

ISTQB-CTFL Dumps

ISTQB-Foundation Level Exam

<https://www.certleader.com/ISTQB-CTFL-dumps.html>



NEW QUESTION 1

Consider the following testing levels:

- 1) Component Testing
- 2) Integration Testing
- 3) System Testing
- 4) Acceptance Testing

Which of the following statements is true?

- A. Integration and system testing are applicable when V-model is used. Component and acceptance testing are applicable when iterative development models are used.
- B. All the testing levels are applicable to V-model for software development.
- C. Only acceptance testing is applicable for iterative models.
- D. Acceptance testing is applicable for all software development model.
- E. Component and system testing are applicable only for the V-model.
- F. All testing levels are applicable, independent of which software development life-cycle process (V-model, iterative, incremental) is used.
- G. iterative, incremental) is used.

Answer: D

Explanation:

All testing levels are applicable, independent of which software development life-cycle process (V-model, iterative, incremental) is used. Testing levels are defined based on the scope and objectives of testing, not on the software development model. Component testing, integration testing, system testing and acceptance testing are common testing levels that can be applied to any software development model, as long as they are planned and executed properly. The V-model is a software development model that emphasizes the relationship between each development phase and its corresponding testing phase. Iterative and incremental models are software development models that divide the development process into smaller cycles or iterations, where each iteration produces a working version of the software that can be tested and evaluated. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 18.

NEW QUESTION 2

Which of the following statements regarding inspection is NOT true?

- A. An inspection may be led by a trained moderator who shall not be the author.
- B. The main purpose of an inspection is to find solutions to the problems.
- C. An inspection can be performed by peers.
- D. An inspection shall follow a formal process based on rules and checklists with entry and exit criteria.

Answer: B

Explanation:

An inspection is a type of review that follows a defined process with formal entry and exit criteria and roles and responsibilities for participants. An inspection can be performed by peers with different roles, such as moderator, author, reviewer and scribe. The following statement about inspection is not true:

? B) The main purpose of an inspection is to find solutions to the problems. This statement is not true, as the main purpose of an inspection is to find defects or issues in a work product, not to find solutions to the problems. Finding solutions to the problems is a debugging or problem-solving activity that is usually performed by the author or developer after receiving the inspection report. The following statements about inspection are true:

? A) An inspection may be led by a trained moderator who shall not be the author.

This statement is true, as an inspection requires a moderator role who leads the inspection process and ensures that it follows the rules and standards. The moderator should be trained in inspection techniques and should not be the author of the work product under inspection, in order to avoid bias or conflict of interest.

? C) An inspection can be performed by peers. This statement is true, as an inspection involves peer review, which means that the work product under inspection is evaluated by people who have similar roles or expertise as the author, but who are not directly involved in creating or modifying the work product.

? D) An inspection shall follow a formal process based on rules and checklists with entry and exit criteria. This statement is true, as an inspection follows a formal process that consists of six main steps: planning, kick-off meeting, individual preparation, review meeting, rework and follow-up. Each step has defined rules and checklists to guide the participants and ensure consistency and quality. Each step also has entry and exit criteria to ensure that the prerequisites and objectives are met before moving to the next step. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 28-29.

NEW QUESTION 3

Which of the following is a correct reason to apply test automation?

- A. When a new test automation tool is launched
- B. When there are a lot of repetitive testing tasks
- C. When it is easy to automate
- D. When it is cheap to buy test automation tools

Answer: B

Explanation:

A correct reason to apply test automation is when there are a lot of repetitive testing tasks. Test automation is the use of software tools or scripts to perform or support testing activities, such as test case execution, test result comparison, test data generation, etc. Test automation can be beneficial when there are a lot of repetitive testing tasks that need to be performed frequently or consistently, such as regression testing, performance testing, load testing, etc. Test automation can help to save time and effort, increase reliability and accuracy, and improve coverage and efficiency of testing. The other options are not correct reasons to apply test automation. When a new test automation tool is launched is not a reason to apply test automation, but rather a factor for choosing a test automation tool.

