BCS

Exam Questions CTFL4

ISTQB Certified Tester Foundation Level CTFL 4.0 Exam

NEW QUESTION 1
Consider the following user story about the authentication functionality of an e-commerce website:
"As a logged-in user, I want to change my current password with a new one, so that I can make my account safer".
The following are some of the acceptance criteria defined for the user story:
[a] After the logged-in user has successfully changed his password, an email confirming the change must be sent to him
[b] To successfully change the password, the logged-in user must enter the current password, enter a new valid password, and finally confirm by pressing the 'Change Password' button
[c] To be valid, the new password entered by the logged-in user is not only required to meet the criteria related to the length and type of characters, but must also be different form the last 5 passwords of that user
[d] A dedicated error message must be presented to the logged-in user when he enters a wrong current password
[e] A dedicated error message must be presented to the logged-in user when he enters the correct current password, but enters an invalid password
Based only on the given information, which of the following ATDD tests is most likely to be written first?

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

Answer: B

Explanation:
ATDD stands for Acceptance Test-Driven Development, which is a collaborative approach to software development and testing, in which the acceptance criteria of a user story are defined and automated as executable tests before the implementation of the software system. ATDD tests are usually written in a Given-When-Then format, which describes the preconditions, the actions, and the expected outcomes of a test scenario. ATDD tests are intended to verify that the software system meets the expectations and the needs of the users and the stakeholders, as well as to provide feedback and guidance for the developers and the testers.
Based on the given information, the ATDD test that is most likely to be written first is the one that corresponds to option B, which is:
Given the logged-in user is on the Change Password page When the user enters the correct current password, enters a valid new password (different from the last 5 passwords), and presses the Change Password button Then the user receives an email confirming that the password has been successfully changed
This ATDD test is most likely to be written first, because it covers the main functionality and the happy path of the user story, as well as the most important acceptance criterion [a]. It also verifies that the user can change the password with a valid new password that meets the criteria related to the length, the type of characters, and the history of the passwords, as specified in the acceptance criterion [c]. The other options are not likely to be written first, because they either cover less critical or less frequent scenarios, such as entering a wrong current password [d] or an invalid new password [e], or they are not related to the user story or the acceptance criteria at all, such as submitting a purchase order [d]. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in
Software Development Lifecycles1
? ISTQB® Glossary of Testing Terms v4.0, Acceptance Test-Driven Development, User Story, Acceptance Criterion, Given-When-Then2

NEW QUESTION 2
A calculator software is used to calculate the result for 5+6. The user noticed that the result given is 6.
This is an example of;

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

Answer: D

Explanation:
According to the ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 18, a failure is ??an event in which a component or system does not perform a required function within specified limits??. In this case, the calculator software does not perform the required function of calculating the correct result for 5+6 within the specified limits of accuracy and precision. Therefore, this is an example of a failure.
The other options are incorrect because:
? A mistake is ??a human action that produces an incorrect result?? (page 25). A mistake is not an event, but an action, and it may or may not lead to a failure. For example, a mistake could be a typo in the code, a wrong assumption in the design, or a misunderstanding of the requirement.
? A fault is ??a defect in a component or system that can cause the component or system to fail to perform its required function?? (page 16). A fault is not an event, but a defect, and it may or may not cause a failure. For example, a fault could be a logical error in the code, a missing specification in the design, or a contradiction in the requirement.
? An error is ??the difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition?? (page 15). An error is not an event, but a difference, and it may or may not result in a failure. For example, an error could be a rounding error in the calculation, a measurement error in the observation, or a deviation error in the condition.
References = ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 15-18, 25;
ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 96, page 34.

NEW QUESTION 3
The acceptance criteria associated with a user story:

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

NEW QUESTION 4
Which of the following characterizations applies to a test tool used for the analysis of a developer's code prior to its execution?

A. Tool support for test design and implementation.
B. Tool support for static testing.
C. Tool support for test execution and logging.
D. Tool support for performance measurement and dynamic analysis.

Answer: B

Explanation:
A test tool used for the analysis of a developer's code prior to its execution falls under the category of static testing tools. Static testing involves examining the code and documentation without executing the code. These tools are used to perform static analysis, which helps in identifying potential defects and code quality issues early in the development process. The ISTQB CTFL syllabus specifies that static analysis tools are essential for finding defects that do not manifest themselves during the execution of the program.
References:ISTQB CTFL Syllabus, Section 3.1, "Static Testing."

NEW QUESTION 5
Which of the following best describes the way in which statement coverage is measured?

A. Measured as the number of decision outcomes executed by the tests, divided by the total number of decision outcomes in the test object.
B. It is not possible to accurately measure statement coverage.
C. Measured as the number of statements executed by the tests, divided by the total number of executable statements in the code.
D. 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.

Answer: C

Explanation:
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.

NEW QUESTION 6
Which of the following is not an example of a typical generic skill required for testing?

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

Answer: A

Explanation:
Test-driven development is not an example of a typical generic skill required for testing, but rather an example of a specific technical skill or a development practice that may or may not be relevant for testing, depending on the context and the objectives of the testing activities. Test-driven development is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design, and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a skill that is generally expected or needed for testers, especially for testers who are not involved in unit testing or who do not have access to the source code. The other options are examples of typical generic skills required for testing, which are skills that are applicable and beneficial for testing in any context or situation, regardless of the specific testing techniques, tools, or methods used. The typical generic skills required for testing include:
? Be able to use test management tools and defect tracking tools: These are tools that help testers to plan, organize, monitor, and control the testing activities andresources, as well as to record, track, analyze, and resolve the defects detected during testing. These tools can improve the efficiency, the effectiveness, and the communication of the testing process, as well as to provide traceability, metrics, and reports for the testing outcomes.
? Be able to communicate defects and failures to developers as objectively as possible: This is a skill that involves the ability to report and describe the defects and failures found during testing in a clear, concise, accurate, and unbiased manner, using relevant information, evidence, and terminology, without making assumptions, judgments, or accusations. This skill can facilitate the collaboration, the understanding, and the resolution of the defects and failures between the testers and the developers, as well as to prevent conflicts, misunderstandings, or blame games.
? Possess the necessary social skills that support effective teamwork: These are skills that involve the ability to interact, cooperate, and coordinate with other people involved in or affected by the testing activities, such as the test manager, the test team, the project manager, the developers, the customers, the users, etc. These skills can include communication, negotiation, leadership, motivation, feedback, conflict resolution, etc. These skills can enhance the quality, the productivity, and the satisfaction of the testing process, as well as to foster a positive and constructive testing culture. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches
? ISTQB® Glossary of Testing Terms v4.0, Test-driven Development, Test Management Tool, Defect Tracking Tool, Defect Report, Failure, Social Skill2

