automated acceptance tests as part of a continuous integration process
Test-Driven Development (TDD) includes refactoring as a key practice. After writing tests and the code to satisfy those tests, refactoring is performed to improve the code and test quality without changing the functionality. This continuous process helps maintain clean, efficient, and manageable code.

References:

* ISTQB CTFL Syllabus 4.0, Chapter 2.1.3, page 25: TDD, ATDD, and BDD Practices

Q83. The following 4 equivalence classes are given:



x <= -100
-100 < x < 100
100 <= x < 1000
x >= 1000

Which of the following alternatives includes correct test values for x. based on equivalence partitioning?
* -100; 100:1000; 1001
* -500; 0; 100; 1000
* -99; 99:101; 1001
* -1000; -100; 100; 1000

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

(x leq -100)

(-100 < x < 100)

(100 leq x < 1000)

(x geq 1000)

Option D provides a value from each of these partitions:

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

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

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

For (x geq 1000), it gives 1500.

So, option D covers all four given equivalence classes with appropriate values.

Reference:

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

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

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

Q84. In a two-hour uninterrupted test session, performed as part of an iteration on an Agile project, a heuristic checklist was used to help the tester focus on some specific usability issues of a web application.

The unscripted tests produced by the tester&#8217;s experience during such session belong to which one of the following testing quadrants?
* Q1
* Q2
* Q3
* Q4

The unscripted tests produced by the tester&#8217;s experience during the two-hour test session belong to the testing quadrant Q3. The testing quadrants are a classification of testing types based on two dimensions: the test objectives (whether the testing is focused on supporting the team or critiquing the product) and the test basis (whether the testing is based on the technology or the business). The testing quadrants are labeled as Q1, Q2, Q3, and Q4, and each quadrant represents a different testing perspective, such as unit testing, acceptance testing, usability testing, or performance testing. The testing quadrant Q3 corresponds to the testing types that have the objective of critiquing the product from the business perspective, such as exploratory testing, usability testing, user acceptance testing, alpha testing, beta testing, etc. The unscripted tests performed by the tester in the given scenario are examples of exploratory testing and usability testing, as they are based on the tester&#8217;s experience, intuition, and learning of the web application, and they focus on some specific usability issues, such as the user interface, the user satisfaction, the user feedback, etc. The other options are incorrect, because:

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

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

* The testing quadrant Q4 corresponds to the testing types that have the objective of critiquing the product from the technology perspective, such as performance testing, security testing, reliability testing, compatibility testing, etc. These testing types are usually performed by testers or specialists who have access to the tools, the metrics, the standards, or the benchmarks of the software system, and they aim to evaluate the non-functional aspects of the software system, such as the efficiency, the security, the reliability, or the compatibility of the software system under different conditions or environments.

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

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

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

Q85. Which of the following statements best describes how configuration management supports testing?
* 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
* 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&#8217;
* 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
* 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 answer is correct because configuration management is a process of establishing and maintaining consistency of a product&#8217;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. Reference: : ISTQB Glossary of Testing Terms v4.0, : ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.2

Q86. 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.

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

Q87. 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
Explanation

This answer is correct because in Agile software development, work product documentation, such as user stories, acceptance criteria, or test cases, tends to be lightweight and concise, as the focus is on working software and frequent communication rather than comprehensive documentation. Manual tests tend to be often unscripted, as they are often produced using experience-based test techniques, such as error guessing or exploratory testing, which rely onthe tester&#8217;s skills, knowledge, and creativity to find defects and provide feedback. References: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.2, Section 3.2.1.2

Q88. Which of the following statements describes regression testing?

I) Retesting of a fixed defect

II) Testing of an already tested program

III) Testing of new functionality in a program

IV) Regression testing applies only to functional testing

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

Q89. Which of the following statements about branch coverage is true?
* The minimum number of test cases needed to achieve full branch coverage, is usually lower than that needed to achieve full statement coverage
* If full branch coverage has been achieved, then all unconditional branches within the code have surely been exercised
* If full branch coverage has been achieved, then all combinations of conditions in a decision table have surely been exercised
* Exercising at least one of the decision outcomes for all decisions within the code, ensures achieving full branch coverage
Exercising at least one of the decision outcomes for all decisions within the code, ensures achieving full branch coverage, which is a test coverage criterion that requires that all branches in the control flow of the code are executed at least once by the test cases. A branch is a basic block of code that has a single entry point and a single exit point, and a decision is a point in the code where the control flow can take more than one direction, such as an if-then-else statement, a switch-case statement, a loop statement, etc. The decision outcomes are the possible paths that can be taken from a decision, such as the then branch or the else branch, the case branch or the default branch, the loop body or the loop exit, etc. The other statements are false, because:

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

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

If full branch coverage has been achieved, then all combinations of conditions in a decision table have not necessarily been exercised, as a decision table is a test design technique that represents the logical relationships between multiple conditions and their corresponding actions, in a tabular format. A decision table can have more combinations of conditions than the number of decision outcomes in the code, as each condition can have two or more possible values, such as true or false, yes or no, etc. For example, a decision table with four conditions can have 16 combinations of conditions, but the corresponding code may have only two decision outcomes, such as pass or fail. To exercise all combinations of conditions in a decision table, a stronger test coverage criterion is needed, such as condition combination coverage, which requires that all possible combinations of condition outcomes in the code are executed at least once by the test cases. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.3.1, Test Coverage Criteria Based on the Structure of the Software ISTQB® Glossary of Testing Terms v4.0, Branch Coverage, Statement Coverage, Branch, Decision, Decision Outcome, Unconditional Branch, Decision Table, Condition Combination Coverage

Q90. Which ONE of the following is a characteristic of exploratory testing?
* Effectiveness depends on the individual testers&#8217; skills
* Usually conducted when there is sufficient time for testing
* Test cases are written once the specifications become available
* Testing without defined time-boxes

Exploratory testing is characterized by its reliance on the skills and experience of the tester. The effectiveness of exploratory testing depends heavily on the tester&#8217;s ability to design and execute tests based on their intuition and knowledge of the application. This type of testing is often performed without predefined test cases, making the individual tester&#8217;s expertise crucial.

References:

* ISTQB CTFL Syllabus V4.0, Section 4.4 on experience-based testing techniques, including exploratory testing, which highlights the importance of the tester&#8217;s skills.

CTFL_Syll_4.0 Dumps are Available for Instant Access: https://www.vceprep.com/CTFL_Syll_4.0-latest-vce-prep.html]