When it is easy to automate is not a reason to apply test automation, but rather a factor for evaluating the feasibility of test automation. When it is cheap to buy test automation tools is not a reason to apply test automation, but rather a factor for estimating the cost and benefit of test automation. Verified References: A Study Guide to the ISTQB®

Foundation Level 2018 Syllabus - Springer, page 10.

NEW QUESTION 4

For withdrawing money from an Automated Teller Machine (ATM), the following conditions are required:

- The bank card is valid

- The PIN code is correct
- Money is available in the user's account
The following are some possible interactions between the user and the ATM:
- The entered card is invalid The card is rejected
- The PIN code is wrong The ATM asks for another PIN code
- The requested amount is more than available in the user's account: The ATM asks for another amount
- The requested amount is available in the user's account The ATM dispenses the money Which test design technique should be used to cover all possible combinations of the in put conditions?

- A. Use case based testing
- B. Decision table
- C. Boundary value analysis
- D. Equivalence class partitioning

Answer: B

Explanation:

A decision table is a technique that should be used to cover all possible combinations of input conditions for withdrawing money from an Automated Teller Machine (ATM). A decision table shows combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects). A decision table consists of four quadrants: conditions (inputs), actions (outputs), condition entries (values) and action entries (results). A decision table can be used to test components that have multiple inputs and outputs that depend on logical combinations of conditions. For example, for testing the ATM, we can identify three input conditions: the bank card is valid, the PIN code is correct, and money is available in the user's account. We can also identify four output actions: the card is rejected, the ATM asks for another PIN code, the ATM asks for another amount, and the ATM dispenses the money. A decision table can show all possible combinations of these conditions and actions in a systematic way.

Use case based testing is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Use case based testing is a technique that verifies that a software product or system meets its specified requirements or user stories by executing realistic scenarios or workflows. Use case based testing can be used to test components that have complex or dynamic interactions with users or other systems. For example, for testing the ATM, we can identify several use cases, such as withdraw money, check balance, transfer money, etc. Each use case can have one or more scenarios that describe the steps and outcomes of the interaction. However, use case based testing may not cover all possible combinations of input conditions, as some scenarios may be omitted or overlooked.

Boundary value analysis is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Boundary value analysis is a technique that tests boundary values between partitions of equivalent data. Boundary values are values at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Boundary value analysis can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify boundary values for the input amount, such as the minimum and maximum amount allowed by the system or the user's account. However, boundary value analysis may not cover all possible combinations of input conditions, as some conditions may not have boundary values or may not be related to input values.

Equivalence class partitioning is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Equivalence class partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component. Equivalence class partitioning can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify equivalence partitions for the input amount, such as valid amount (within the range allowed by the system and the user's account) and invalid amount (outside the range allowed by the system or the user's account). However, equivalence class partitioning may not cover all possible combinations of input conditions, as some conditions may not be related to input values or may have more than two partitions. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 34-46.

NEW QUESTION 5

Which of the following statements about testing in the context of an agile (iterative- incremental) development model is correct?

- A. Unit test and acceptance test ate the most important tests to make sure that the system works as expected.
- B. Each iteration of testing has to be completely finished before a new Iteration of development starts.
- C. Regression testing is necessary whenever a new Increment Is added to the existing syste
- D. Only certain types of non-functional and explorative testing are performed.

Answer: C

Explanation:

In the context of agile (iterative-incremental) development models, testing is integrated into the development process and occurs continuously throughout the lifecycle of the project. Agile testing emphasizes adaptability and the need for feedback at various stages of development.

Option C is correct because regression testing is indeed necessary whenever a new increment is added to the existing system. Agile development often involves frequent changes and additions to the codebase, which can potentially introduce new defects into previously tested code. Regression testing ensures that new changes have not adversely affected existing functionality.

Options A, B, and D present misconceptions about agile testing:

? A is incorrect because, in agile, all types of testing (unit, integration, system, acceptance) are important and occur throughout the iteration, not just unit and acceptance tests.

? B is incorrect because agile methodologies advocate for continuous integration and testing, where development and testing activities overlap and support each other throughout an iteration.