NEW QUESTION 7
Consider the following examples of risks identified in different software development projects
[1]. It may not be possible to generate the expected workloads to run performance tests, due to the poor hardware equipment of the machines (load injectors) that

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

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

Answer: A

Explanation:
In software testing, risks are categorized into product risks and project risks. Product risks are associated with the potential of a product to fail in meeting its quality criteria. Project risks are related to potential issues that could affect the project's ability to deliver a product.
? [i] is a project risk because it concerns the availability and adequacy of hardware resources for performance testing.
? [ii] is a product risk because it pertains to a security and functionality issue within the web application.
? [iii] is a project risk because it involves the availability of necessary requirements documentation for the testing process.
? [iv] is a product risk because it relates to the system's functionality and data integrity after a failure.
Thus, statement A correctly classifies [ii] and [iv] as product risks and [i] and [iii] as project risks.

NEW QUESTION 8
Which of the following statements is not correct?

A. Looking for defects in a system may require Ignoring system details
B. Identifying defects may be perceived as criticism against product
C. Looking for defects in system requires professional pessimism and curiosity
D. Testing is often seen as a destructive activity instead of constructive activity

Answer: A

Explanation:
? 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??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??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??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??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.

NEW QUESTION 9
Which of the following statements about statement coverage is TRUE?

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

Answer: D

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

NEW QUESTION 10
Which of the following is a typical potential risk of using test automation tools?

A. Reduced feedback times regarding software quality compared to manual testing.
B. Reduced test execution limes compared to manual testing.
C. Reduced repeatability and consistency of tests compared to manual testing
D. Underestimation of effort required to maintain test scripts.

Answer: D

Explanation:
One of the common risks associated with test automation tools is the underestimation of the effort required to maintain test scripts. Test scripts can become outdated or broken due to changes in the application, requiring significant effort to update and maintain them. This risk is highlighted in the ISTQB CTFL syllabus under the discussion of the benefits and risks of test automation.
References:ISTQB CTFL Syllabus, Section on test tools and automation.

NEW QUESTION 10
Determining the schedule for each testing activity and test milestones for a test project, using activity estimates, available resources, and other constraints is a typical task performed during

A. Test execution
B. Test design.
C. Test analysis.
D. Test planning

Answer: D

Explanation:
Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.

NEW QUESTION 12
A financial institution is to implement a system that calculates the interest rates paid on investment accounts based on the sum invested.
You are responsible for testing the system and decide to use equivalence partitioning and boundary value analysis to design test cases. The requirements describe the following expectations:
Investment range| Interest rate R500 to RIO 00010%
RIO 001 to R50 00011% R50 001 to RIOOOOO12% RIOOOOl to R500 000| 13%
What is the minimum number of test cases required to cover all valid equivalence partitions
for calculating the interest?

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

Answer: B

Explanation:
Using equivalence partitioning, the investment ranges are divided into four partitions:
? R500 to R10,000 (10%)
? R10,001 to R50,000 (11%)
? R50,001 to R100,000 (12%)
? R100,001 to R500,000 (13%)
Thus, the minimum number of test cases required to cover all valid equivalence partitions for calculating the interest is 4.

NEW QUESTION 16
The four test levels used in ISTQB syllabus are:
* 1. Component (unit) testing
* 2. Integration testing
* 3. System testing
* 4. Acceptance testing
An organization wants to do away with integration testing but otherwise follow V-model. Which of the following statements is correct?

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

Answer: D

Explanation:
The V-model is a software development life cycle model that defines four
test levels that correspond to four development phases: component (unit) testing with component design, integration testing with architectural design, system testing with system requirements, and acceptance testing with user requirements. The V-model emphasizes the importance of verifying and validating each phase of development with a corresponding level of testing, and ensuring that the test objectives, test basis, and test artifacts are aligned and consistent across the test levels. Therefore, an organization that wants to follow the V-model cannot do away with integration testing, as it would break the symmetry and completeness of the V-model, and compromise the quality and reliability of the software or system under test. Integration testing is a test level that aims to test the interactions and interfaces between components or subsystems, and to detect any defects or inconsistencies that may arise from the integration of different parts of the software or system. Integration testing is essential for ensuring the functionality, performance, and compatibility of the software or system as a whole, and for identifying and resolving any integration issues early in the development process. Skipping integration testing would increase the risk of finding serious defects later in the test process, orworse, in the production environment, which would be more costly and difficult to fix, and could damage the reputation and credibility of the organization. Therefore, the correct answer is D.
The other options are incorrect because:
? A. It is not allowed as organizations can decide on the test levels to do depending on the context of the system under test. While it is true that the choice and scope of test levels may vary depending on the context of the system under test, such as the size, complexity, criticality, and risk level of the system, the organization cannot simply ignore or skip a test level that is defined and required by the chosen software development life cycle model. The organization must follow the principles and guidelines of the software development life cycle model, and ensure that the test levels are consistent and coherent with the development phases. If the organization wants to have more flexibility and adaptability in choosing the test levels, it should consider using a different software development life cycle model, such as an agile or iterative model, that allows for more dynamic and incremental testing approaches.
? B. It is not allowed because integration testing is not an important test level and can be dispensed with. This statement is false and misleading, as integration testing is a very important test level that cannot be dispensed with. Integration testing is vital for testing the interactions and interfaces between components or subsystems, and for ensuring the functionality, performance, and compatibility of the software or system as a whole. Integration testing can reveal defects or inconsistencies that may not be detected by component (unit) testing alone, such as interface errors, data flow errors, integration logic errors, or performance degradation. Integration testing can also help to verify and validate the architectural design and the integration strategy of the software or system, and to ensure that the software or system meets the specified and expected quality attributes, such as reliability, usability, security, and maintainability. Integration testing can

