Free Questions for CTFL_Syll_4.0 by actualtestdumps

Shared by Hurst on 22-07-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

The acceptance criteria associated with a user story:

Options:

A- are often written in a rule-oriented format using the template referred to as 'Given/When/Then'

B- are often documented following in rule-oriented format using the following template: 'As a [role], I want [feature], so that I can [benefit]'

C- 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'

D- must be written in one of the two following formats: scenario-oriented or rule-oriented

Answer:

C

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 "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.

Question 2

Question Type: **MultipleChoice**

Following a risk-based testing approach you have designed 10 tests to cover a product risk with a high-risk level. You want to estimate, adopting the three-point test estimation technique, the test effort required to reduce the risk level to zero by executing those 10 tests. You made the following three initial estimates:

* most optimistic = 6 person hours

* most likely = 30 person hours

* most pessimistic = 54 person hours

Based only on the given information, which of the following answers about the three-point test estimation technique applied to this problem is true?

Options:

A- The final estimate is between 22 person hours and 38 person hours

B- The final estimate is exactly 30 person hours because the technique uses the initial most likely estimate as the final estimate

C- The final estimate is between 6 person hours and 54 person hours

D- The final estimate is exactly 30 person hours because the technique uses the arithmetic mean of the three initial estimates as the final estimate

Answer:

A

Explanation:

The three-point test estimation technique is a method of estimating the test effort based on three initial estimates: the most optimistic, the most likely, and the most pessimistic. The technique uses a weighted average of these three estimates to calculate the final estimate, which is also known as the expected value. The formula for the expected value is:

Expected value = (most optimistic + 4 * most likely + most pessimistic) / 6

Using the given values, the expected value is:

Expected value = (6 + 4 * 30 + 54) / 6 Expected value = 30 person hours

However, the expected value is not the only factor to consider when estimating the test effort. The technique also calculates the standard deviation, which is a measure of the variability or uncertainty of the estimates. The formula for the standard deviation is:

Standard deviation = (most pessimistic - most optimistic) / 6

Using the given values, the standard deviation is:

Standard deviation = (54 - 6) / 6 Standard deviation = 8 person hours

The standard deviation can be used to determine a range of possible values for the test effort, based on a certain level of confidence. For example, using a 68% confidence level, the range is:

Expected value standard deviation

Using the calculated values, the range is:

30 8 person hours

Therefore, the final estimate is between 22 person hours and 38 person hours, which is option A.

Question 3

Question Type: **MultipleChoice**

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

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

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

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

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

Which of the following statements is true?

Options:

A- [I] and [III] are product risks; [II] and [IV] are project risks

B- [I] and [IV] are product risks. [II] and [III] are project risks

C- [II], [III] and [IV] are product risks; [I] is a project risk

D- [IV] is a product risk; [I]. [II] and [III] are project risks

Answer:

B

Explanation:

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

Question 4

Question Type: MultipleChoice

A typical objective of testing is to ensure that:

Options:

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

Question 5

Question Type: MultipleChoice

The following rules determine the annual bonus to be paid to a salesman of a company based on the total annual amount of the sales made (referred to as TAS).

If the TAS is between 50k and 80k, the bonus is 10%. If the TAS exceeds 80k by a value not greater than 40k, the bonus is 15%. Finally, if the TAS

exceeds the maximum threshold which entitles to a 15% bonus, the bonus is 22%.

Consider applying equivalence partitioning to the TAS (Note: 1k = 1000 euros).

Which one of the following answers contain only test cases that belong to the same equivalence partition?

Options:

A- TC1 = 81 k; TC2= 97k; TC3=111k; TC4=118k

B- TC1 = 40k; TC2= 46k; TC3=51k; TC4=53k

C- TC1 = 79k; TC2= 80k; TC3=81k; TC4=82k

D- TC1 = 90k; TC2= 110k; TC3=125k: TC4=140k

Answer:

A

Explanation:

This answer is correct because equivalence partitioning is a test design technique that divides the input domain of a system or component into partitions of equivalent data, such that each partition is expected to produce the same output or behavior. Equivalence partitioning aims to reduce the number of test cases by selecting one representative value from each partition. In this case, the input domain of the TAS can be divided into four partitions based on the bonus rules: less than 50k, between 50k and 80k, between 80k and 120k, and more than 120k. The test cases in the answer belong to the same partition, which is between 80k and 120k, and they are expected to produce the same output, which is a bonus of 15%.Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.1

Question 6

Question Type: MultipleChoice

A virtual service emulating a real third-party service and the automated test scripts (aimed at testing the system under test) that interact with that service, are test work products that are typically created during:

Options:

A- Test monitoring and control

B- Test implementation

C- Test design

D- Test analysis

Answer:

B

Explanation:

This answer is correct because test implementation is the activity where test work products, such as test cases, test data, test scripts, test harnesses, test stubs, or virtual services, are created and verified. Test implementation also involves setting up the test environment and preparing the test execution schedule. A virtual service emulating a real third-party service and the automated test scripts that interact with that service are examples of test work products that are typically created during test implementation.Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.3

Question 7

Question Type: MultipleChoice

Which of the following statements refers to good testing practice to be applied regardless of the chosen software development model?