? D is incorrect because agile methodologies encourage a wide range of testing types, including both functional and non-functional, as well as exploratory testing, to ensure a comprehensive quality assessment.

NEW QUESTION 6

Which of the following would be LEAST appropriate as part of an incident report covering the observation of a failure during testing?

- A. SOL injection into the username entry field allowed a variety of SQL commands to be executed by the application without the appropriate authority.
- B. The user interface was complicated and confusing and I found It quite difficult to follow the test script.
- C. The updates made as part of the add new member' function did not reflect the expected change as the name was written into the address field.
- D. The expected result for the 'list friends' response time was less than 10 seconds, whereas the average response time obtained was 13 seconds.

Answer: B

Explanation:

An incident report during testing should focus on factual observations of failures or defects in the system, including their impacts and how they deviate from expected results. Options A, C, and D describe specific issues that are directly related to the system's behavior or performance and are suitable for inclusion in an

incident report. Option B, which describes the user interface as "complicated and confusing" and relates to the tester's personal difficulty in following the test script, is more subjective and relates to the tester's experience rather than an objective observation of a system failure. Therefore, option B is the least appropriate for an incident report.

NEW QUESTION 7

The four test levels used in ISTQB syllabus are:

- * 1. Component (unit) testing
- * 2. Integration testing
- * 3. System testing
- * 4. Acceptance testing

An organization wants to do away with integration testing but otherwise follow V-model. Which of the following statements is correct?

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

Answer: D

Explanation:

The V-model is a software development life cycle model that defines four test levels that correspond to four development phases: component (unit) testing with component design, integration testing with architectural design, system testing with system requirements, and acceptance testing with user requirements. The V-model emphasizes the importance of verifying and validating each phase of development with a corresponding level of testing, and ensuring that the test objectives, test basis, and test artifacts are aligned and consistent across the test levels. Therefore, an organization that wants to follow the V-model cannot do away with integration testing, as it would break the symmetry and completeness of the V-model, and compromise the quality and reliability of the software or system under test. Integration testing is a test level that aims to test the interactions and interfaces between components or subsystems, and to detect any defects or inconsistencies that may arise from the integration of different parts of the software or system. Integration testing is essential for ensuring the functionality, performance, and compatibility of the software or system as a whole, and for identifying and resolving any integration issues early in the development process. Skipping integration testing would increase the risk of finding serious defects later in the test process, or worse, in the production environment, which would be more costly and difficult to fix, and could damage the reputation and credibility of the organization. Therefore, the correct answer is D.

The other options are incorrect because:

? A. It is not allowed as organizations can decide on the test levels to do depending on the context of the system under test. While it is true that the choice and scope of test levels may vary depending on the context of the system under test, such as the size, complexity, criticality, and risk level of the system, the organization cannot simply ignore or skip a test level that is defined and required by the chosen software development life cycle model. The organization must follow the principles and guidelines of the software development life cycle model, and ensure that the test levels are consistent and coherent with the development phases. If the organization wants to have more flexibility and adaptability in choosing the test levels, it should consider using a different software development life cycle model, such as an agile or iterative model, that allows for more dynamic and incremental testing approaches.

? B. It is not allowed because integration testing is not an important test level and can be dispensed with. This statement is false and misleading, as integration testing is a very important test level that cannot be dispensed with. Integration testing is vital for testing the interactions and interfaces between components or subsystems, and for ensuring the functionality, performance, and compatibility of the software or system as a whole. Integration testing can reveal defects or inconsistencies that may not be detected by component (unit) testing alone, such as interface errors, data flow errors, integration logic errors, or performance degradation. Integration testing can also help to verify and validate the architectural design and the integration strategy of the software or system, and to ensure that the software or system meets the specified and expected quality attributes, such as reliability, usability, security, and maintainability. Integration testing can also provide feedback and confidence to the developers and stakeholders about the progress and quality of the software or system development. Therefore, integration testing is a crucial and indispensable test level that should not be skipped or omitted.