also provide feedback and confidence to the developers and stakeholders about the progress and quality of the software or system development. Therefore, integration testing is a crucial and indispensable test level that should not be skipped or omitted.
? C. It is not allowed because integration testing is a very important test level and ignoring it means definite poor product quality. This statement is partially true, as integration testing is a very important test level that should not be ignored, and skipping it could result in poor product quality. However, this statement is too strong and absolute, as it implies that integration testing is the only factor that determines the product quality, and that ignoring it would guarantee a poor product quality. This is not necessarily the case, as there may be other factors that affect the product quality, such as the quality of the requirements, design, code, and other test levels, the effectiveness and efficiency of the test techniques and tools, the competence and experience of the developers and testers, the availability and adequacy of the resources and environment, the management and communication of the project, and the expectations and satisfaction of the customers and users. Therefore, while integration testing is a very important test level that should not be skipped, it is not the only test level that matters, and skipping itdoes not necessarily mean definite poor product quality, but rather a higher risk and likelihood of poor product quality.
References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.3, pages 16-18; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 38-39; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 104, page 36.

NEW QUESTION 21
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:

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

Answer: C

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

NEW QUESTION 22
Which of the following statements about white-box test techniques is true?

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

Answer: D

Explanation:
This answer is correct because code-related white-box test techniques are test design techniques that use the structure of the code to derive test cases. They provide an objective measure of coverage, such as statement coverage, branch coverage, or path coverage, which indicate how much of the code has been exercised by the test cases. Code-related white-box test techniques can be used to complement black-box test techniques, which are test design techniques that use the functional or non-functional requirements of the system or component to derive test cases. By combining both types of techniques, testers can increase their confidence in the code and find more
defects. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level
Syllabus v4.0, Section 2.3.2.2

NEW QUESTION 24
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?

A. 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.i
B. 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.ii
C. 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.I
D. Click on the checkout button, select the payment method, make payment, confirmation received of payment and shipping date.
E. 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
F. I, ii and v
G. iv
H. i and iii
I. v

Answer: C

Explanation:
Reference:ISTQB CTFL Syllabus V4.0, Section 5.2.2

NEW QUESTION 26
Which of the following coverage criteria results in the highest coverage for state transition based test cases?

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

D. Covering all states at least once

Answer: B

Explanation:
Covering all transitions at least once is the highest coverage criterion for state transition based test cases, because it ensures that every possible change of state is tested at least once. This means that all the events that trigger the transitions, as well as the actions and outputs that result from the transitions, are verified. Covering all transitions at least once also implies covering all states at least once, but not vice versa. Therefore, option D is not the highest coverage criterion. Option C is the lowest coverage criterion, because it only tests the initial and final states of the system or component, without checking the intermediate states or transitions. Option A is incorrect, because the coverage criteria for state transition based test cases can be determined and compared based on the number of transitions and states covered. References = CTFL 4.0 Syllabus, Section 4.2.3, page 49-50.

NEW QUESTION 31
You are testing the latest version of an air-traffic control system prior to production deployment using exploratory testing. After following an unusual sequence of input steps, the system crashes. After the crash, you document a defect report with the following information:
•Title: System crashes unexpectedly during input.
•Brief summary: System crashes when an unusual sequence of inputs is used.
•Version: V1.001
•Test: Exploratory testing prior to production deployment
•Priority: Urgent
•Risk: High
•References: Screenshot of crashed application What critical Information Is missing from this report?

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

Answer: C

Explanation:
The critical information missing from the defect report is a detailed description of the defect to enable reproduction. A clear and concise description of the steps taken to reproduce the defect is essential for developers to understand the context and to be able to replicate the issue in their environment. Without this information, it can be challenging to diagnose and fix the defect. The ISTQB CTFL syllabus emphasizes the importance of providing all necessary details in a defect report to facilitate effective communication and resolution.
References:ISTQB CTFL Syllabus, Section 5.5, "Defect Management."

NEW QUESTION 35
Which of the following is a factor that contributes to a successful review?

A. All participants in the review are aware they will be evaluated based on the defects they will find
B. The author of the work product to be reviewed leads the review meeting.
C. All participants in the review are trained to deal with the review type and its objectives.
D. Review metrics must be collected to improve the review process

Answer: C

Explanation:
A successful review process involves all participants being trained in the review type and understanding its objectives. This ensures that everyone can contribute effectively and understand what is expected from the review. Proper training helps to identify defects accurately and facilitates constructive feedback, leading to a more efficient and effective review process. Hence, statement C is correct according to the ISTQB CTFL syllabus.

NEW QUESTION 36
In addition to thorough testing of the requirements specification, a development team aims to involve users as early as possible in the development process, using practices such as prototyping, to ensure that the software systems being developed will meet the users' expectations. This approach is especially useful at mitigating the risks associated with one of the seven testing principles, which one?

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

Answer: B

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

NEW QUESTION 40
A test status report SHOULD:

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

Answer: A

Explanation:
A test status report is a document that provides a snapshot of the testing activities and their progress during a particular period. It should include information about any impediments encountered during the test execution and the actions taken to resolve them, which helps stakeholders understand the challenges and how they were addressed .
Option B describes an activity related to test completion rather than ongoing status reporting. Option C is incorrect because the structure and contents of the report may vary based on the audience's needs. Option D, while important, is not the primary purpose of a test status report, which focuses more on the current status and impediments.

NEW QUESTION 44
A program is used to control a manufacturing line (turn machines on and off. start and stop conveyer belts, add raw materials to the flow. etc.). Not all actions are possible at all times. For example, there are certain manufacturing stages that cannot be stopped - unless there is an emergency. A tester attempts to evaluate if all such cases (where a specific action is not allowed) are covered by the tests.
Which coverage metric will provide the needed information for this analysis?