Options:

A- Tests should be written in executable format before the code is written and should act as executable specifications that drive coding

B- Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level

C- Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly

D- Involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle

Answer:

D

Explanation:

The statement that refers to good testing practice to be applied regardless of the chosen software development model is option D, which says that involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle. Work product reviews are static testing techniques, in which the work products of the software development process, such as the requirements, the design, the code, the test cases, etc., are examined by one or more reviewers, with or without the author, to identify defects, violations, or improvements. Involvement of testers in work product reviews can provide various benefits for the testing process, such as improving the test quality, the test efficiency, and the test communication. The early testing principle states that testing activities should start as early as possible in the software development lifecycle, and should be performed iteratively and continuously throughout the lifecycle. Applying the early testing principle can help to prevent, detect, and remove defects at an early stage, when they are easier, cheaper, and faster to fix, as well as to reduce the risk, the cost, and the time of the testing process. The other options are not good testing practices to be applied regardless of the chosen software development model, but rather specific testing practices that

may or may not be applicable or beneficial for testing, depending on the context and the objectives of the testing activities, such as:

Tests should be written in executable format before the code is written and should act as executable specifications that drive coding: This is a specific testing practice that is associated with test-driven development, which is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design, and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a good testing practice to be applied regardless of the chosen software development model, as it may not be feasible, suitable, or effective for testing in some contexts or situations, such as when the requirements are unclear, unstable, or complex, when the test automation tools or skills are not available or adequate, when the testing objectives or levels are not aligned with the unit testing, etc.

Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level: This is a specific testing practice that is associated with sequential software development models, such as the waterfall model, the V-model, or the W-model, in which the software development and testing activities are performed in a linear and sequential order, with well-defined phases, deliverables, and dependencies. Test levels are the stages of testing that correspond to the levels of integration of the software system, such as component testing, integration testing, system testing, and acceptance testing. Test levels should have clear and measurable entry criteria and exit criteria, which are the conditions that must be met before starting or finishing a test level. In sequential software development models, the exit criteria of one test level are usually part of the entry criteria for the next test level, to ensure that the software system is ready and stable for the next level of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be relevant, flexible, or efficient for testing in some contexts or situations, such as when the software development and testing activities are performed in an iterative and incremental order, with frequent changes, feedback, and adaptations, as in agile software development models, such as Scrum, Kanban, or XP, when the test levels are not clearly defined or distinguished, or when the test levels are performed in parallel or concurrently, etc.

Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly: This is a specific testing practice that is associated with uniform software development models, such as the spiral model, the incremental

model, or the prototyping model, in which the software development and testing activities are performed in a cyclical and repetitive manner, with similar phases, deliverables, and processes. Test objectives are the goals or the purposes of testing, which can vary depending on the test level, the test type, the test technique, the test environment, the test stakeholder, etc. Test objectives can be defined in terms of the test basis, the test coverage, the test quality, the test risk, the test cost, the test time, etc. Test objectives should be specific, measurable, achievable, relevant, and time-bound, and they should be aligned with the project objectives and the quality characteristics. In uniform software development models, the test objectives may be the same for all test levels, as the testing process is repeated for each cycle or iteration, with similar focus, scope, and perspective of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be appropriate, realistic, or effective for testing in some contexts or situations, such as when the software development and testing activities are performed in a hierarchical and modular manner, with different phases, deliverables, and dependencies, as in sequential software development models, such as the waterfall model, the V-model, or the W-model, when the test objectives vary according to the test levels, such as component testing, integration testing, system testing, and acceptance testing, or when the test objectives change according to the feedback, the learning, or the adaptation of the testing process, as in agile software development models, such as Scrum, Kanban, or XP, etc. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle1

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

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches1

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

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

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

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

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

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

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

ISTQB Glossary of Testing Terms v4.0, Work Product Review, Static Testing, Early Testing, Test-driven Development, Test Level, Entry Criterion, Exit Criterion, Test Objective, Test Basis, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time2

Question 8

Question Type: MultipleChoice

Which of the following statements about the value of maintaining traceability between the test basis and test work products is not true?

Options:

A- Traceability can be useful for assessing the impact of a change to a test basis item on the corresponding tests

B- Traceability can be useful for determining how many test basis items are covered by the corresponding tests

C- Traceability can be useful for determining the most suitable test techniques to be used in a testing project

D- Traceability can be useful to support the needs required by the auditing of testing

Answer:

C

Explanation:

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

Question 9

Question Type: MultipleChoice

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

Options:

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.Reference: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.2

Question 10

Question Type: MultipleChoice

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

Options:

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.Reference: 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