? C. It is not allowed because integration testing is a very important test level and ignoring it means definite poor product quality. This statement is partially true, as integration testing is a very important test level that should not be ignored, and skipping it could result in poor product quality. However, this statement is too strong and absolute, as it implies that integration testing is the only factor that determines the product quality, and that ignoring it would guarantee a poor product quality. This is not necessarily the case, as there may be other factors that affect the product quality, such as the quality of the requirements, design, code, and other test levels, the effectiveness and efficiency of the test techniques and tools, the competence and experience of the developers and testers, the availability and adequacy of the resources and environment, the management and communication of the project, and the expectations and satisfaction of the customers and users. Therefore, while integration testing is a very important test level that should not be skipped, it is not the only test level that matters, and skipping it does not necessarily mean definite poor product quality, but rather a higher risk and likelihood of poor product quality.

References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.3, pages 16-18; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 38-39; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 104, page 36.

NEW QUESTION 8

Which of the following is the most important task of a typical test leader?

- A. To automate tests.
- B. To prepare and acquire test data.
- C. To set up the test environment.
- D. To coordinate the test strategy with project managers.

Answer: D

Explanation:

The most important task of a typical test leader is to coordinate the test strategy with project managers. The test strategy is a high-level document that defines the general approach and objectives of testing for a project or an organization. The test leader is responsible for defining, documenting, communicating, and implementing the test strategy in alignment with the project goals and constraints. The test leader also needs to coordinate with project managers and other stakeholders to ensure that the test strategy is feasible, effective, and efficient. The other options are not the most important tasks of a typical test leader. To automate tests is a task of a test automation engineer or a test automation specialist. To prepare and acquire test data is a task of a test analyst or a test engineer. To set up the test environment is a task of a test environment manager or a test environment specialist. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 13.

NEW QUESTION 9

Which of the following is a function of a dynamic analysis tool?

- A. Provide support for traceability of tests, test results and incidents to source documents

- B. Monitor the allocation, use and de-allocation of memory during run-time of a program
- C. Execute programs step-by-step in order to reproduce failures and find corresponding defects
- D. Provide support for release of baselines consisting of configuration items

Answer: B

Explanation:

A dynamic analysis tool is a tool that performs analysis of a software product based on its behavior during execution. A dynamic analysis tool can monitor various aspects of a program's run-time performance, such as memory usage, CPU load, response time, or resource leaks. A dynamic analysis tool can monitor the allocation, use and de- allocation of memory during run-time of a program, which can help detect defects such as memory leaks, buffer overflows, or memory corruption. A dynamic analysis tool cannot provide support for traceability of tests, test results and incidents to source documents, as this is a function of a test management tool. A dynamic analysis tool cannot execute programs step-by-step in order to reproduce failures and find corresponding defects, as this is a function of a debugging tool. A dynamic analysis tool cannot provide support for release of baselines consisting of configuration items, as this is a function of a configuration management tool. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 6, page 56-57.

NEW QUESTION 10

Given the following review types and review characteristics:

- * a. Pair review
- * b. Walkthrough
- * c. Technical review
- * d. Inspection
- * 1. Formal
- * 2. Informal
- * 3. Purposes include evaluating the quality of the work product under review and generating new ideas (e.g., brainstorming solutions)
- * 4. Purposes include Improving the software product and training the review participants Which of the following BEST matches the review type with the review characteristic?

- A. a-1. b-4, c-3. d-2
- B. a-4, b-3. c-2. d-1
- C. a-2, b-3, c-4, d-1
- D. a-2, b-4, c-3. d-1

Answer: C

Explanation:

Pair reviews are informal and typically involve two people reviewing the work product together, often in an informal setting. Walkthroughs are more formal and aim to educate stakeholders and evaluate the product, serving the dual purpose of improving the product and training participants. Technical reviews have a strong focus on improving the product's quality, often involving technical stakeholders. Inspections are the most formal review type and are aimed primarily at detecting defects. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2.3 "Review Types".

NEW QUESTION 10

Which of the following statements about Experience Based Techniques (EBT) is correct?