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

Answer: D

Explanation:
Branch coverage is a type of structural coverage metric that measures the percentage of branches or decision outcomes that are executed by the test cases. A branch is a point in the code where the control flow can take two or more alternative paths based on a condition. For example, an if-else statement is a branch that can execute either the if-block or the else-block depending on the evaluation of the condition. Branch coverage ensures that each branch is taken at least once by the test cases, and thus reveals the behavior of the software under different scenarios. Branch coverage is also known as decision coverage or all-edges coverage.
Branch coverage is suitable for testing the cases where a specific action is not allowed, because it can verify that the test cases cover all the possible outcomes of the conditions that determine the action. For example, if the program has a condition that checks if the manufacturing stage can be stopped, then branch coverage can ensure that the test cases cover both the cases where the stage can be stopped and where it cannot be stopped. This way, branch coverage can help identify any missing or incorrect branches that may lead to undesired or unsafe actions.
The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types of coverage metrics, such as statement coverage, branch coverage, data flow coverage, etc. Code coverage does not specify which type of coverage metric is used for the analysis. Data flow coverage is a type of structural coverage metric that measures the percentage of data flow paths that are executed by the test cases. A data flow path is a sequence of statements that define, use, or kill a variable. Data flow coverage is useful for testing the correctness and completeness of the data manipulation in the software, but not for testing the conditions that determine the actions. Statement coverage is a type of structural coverage metric that measures the percentage of statements or lines of code that are executed by the test cases. Statement coverage ensures that each statement is executed at least once by the test cases, but it does not reveal the behavior of the software under different scenarios. Statement coverage isa weaker criterion than branch coverage, because it does not account for the branches or decision outcomes in the code. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

NEW QUESTION 46
Which of the following best describes the relationship between a test progress report and a test summary report?

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

Answer: A

Explanation:
Reference:ISTQB CTFL Syllabus V4.0, Section 5.3.2

NEW QUESTION 47
Consider a review for a high-level architectural document written by a software architect. The architect does most of the review preparation work, including distributing the document to reviewers before the review meeting. However, reviewers are not required to analyze the document in advance, and during the review meeting the software architect explains the document step by step. The only goal of this review is to establish a common understanding of the software architecture that will be used in a software development project.
Which of the following review types does this review refer to?

A. Inspection
B. Audit
C. Walkthrough
D. Informal review

Answer: C

Explanation:
This answer is correct because a walkthrough is a type of review where the author of the work product leads the review process and explains the work product to the reviewers. The reviewers are not required to prepare for the review in advance, and the main objective of the walkthrough is to establish a common understanding of the work product and to identify any major defects or issues. A walkthrough is usually informal and does not follow a defined process or roles. In this case, the review for a high-level architectural document written by a software architect matches the characteristics of a walkthrough. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.4.2.2

NEW QUESTION 49
Which of the following statements is TRUE'?

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

Answer: D

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

NEW QUESTION 51
Test automation allows you to:

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

Answer: B

Explanation:
Test automation allows you to produce tests that are less subject to human errors, as they can execute predefined test scripts or test cases with consistent inputs, outputs, and expected results. Test automation can also reduce the manual effort and time required to execute repetitive or tedious tests, such as regression tests, performance tests, or data- driven tests. Test automation does not demonstrate the absence of defects, as it can only verify the expected behavior of the system under test, not the unexpected or unknown behavior. Test automation does not avoid performing exploratory testing, as exploratory testing is a valuable technique to discover new information, risks, or defects that are not covered by automated tests. Test automation does not increase test process efficiency by facilitating management of defects, as defect management is a separate activity that involves reporting, tracking, analyzing, and resolving defects, which may or may not be related to automated tests. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 3.3.1, Test
Automation1
? ISTQB® Glossary of Testing Terms v4.0, Test Automation2

NEW QUESTION 56
Which of the following is a task the Author is responsible for, as part of a typical formal review?

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

Answer: C

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

NEW QUESTION 60
Which of the following statements is true?

A. A defect does not always produce a failure, while a bug always produces a failure
B. A defect may cause a failure which, when occurring, always causes an error
C. Failures can be caused by defects, but also by environmental conditions
D. Bugs are defects found during component testing, while failures are defects found athigher test levels

Answer: C

Explanation:
Failures can be caused by defects, but also by environmental conditions. A failure is an event in which the software system does not perform a required function or performs a function incorrectly, according to the expected behavior. A defect is a flaw in the software system or a deviation from the requirements or the specifications, that may cause a failure. However, not all failures are caused by defects, as some failures may be caused by environmental conditions, such as hardware malfunctions, network interruptions, power outages, incompatible configurations, etc. Environmental conditions are factors that affect the operation of the software system, but are not part of the software system itself. The other statements are false, because:
? A defect does not always produce a failure, while a bug always produces a failure.
This statement is false, because a defect may or may not produce a failure, depending on the inputs, the outputs, the states, or the scenarios of the software system, and a bug is just another term for a defect, so it has the same possibility of producing a failure as a defect. For example, a defect in a rarely used feature or a hidden branch of the code may never produce a failure, while a defect in a frequently used feature or a critical path of the code may produce a failure often. A bug is not a different concept from a defect, but rather a synonym or a colloquial term for a defect, so it has the same definition and implications as a defect.
? A defect may cause a failure which, when occurring, always causes an error. This
statement is false, because an error is not a consequence of a failure, but rathera cause of a defect. An error is a human action or a mistake that produces a defect in the software system, such as a typo, a logic flaw, a requirement misunderstanding, etc. An error is not observable in the software system, but rather in the human mind or the human work products, such as the code, the design, the documentation, etc. A failure is not a cause of an error, but rather a result of a defect, which is a result of an error. For example, an error in the code may cause a defect in the software system, which may cause a failure in the software behavior.

? Bugs are defects found during component testing, while failures are defects found at higher test levels. This statement is false, because bugs and failures are not different types of defects, but rather different terms for defects and their manifestations. As mentioned before, bugs are just another word for defects, and failures are the events in which the software system does not perform as expected due to defects. Bugs and failures can be found at any test level, not only at component testing or higher test levels. Test levels are the stages of testing that correspond to the levels of integration of the software system, such as component testing, integration testing, system testing, and acceptance testing. Defects and failures can occur and be detected at any test level, depending on the test objectives, the test basis, the test techniques, and the test environment. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles1
? ISTQB® Glossary of Testing Terms v4.0, Failure, Defect, Bug, Environmental Condition, Error, Test Level2

NEW QUESTION 61
What type of testing measures its effectiveness by tracking which lines of code were executed by the tests?