- A. EBT use tests derived from the test engineers' previous experience with similar technologies.
- B. EBT is based on the ability of the test engineer to implement various testing techniques.
- C. EBT is done as a second stage of testing, after non-experienced-based testing took place.
- D. EBT require broad and deep knowledge in testing but not necessarily in the application or technological domain.

Answer: A

Explanation:

Experience based techniques (EBT) are techniques that use the knowledge, intuition and skills of the test engineers to design and execute tests. EBT use tests derived from the test engineers' previous experience with similar technologies, domains, applications or systems. EBT are not based on the ability of the test engineer to implement various testing techniques, but rather on their personal judgment and creativity. EBT are not done as a second stage of testing, after non-experience-based testing took place, but rather as a complementary or alternative approach to other techniques. EBT require broad and deep knowledge in both testing and the application or technological domain, as this can help the test engineer identify potential risks, scenarios or defects. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 5, page 48-49.

NEW QUESTION 15

A test engineer finds a defect while testing. After the developer has fixed the defect, the test engineer decides to re-run a complete section of the tests. Which of the following is correct?

- A. The test engineer should not re-run the tests, as they have already been run, and results recorded.
- B. The test engineer should not re-run the tests, they should be part of the developer tests.
- C. The test engineer should re-run the tests, in order to ensure that new defects have not been introduced by the fix.
- D. The test engineer should re-run the tests, because the defect shows that the test cases need to be updated.

Answer: C

Explanation:

The test engineer should re-run the tests, in order to ensure that new defects have not been introduced by the fix. This is also known as regression testing, which is a type of testing that verifies that previously tested software still performs correctly after a change. Regression testing helps to detect any side effects or unintended consequences of a fix or a modification. The other options are incorrect reasons for re-running the tests. The test engineer should not re-run the tests, as they have already been run, and results recorded, because this ignores the possibility of new defects caused by the fix. The test engineer should not re-run the tests, they should be part of the developer tests, because this assumes that developer tests are sufficient and reliable, which may not be true. The test engineer should not re-run the tests, because the defect shows that the test cases need to be updated, because this does not address the impact of the fix on other test cases or functionalities. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 41.

NEW QUESTION 19

The following requirement is given "Set X to be the sum of Y and Z". All the following four implementations have bugs. Which one of the following bugs can be caught by Static Analysis?

- A. int x = 1. int y = 2. int y = 3.X = y=z;
- B. int x = 1. int y = 2. int z = 3.X = z-y
- C. int x = 1. Int y = 2. Int z = 3.Z = x +y
- D. int y = 2 Int z = 3. Y = z+y

Answer: A

Explanation:

Static analysis is a technique that analyzes the source code or other software artifacts without executing them. Static analysis can detect defects such as syntax errors, coding standards violations, potential security vulnerabilities, or logical flaws. Static analysis can catch the bug in the first implementation, as it contains two syntax errors: the variable y is declared twice, and the assignment statement X = y=z is invalid. Static analysis cannot catch the bugs in the other three implementations, as they are logical errors that do not violate any syntax rules, but produce incorrect results. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 3, page 25-26.

NEW QUESTION 23

A class grade application for instructors assigns letter grades based on students' numerical grades.

The letter grades for different numerical grades should be:

Above 89, up to 100 - A

Above 79, up to 89 • B

Above 69, up to 79 • C

Above 59, up to 69 - D

Below 60- F

Which of the following sets of test inputs would achieve the relatively highest equivalence partition coverage?

- A. 0, 58.59,70, 80
- B. 74, 79.84,85, 89
- C. 79, 89.90,99, 100
- D. 69, 79. 80, 89, 90

Answer: D

Explanation:

The set of test inputs that achieve the relatively highest equivalence partition coverage for grading students is option D: 69, 79, 80, 89, 90. This set effectively tests the boundaries between each grade category, ensuring that the grading system accurately transitions from one grade to another at the correct thresholds (ISTQB Main Web)

.References:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION 24

Given the following review process main activities and specific review activities:

- * a. Planning
- * b. Initiate review
- * c. Issue communication and analysis d.Fixing and reporting
- * 1. Creating defect reports
- * 2. Estimating effort and timeframe
- * 3. Recording updated status of defects
- * 4. Selecting the people to participate
- * 5. Distributing the work product and other material
- * 6. Evaluating the review findings

Which of the following BEST matches the review process main activities with the appropriate specific review activities?