A. Acceptance testing
B. Structural testing
C. Integration testing
D. Exploratory testing

Answer: B

Explanation:
Structural testing is a type of testing that measures its effectiveness by tracking which lines of code were executed by the tests. Structural testing, also known as white-box testing or glass-box testing, is based on the internal structure, design, or implementation of the software. Structural testing aims to verify that the software meets the specified quality attributes, such as performance, security, reliability, or maintainability, by exercising the code paths, branches, statements, conditions, or data flows. Structural testing uses various coverage metrics, such as function coverage, line coverage, branch coverage, or statement coverage, to determine how much of the code has been tested and to identify any untested or unreachable parts of the code. Structural testing can be applied at any level of testing, such as unit testing, integration testing, system testing, or acceptance testing, but it is more commonly used at lower levels, where the testers have access to the source code.
The other options are not correct because they are not types of testing that measure their effectiveness by tracking which lines of code were executed by the tests. Acceptance testing is a type of testing that verifies that the software meets the acceptance criteria and the user requirements. Acceptance testing is usually performed by the end-users or customers, who may not have access to the source code or the technical details of the software. Acceptance testing is more concerned with the functionality, usability, or suitability of the software, rather than its internal structure or implementation. Integration testing is a type of testing that verifies that the software components or subsystems work together as expected. Integration testing is usually performed by the developers or testers, who may use both structural and functional testing techniques to check the interfaces, interactions, or dependencies between the componentsor subsystems. Integration testing is more concerned with the integration logic, data flow, or communication of the software, rather than its individual lines of code. Exploratory testing is a type of testing that involves simultaneous learning, test design, and test execution. Exploratory testing is usually performed by the testers, who use their creativity, intuition, or experience to explore the software and discover any defects, risks, or opportunities for improvement. Exploratory testing is more concerned with the behavior, quality, or value of the software, rather than its internal structure or implementation. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54; Chapter 1: Fundamentals of Testing, Section 1.4: Testing Throughout the Software Development Lifecycle, Pages 11-13; Chapter 3: Static Testing, Section 3.4: Exploratory Testing, Pages 40-41.

NEW QUESTION 63
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:

A. Test planning.
B. Test analysis.
C. Test design.
D. Test monitoring and control.

Answer: D

Explanation:
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.

NEW QUESTION 67
Which of the following statements refers to good testing practice to be applied regardless of the chosen software development model?

A. Tests should be written in executable format before the code is written and should act as executable specifications that drive coding
B. Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level
C. Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly
D. Involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle

Answer: D

Explanation:
The statement that refers to good testing practice to be applied regardless of the chosen software development model is option D, which says that involvement of testers in work product reviews should occur as early as possible to take advantage ofthe 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 ensurethat the software system is ready and stable for the next level of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be relevant, flexible, or efficient for testing in some contexts or situations, such as when the software development and testing activities are performed in an iterative and incremental order, with frequent changes, feedback, and adaptations, as in agile software development models, such as Scrum, Kanban, or XP, when the test levels are not clearly defined or distinguished, or when the test levels are performed in parallel or concurrently, etc.
? Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly: This is a specific testing practice that is associated with uniform software development models, such as the spiral model, the incremental model, or the prototyping model, in which the software development and testing activities are performed in a cyclical and repetitive manner, with similar phases, deliverables, and processes. Test objectives are the goals or the purposes of testing, which can vary depending on the test level, the test type, the test technique, the test environment, the test stakeholder, etc. Test objectives can be defined in terms of the test basis, the test coverage, the test quality, the test risk, the test cost, the test time, etc. Test objectives should be specific, measurable, achievable, relevant, and time-bound, and they should be aligned with the project objectives and the quality characteristics. In uniform software development models, the test objectives may be the same for all test levels, as the testing process is repeated for each cycle or iteration, with similar focus, scope, and perspective of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be appropriate, realistic, or effective for testing in some contexts or situations, such as when the software development and testing activities are performed in a hierarchical and modular manner, with different phases, deliverables, and dependencies, as in sequential software development models, such as the waterfall model, the V-model, or the W-model, when the test objectives vary according to the test levels, such as component testing, integration testing, system testing, and acceptance testing, or when the test objectives change according to the feedback, the learning, or the adaptation of the testing process, as in agile software development models, such as Scrum, Kanban, or XP, etc.References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.2, Test Monitoring and Control1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.4, Test Implementation1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.5, Test Execution1
? ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.6, Test Closure1
? ISTQB® Glossary of Testing Terms v4.0, Work Product Review, Static Testing, Early Testing, Test-driven Development, Test Level, Entry Criterion, Exit Criterion, Test Objective, Test Basis, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time2

NEW QUESTION 69
A requirement specifies that if the total amount of sales (TAS) made during the year by a corporate seller is 300,000€ or more, the bonus that must be paid to the seller is 100% of a certain amount agreed upon at the beginning of the year. The software contains a fault as it implements this requirement with the decision "IF (TAS = 300,000)" instead of "IF (TAS >= 300.000)". The application of the 3-value boundary value analysis to this problem consists of the following three test cases (TAS is an integer variable):
TC1 = 299,999 TC2=300,000 TC=300,001
Which of the following statements is TRUE?

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

Answer: B

Explanation:
The requirement specifies that a bonus should be paid if the total amount of sales (TAS) made during the year is 300,000€ or more. The software incorrectly implements this requirement with "IF (TAS = 300,000)" instead of "IF (TAS >= 300,000)". Using boundary value analysis (BVA), which is a common technique in software testing, the three test cases provided (TC1 = 299,999, TC2 = 300,000, and TC3 = 300,001) cover the critical boundary values around the condition.
? TC1 tests just below the boundary (299,999),
? TC2 tests exactly at the boundary (300,000),
? TC3 tests just above the boundary (300,001).
Since the software incorrectly checks for TAS equal to 300,000, only TC2 will fail because the condition is exactly met and highlights the incorrect implementation of the decision logic.
Reference: ISTQB CTFL Syllabus V4.0, Chapter 4.2.2, Boundary Value Analysis (BVA).

NEW QUESTION 72
Which of the following statements about white-box testing is FALSE?

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

Answer: C

Explanation:

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

NEW QUESTION 77
A new web app aims at offering a rich user experience. As a functional tester, you have run some functional tests to verify that, before releasing the app, such app works correctly on several mobile devices, all of which are listed as supported devices within the requirements specification. These tests were performed on stable and isolated test environments where you were the only user interacting with the application. All tests passed, but in some of those tests you observed the following issue: on some mobile devices only, the response time for two web pages containing images was extremely slow.
Based only on the given information, which of the following recommendation would you follow?

A. You should open a defect report providing detailed information on which devices and by running which tests you observed the issue
B. The issue is related to performance efficiency, not functionalit
C. Thus, as a functional tester, you should not open any defect report as all the functional tests passed
D. You should not open any defect report as the problem is most likely due to poor hardware equipment on the devices where you observed the issue
E. You should not open any defect report and inform the test manager that the devices on which you observed the issue should no longer be supported so that they will be removed from the requirements specification

Answer: A

Explanation:
As a functional tester, you should open a defect report providing detailed information on which devices and by running which tests you observed the issue. A defect report is a document that records the occurrence, nature, and status of a defect detected during testing, and provides information for further investigation and resolution. A defect report should include relevant information such as the defect summary, the defectdescription, the defect severity, the defect priority, the defect status, the defect origin,
the defect category, the defect reproduction steps, the defect screenshots, the defect attachments, etc. Opening a defect report is a good practice for any tester who finds a defect in the software system, regardless of the type or level of testing performed. The other options are not recommended, because:
? The issue is related to performance efficiency, not functionality, but that does not mean that as a functional tester, you should not open any defect report as all the functional tests passed. Performance efficiency is a quality characteristic that measures how well the software system performs its functions under stated conditions, such as the response time, the resource utilization, the throughput, etc. Performance efficiency is an important aspect of the user experience, especially for web applications that run on different devices and networks. Even if the functional tests passed, meaning that the software system met the functional requirements, the performance issue observed on some devices could still affect the user satisfaction, the usability, the reliability, and the security of the software system. Therefore, as a functional tester, you have the responsibility to report the performance issue as a defect, and provide as much information as possible to help the developers or the performance testers to investigate and resolve it.

NEW QUESTION 80
To be able to define testable acceptance criteria, specific topics need to be addressed. In the table below are the topics matched to an incorrect description. Match the topics (the left column) with the correct description (the right column)

A. Mastered
B. Not Mastered

Answer: A

NEW QUESTION 84
Which one of the following statements relating to the benefits of static testing is NOT correct?

A. Static testing enables early detection of defects before dynamic testing is performed.
B. Static testing reduces testing costs and time.
C. Static testing increases development costs and time.
D. Static testing identifies defects which are not easily found by dynamic testing.

Answer: C

Explanation:
The statement that "static testing increases development costs and time" is NOT correct. Static testing actually helps to reduce development costs and time by identifying defects early in the development process before dynamic testing is performed. Early detection of defects reduces the cost and effort required to fix them and prevents the propagation of defects to later stages, thus reducing overall testing and development costs. References:ISTQB CTFL Syllabus, Section 3.1.2, "The Value of Static Testing."

NEW QUESTION 88
Consider the following examples of risks identified in different software development projects:
[I]. The contrast color ratio for both normal text and large text of a website does not comply with the applicable accessibility guidelines, making it difficult for many users to read the content on the pages
[II]. A development vendor fails to deliver their software system on time, causing significant delays to system integration testing activities that have been planned as part of a development project for a system of systems
[III]. People in the test team do not have sufficient skills to automate tests at the test levels required by the test automation strategy which does not allow production of an effective regression test suite
[IV]. In a web application, data from untrusted sources is not subject to proper input validation, making the application vulnerable to several security attacks
Which of the following statements is true?

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

Answer: B

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

NEW QUESTION 90
Select which of the following statements describe the key principles of software testing?

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

Answer: A

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

NEW QUESTION 92
Confirmation testing is performed after:

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

Answer: D

Explanation:
Confirmation testing is performed after a defect is fixed, and if such testing is successful then the regression tests that are relevant for such fix can be executed. Confirmation testing, also known as re-testing, is the process of verifying that a defect has been resolved by running the test case that originally detected the defect. Confirmation testing is usually done before regression testing, which is the process of verifying that no new defects have been introduced in the software as a result of changes or fixes. Therefore, option D is the correct answer.
References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 2.4.1, page 28; ISTQB® Glossary v4.02, page 15.

NEW QUESTION 97
Which of the following statements best describes the way in which decision coverage is measured?
A. Measured as the number of statements executed by the tests, divided by the total number of executable statements in the code.
B. Measured as the number of lines of code executed by the tests, divided by the total number of lines of code in the test object.
C. Measured as the number of decision outcomes executed by the tests, divided by the total number of decision outcomes in the test object.
D. It is not possible to accurately measure decision coverage.

Answer: C

Explanation:
Reference:ISTQB CTFL Syllabus V4.0, Section 4.3.2

NEW QUESTION 100
Which of the following issues cannot be identified by static analysis tools?

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

Answer: A

Explanation:
Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standardsviolations, security vulnerabilities, and potential bugs12. However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)3. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures. MTBF can only be estimated by dynamic testing, which involves

executing the program under various conditions and collecting data on its failures4. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools. The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can be identified by static analysis tools. Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false5. Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables6. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption. References = What Is Static Analysis? Static Code Analysis Tools - Perforce Software, How Static Code Analysis Works | Perforce, Static CodeAnalysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs - CareerRide, ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

NEW QUESTION 104
Which of the following answers describes a reason for adopting experience-based testing techniques?

A. Experience-based test techniques provide more systematic coverage cnteria than black- box and white-box test techniques
B. Experience-based test techniques completely rely on the tester's past experience for designing test cases.
C. Experience-based test techniques allow designing test cases that are usually easier to reproduce than those designed with black-box and white-box test techniques.
D. Experience-based test techniques tend to find defects that may be difficult to find with black-box and white-box test techniques and are often useful to complement these more systematic techniques.

Answer: D

Explanation:
Experience-based testing techniques leverage the tester's intuition and prior experience to identify defects that systematic techniques might miss. These techniques are valuable because they can uncover issues based on real-world usage and scenarios that aren't always covered by more formalized black-box and white-box methods. The ISTQB CTFL Syllabus v4.0 highlights the complementary nature of experience-based techniques in providing a broader defect detection strategy.

NEW QUESTION 106
From a testing perspective, configuration management

A. Allows the expected results to be compared with the actual results.
B. Allows the tracking of all changes to versions of the testware.
C. Includes all activities that direct and control an organisation with regard to quality
D. Focuses on configuring static analysis tools to choose the most suitable breadth and depth of analysis.