- A. 2-a, 4-a, 5-b, 6-c, 1-d, 3-d
- B. 2-a, 5-a, 1-b, 4-b, 3-c, 6-d
- C. 1-a, 4-b, 5-b, 6-c, 2-d, 3-d
- D. 2-a, 4-b, 5-c, 1-
- E. 3-d, 6-d

Answer: A

Explanation:

Matching the main review process activities with the specific review activities, we see that planning includes estimating effort and timeframe (2) and selecting people to participate (4). Initiating a review involves distributing work products and other material (5). Issue communication and analysis includes evaluating the review findings (6). Fixing and reporting would entail creating defect reports (1) and recording the updated status of defects (3).References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2 "Review Process".

NEW QUESTION 25

Out of the following. what is not needed to specify in defect report?

- A. Test environment details
- B. How to reproduce the defect
- C. How to fix the defect
- D. Seventy and priority

Answer: C

Explanation:

A defect report is a document that records the details of a defect found during testing. A defect report typically contains the following items:

? Identifier: A unique identifier for the defect report
? Summary: A concise summary of the defect
? Description: A detailed description of the defect, including the steps to reproduce it, the expected and actual results, and any relevant screenshots or logs
? Severity: The degree of impact that the defect has on the system
? Priority: The level of urgency for resolving the defect
? Status: The current state of the defect, such as new, open, resolved, closed, etc.
? Resolution: The action taken to resolve the defect, such as fix, workaround, reject, etc. Out of these items, the one that is not needed to specify in a defect report is how to fix the defect. How to fix the defect is a technical solution that is usually determined by the developer who is assigned to resolve the defect. How to fix the defect is not part of the defect report, but rather part of the code change or patch that is delivered to fix the defect. The other items are needed to specify in a defect report, as they provide essential information for identifying, tracking and resolving defects. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 3, page 32-33.

NEW QUESTION 29

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 33

Which of the following is the BEST reason for selecting a particular type of software development lifecycle model?

- A. The project manager's preference
- B. Tester skill level with the software development lifecycle model
- C. The project team's overall familiarity with the model
- D. The type of product being developed

Answer: D

Explanation:

The choice of a software development lifecycle (SDLC) model is primarily influenced by the type of product being developed. Different products and project requirements may demand different SDLC models to address specific challenges and needs efficiently. For instance, a complex, safety-critical product might best be served by a Waterfall model due to its structured nature and phase dependencies, while a more iterative and incremental model might be suited for projects requiring frequent feedback and changes. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 2.1 "Software Development Lifecycles".

NEW QUESTION 34

In the newest version of payroll system number of changes were made. As a tester you got a task to perform regression and confirmation tests. Which of the following project activities are related to confirmation testing?

- A. Testing due to the application of a new version of the interface
- B. Testing that fixes resolved the defects in the search function
- C. Testing if a system still works after update of an operating system
- D. Testing to ensure the adding of a new functionalities haven't broken existing functions

Answer: B

Explanation:

Confirmation testing, also known as re-testing, is performed to verify that specific defects have been successfully fixed.

Option A: "Testing due to the application of a new version of the interface" would typically involve regression testing, not confirmation testing.

Option B: "Testing that fixes resolved the defects in the search function" fits the description of confirmation testing as it focuses on ensuring that specific issues have been addressed. Option C: "Testing if a system still works after update of an operating system" is an example of regression testing, as it checks the overall system behavior after an update. Option D: "Testing to ensure the adding of new functionalities haven't broken existing functions" is another example of re (ISTQB not-for-profit association) (Udemy) it checks for unintended consequences of new changes.

Therefore, the correct answer is B

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 39

Which of the following project scenario gives the BEST example where maintenance testing should be triggered?