Answer: B

Explanation:
Configuration management in the context of testing involves the systematic control of changes to the configuration items, including testware such as test scripts, test data, and test environments. It ensures that all changes are tracked and recorded, enabling the version control and management of testware .
Option A is related to test execution rather than configuration management. Option C describes quality management in a broader sense, not specifically configuration management. Option D is specific to the configuration of tools, not the overall management of testware versions.

NEW QUESTION 111
Which of the following about typical information found within a test plan is FALSE?

A. The need to temporarily have additional test personnel available for specific test phases and/or test activities
B. The conditions that must be met in order for the test execution activities to be considered completed.
C. The list of the product risks which have not been fully mitigated at the end of test execution.
D. The conditions that must be met for part of all the planned activities to be suspended and resumed.

Answer: C

Explanation:
A typical test plan includes various elements, such as resource requirements, test completion criteria, and suspension/resumption criteria. However, the list of product risks that have not been fully mitigated is generally not included in the test plan but rather in the risk management documentation.
? The test plan focuses on planning and executing tests, including resource
allocation and defining criteria for test suspension and resumption.
? While risk management is crucial, unmitigated risks are typically documented in risk logs or separate risk management plans.
Reference: ISTQB CTFL Syllabus V4.0, Chapter 5.1.1, Test Planning.

NEW QUESTION 116
The whole-team approach:

A. promotes the idea that all team members should have a thorough understanding of test techniques
B. is a consensus-based approach that engages the whole team in estimating the user stories
C. promotes the idea that all team members should be responsible for the quality of the product
D. is mostly adopted in projects aimed at developing safety-critical systems, as it ensures the highest level of testing independence

Answer: C

Explanation:
This answer is correct because the whole-team approach is a way of working in agile projects where all team members share the responsibility for the quality of the product, and collaborate on delivering value to the customer. The whole-team approach involves testers, developers, business analysts, product owners, and other stakeholders in planning, designing, developing, testing, and delivering the product. The whole-team approach fosters communication, feedback, learning, and continuous improvement within the
team. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.1

NEW QUESTION 121
A software company decides to invest in reviews of various types. The thought process they have is that each artifact needs to be reviewed using only one of the review methods depending on the criticality of the artifact.

A. The thought process is incorrec
B. The whole company should adopt same standard for review of all artifacts.
C. The thought process is correc
D. The whole company should decide or the review method based on their CMM level.
E. The thought process is incorrec
F. Same artifact can be reviewed using different review methods
G. The thought process is correc
H. It wastes time to review same artifact using efferent review methods

Answer: C

Explanation:
The thought process of the software company is incorrect, because it assumes that each artifact can be reviewed using only one review method, and that the review method depends solely on the criticality of the artifact. This is a simplistic and rigid approach that does not consider the benefits and limitations of different review methods, the context and purpose of the review, and the feedback and improvement opportunities that can be gained from multiple reviews. According to the CTFL 4.0 Syllabus, the selection of review methods should be based on several factors, such as the type and level of detail of the artifact, the availability and competence of the reviewers, the time and budget constraints, the expected defects and risks, and the desired outcomes and quality criteria. Moreover, the same artifact can be reviewed using different review methods at different stages of the development lifecycle, to ensure that the artifact meets the changing requirements, standards, and expectations of the stakeholders. For example, a requirement specification can be reviewed using an informal review method,such as a walkthrough, to get an initial feedback from the users and developers, and then using a formal review method, such as an inspection, to verify the completeness, correctness, and consistency of the specification. Therefore, the software company should adopt a more flexible and context-sensitive approach to selecting and applying review methods for different artifacts, rather than following a fixed and arbitrary rule. References = CTFL 4.0 Syllabus, Section 3.2.1, page 31-32; Section 3.2.2, page 33-34; Section 3.2.3, page 35-36.

NEW QUESTION 124
Which of the following statements about static testing and dynamic testing is true?

A. Unlike dynamic testing, which can be also performed manually, static testing cannot be performed without specialized tools
B. Static testing is usually much less cost-effective than dynamic testing
C. Unlike dynamic testing, which focuses on detecting potential defects, static testing focuses on detecting failures which may be due to actual defects
D. Both static testing and dynamic testing can be used to highlight issues associated with non-functional characteristics

Answer: D

Explanation:
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 softwareproduct. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.1.1, Section 2.2.1.2

NEW QUESTION 128
Which one of the following statements IS NOT a valid objective of testing?

A. To build confidence in the level of quality of the test object.
B. To find all defects in a product, ensuring the product is defect free.
C. To find failures and defects
D. To evaluate work products such as requirements, user stories, design, and code.

Answer: B

Explanation:
Reference:ISTQB CTFL Syllabus V4.0, Section 1.1.1

NEW QUESTION 130
The following decision table is used to assist a doctor in determining the drug therapy to prescribe for a patient (aged 6 to 65 years) diagnosed with acute sinusitis. The table consists of three Boolean conditions and six actions

	1	2	3	4	5	6	7	8
Conditions								
Is the patient over 18 years old?	F	F	F	F	T	T	T	T
Is the patient allergic to Penicillin?	F	F	T	T	F	F	T	T
Is the patient taking anticoagulant therapy?	F	T	F	T	F	T	F	T
Actions								
Amoxicillin is the therapy of choice					X			
Levofloxacin is the therapy of choice			X				X	
Cefuroxime is the therapy of choice	X							
Necessary consultation with the hematologist		X		X		X		X
Full dosage recommended for 10 days					X		X	
Half of the full recommended dosage for 10 days	X		X					

Based only on the given information, which of the following statements is TRUE?

A. Column 7 represents an impossible situation and thus can be deleted
B. Columns 1 and 3 can be merged into a single column
C. Columns 2, 4, 6 and 8 can be merged into a single column
D. Columns 5 and 7 can be merged into a single column

Answer: B

Explanation:
Decision tables are used to model complex decision logic by considering different combinations of conditions and actions. Based on the given decision table for prescribing drug therapy:
? Column 1 and Column 3 both result in the same actions (prescribing Amoxicillin).
? These columns can be merged because the actions taken do not depend on
whether the patient is taking anticoagulant therapy (both are 'T' for this condition). Thus, combining these columns simplifies the decision table without losing any information. Reference: ISTQB CTFL Syllabus V4.0, Chapter 4.2.3, Decision Table Testing.