- A. Completion of architecture of the bank system
- B. Release of the early draft of the low level project design of an IoT application
- C. Defect was found in a pre-released version of the customer service application
- D. Delivery of the hot fix to mobile operating system and ensuring that it still works

Answer: D

Explanation:

Maintenance testing is triggered by changes such as bug fixes, enhancements, or environmental changes.

Option A: "Completion of architecture of the bank system" is not a typical scenario for maintenance testing, as it describes a design phase rather than an operational change. Option B: "Release of the early draft of the low level project design of an IoT application" is again not suitable for maintenance testing, as it refers to the design phase.

Option C: "Defect was found in a pre-released version of the customer service application" is closer but not quite accurate, as maintenance testing focuses on changes mad (ISTQB not-for-profit association)system is released.

Option D: "Delivery of the hot fix to mobile operating system and ensuring that it still works" is the best example as it directly involves testing after a fix has been implemented. Therefore, the correct answer is D6†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 40

Which of the following statements contradicts the general principles of testing?

- A. Most defects are found in a small subset of a system's modules.
- B. If new defects are to be found we should run the same test set more often.
- C. Testing is better if it starts at the beginning of a project.
- D. How testing is done, is based on the situation in a particular project.

Answer: B

Explanation:

Statement B contradicts the general principles of testing, because running the same test set more often will not increase the chances of finding new defects, unless there are some changes in the system or environment that affect the test results. Running different test sets with different inputs, outputs or conditions would be more effective in finding new defects. Statements A, C and D are consistent with the general principles of testing. Statement A states that most defects are found in a small subset of a system's modules, which is true according to the defect clustering principle. Statement C states that testing is better if it starts at the beginning of a project, which is true according to the early testing principle. Statement D states that how testing is done, is based on the situation in a particular project, which is true according to the context-dependent testing principle. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, pages 4-6.

NEW QUESTION 43

Why is it important to select a test technique?

- A. There are usually loo many test cases that may be run on a syste
- B. Test techniques help reduce the number of tests.
- C. The only way to test a software application is by using well proven test techniques.
- D. Selecting the right test technique in a given situation Increases the effectiveness of the test process Oy creating tests with higher chance of finding bugs.
- E. Test techniques define the number of regression cycles, which in turn impact the project schedule.

Answer: C

Explanation:

Selecting the right test technique is crucial because different techniques are suited to different types of testing and can significantly increase the effectiveness of the testing process by creating tests that are more likely to find defects. While reducing the number of tests (A) and defining the number of regression cycles (D) are considerations in the testing process, they are not the primary reasons for selecting a test technique. The assertion that the only way to test a software application is by using well-proven test techniques (B) is too restrictive and does not acknowledge the adaptability required in testing to suit different contexts and objectives. Therefore, option C is the most comprehensive reason, as it focuses on the effectiveness and efficiency of testing, leading to the creation of high-quality tests that have a higher chance of finding bugs.

NEW QUESTION 47

Which of the following options cover the test types performed during typical system testing phase:

- A. UsabilityII Requirements based scenariosIII Testing parts of the code in isolationIV Correct order of parameters in API calls
- B. I, III
- C. II
- D. I
- E. IV
- F. II
- G. IV

Answer: B

Explanation:

System testing is a level of testing performed to evaluate the behavior and quality of a whole software product or system. System testing can include various types of testing, such as:

? I) Usability testing: A type of testing that evaluates how easy, efficient and satisfying it is to use the software product or system from the user's perspective.

? II) Requirements based scenarios testing: A type of testing that verifies that the software product or system meets its specified requirements or user stories by executing realistic scenarios or workflows. System testing does not include the following types of testing, as they are more suitable for lower levels of testing, such as unit testing or integration testing:

? III) Testing parts of the code in isolation: A type of testing that verifies the functionality and quality of individual software components or units by isolating them from other components or units.

? IV) Correct order of parameters in API calls: A type of testing that verifies the functionality and quality of software components or units that communicate with each other through application programming interfaces (APIs) by checking the correct order and format of parameters in API calls. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 20-21; Chapter 4, page 34-35.