NEW QUESTION 133
You are testing a room upgrade system for a hotel. The system accepts three differed types of room (increasing order of luxury): Platinum. Silver and Gold Luxury. ONLY a Preferred Guest Card holder s eligible for an upgrade.
Below you can find the decision table defining the upgrade eligibility:

Conditions				
Preferred Guest Card holder	YES	YES	NO	NO
Room Type	Silver	Platinum	Silver	Platinum
ons				
r upgrade to Gold Luxury	YES	NO	NO	NO
r upgrade to Silver	N/A	YES	N/A	NO

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

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

Answer: C

Explanation:
According to the decision table in the image, a Preferred Guest Card holder with a Silver room is eligible for an upgrade to Gold Luxury (YES), while a non-Preferred Guest Card holder, regardless of room type, is not eligible for any upgrade (NO).
Therefore, Customer A (a Preferred Guest Card holder with a Silver room) would be offered an upgrade to Gold Luxury, and Customer B (a non-Preferred Guest Card holder with a Platinum room) would not be offered any upgrade. References = The answer is derived directly from the decision table provided in theimage; specific ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents are not referenced.

NEW QUESTION 137
Metrics can be collected during and at the end of testing activities to assess which of the following?

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

Answer: D

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

NEW QUESTION 142
Which of the following statements is true?

A. Functional testing focuses on what the system should do while non-functional testing on the internal structure of the system
B. Non-functional testing includes testing of both technical and non-technical quality characteristics
C. Testers who perform functional tests are generally expected to have more technical skills than testers who perform non-functional tests
D. The test techniques that can be used to design white-box tests are described in the ISO/IEC 25010 standard

Answer: B

Explanation:
Non-functional testing includes testing of both technical and non-technical quality characteristics. Non-functional testing is the process of testing the quality attributes of a system, such as performance, usability, security, reliability, etc. Non-functional testing can be applied at any test level and can use both black-box and white-box test techniques. Non-functional testing can cover both technical aspects, such as response time, throughput, resource consumption, etc., and non-technical aspects, such as user satisfaction, accessibility, compliance, etc. Therefore, option B is the correct answer. References: ISTQB® Certified Tester Foundation Level Syllabus v4.01, Section 1.3.1, page 13; ISTQB® Glossary v4.02, page 40.

NEW QUESTION 144
A system has a self-diagnostics module that starts executing after the system is reset. The diagnostics are running 12 different tests on the systems memory hardware. The following is one of the requirements set for the diagnostics module:
'The time taking the diagnostics tests to execute shall be less than 2 seconds' Which of the following is a failure related to the specified requirement?

A. The diagnostic tests fail to start after a system reset
B. The diagnostic tests take too much time to execute
C. The diagnostic tests that measure the speed of the memory, fail
D. The diagnostic tests fail due to incorrect implementation of the test code

Answer: B

Explanation:
A failure is an event in which a component or system does not perform a required function within specified limits1. A requirement is a condition or capability needed by a user to solve aproblem or achieve an objective2. In this case, the requirement is that the diagnostics tests should execute in less than 2 seconds. Therefore, any event that violates this requirement is a failure. The only option that clearly violates this requirement is B. The diagnostic tests take too much time to execute. If the diagnostic tests take more than 2 seconds to complete, then theydo not meet the specified limit and thus fail. The other options are not necessarily failures related to the specified requirement. Option A. The diagnostic tests fail to start after a system reset is a failure, but not related to the time limit. It is related to the functionality of the self-diagnostics module. Option C. The diagnostic tests that measure the speed of the memory, fail is also a failure, but not related to the time limit. It is related to the accuracy of the memory tests. Option D. The diagnostic tests fail due to incorrect implementation of the test code is also a failure, but not related to
the time limit. It is related to the quality of the test code. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Requirements Engineering Fundamentals.

NEW QUESTION 148
Can "cost" be regarded as Exit criteria?

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

Answer: A

Explanation:
Cost can be regarded as an exit criterion for testing, because it is a factor that affects the profitability and feasibility of the software product. Testing is an investment that aims to improve the quality and reliability of the software product, but it also consumes resources, such as time, money, and human effort. Therefore, testing should be planned and executed in a way that balances the cost and benefit of testing activities. Having cost as an exit criterion helps to avoid spending too much money on testing, which mayresult in an unprofitable product or a loss of competitive advantage. Cost can also help to prioritize and focus the testing efforts on the most critical and valuable features and functions of the software product. However, cost should not be the only exit criterion for testing, as it may not reflect the true quality and risk level of the software product. Other exit criteria, such as defect rate, test coverage, user satisfaction, etc., should also be considered and defined in the test plan.
The other options are incorrect, because they either deny the importance of cost as an exit criterion, or they make false or unrealistic assumptions about the cost of testing. Option B is incorrect, because the financial value of product quality can be estimated, for example, by using cost-benefit analysis, return on investment, or cost of quality models. Option C is incorrect, because going by cost as an exit criterion does not necessarily constrain the testing project or help achieve the desired quality level. Cost is a relative and variable factor that depends on the scope, complexity, and context of the software product and the testing project. Option D is incorrect, because the cost of testing can be measured effectively, for example, by using metrics, such as test effort, test resources, test tools, test environment, etc.

NEW QUESTION 150
In which of the following test documents would you expect to find test exit criteria described9

A. Test design specification
B. Project plan
C. Requirements specification
D. Test plan

Answer: D

Explanation:
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 requirementsspecification 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. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Entry and Exit Criteria in Software Testing | Baeldung onComputer Science, Entry And Exit
Criteria In Software Testing - Rishabh Software, Entry and Exit Criteria in Software Testing Life Cycle - STLC [2022 Updated] - Testsigma Blog, ISTQB® releases Certified Tester Foundation Level v4.0 (CTFL).

NEW QUESTION 153
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

CTFL4 Practice Exam Features:

* CTFL4 Questions and Answers Updated Frequently

* CTFL4 Practice Questions Verified by Expert Senior Certified Staff

* CTFL4 Most Realistic Questions that Guarantee you a Pass on Your FirstTry

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