NEW QUESTION 50

In maintenance testing, what is the relationship between impact analysis and regression testing?

- A. Impact analysis requires a regression testing for only the tests that have detected faults in previous SW release
- B. There is no relationship between impact analysis and regression testing.
- C. Impact analysis requires a regression testing for all program elements which were newly integrated (new functionalities).
- D. The impact analysis is used to evaluate the amount of regression testing to be performed.

Answer: D

Explanation:

In maintenance testing, the relationship between impact analysis and regression testing is that the impact analysis is used to evaluate the amount of regression testing to be performed. Maintenance testing is a type of testing that is performed on an existing software product after it has been delivered or deployed, in order to ensure that it still meets its requirements and functions correctly after a change or a modification. Maintenance testing can be triggered by various reasons, such as corrective maintenance (fixing defects), adaptive maintenance (adapting to new environments), perfective maintenance (improving performance), preventive maintenance (avoiding future problems), etc. Impact analysis is a technique that is used to assess the extent and nature of changes introduced by maintenance activities on the software product or project. Impact analysis helps to identify which parts of the software product are affected by the changes, which parts need to be modified or updated accordingly, which parts need to be retested or verified for correctness or compatibility, etc. Regression testing is a type of testing that verifies that previously tested software still performs correctly after a change or a modification. Regression testing helps to detect any side effects or unintended consequences of maintenance activities on the software product's functionality or quality. Regression testing can be performed at various levels and scopes depending on the impact analysis results. Therefore, in maintenance testing, impact analysis is used to evaluate the amount of regression testing to be performed. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 20.

NEW QUESTION 52

During system testing phase of a word processor, a tester finds that on opening a file from a particular set of files, which are part of a critical workflow, the word processor crashes. Which of the following is the next step the tester should take prior to recording the deviation?

- A. Try to recreate the incident before reporting
- B. Try to identify the code fragment causing the problem
- C. Send an email to the developer and not report the bug
- D. Report the incident as is without any further action

Answer: A

Explanation:

An incident is any event that occurs during testing that requires investigation. An incident report is a document that records the details of an incident. The next step the tester should take prior to recording the deviation is to try to recreate the incident before reporting. This can help confirm that the incident is reproducible and not caused by a random or external factor. This can also help gather more information about the incident, such as the steps to reproduce it, the expected and actual results, the severity and priority of the incident, or any screenshots or logs that can illustrate the incident. Trying to identify the code fragment causing the problem is not the next step the tester should take prior to recording the deviation, as this is a debugging activity that is usually performed by developers after receiving the incident report. Sending an email to the developer and not reporting the bug is not the next step the tester should take prior to recording the deviation, as this is an informal and unstructured way of communicating incidents that can lead to confusion, inconsistency or loss of information. Reporting the incident as is without any further action is not the next step the tester should take prior to recording the deviation, as this can result in incomplete or inaccurate incident reports that can hamper the investigation and resolution of incidents. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33.

NEW QUESTION 57

As the last stage of a test cycle of an embedded device, you are performing exploratory testing. You observed that some character. (A, X and Z) sent via a serial port to the device do not get registered on the device whereas they should be. You suspect that this could be due to a wrong configuration of the "bit parity" parameter.

Which of the following items of an incident report would you be UNABLE to write down based on this information?

- A. Expected result
- B. Test case identifier
- C. Test setup details
- D. Actual result

Answer: B

Explanation:

An incident report is a document that records the details of an incident. An incident report typically contains the following items:

? Identifier: A unique identifier for the incident report

? Summary: A concise summary of the incident

? Description: A detailed description of the incident, including the steps to reproduce it, the expected and actual results, and any relevant screenshots or logs

? Severity: The degree of impact that the incident has on the system

? Priority: The level of urgency for resolving the incident

? Status: The current state of the incident, such as new, open, resolved, closed, etc.

? Resolution: The action taken to resolve the incident, such as fix, workaround, reject, etc. Based on the information given in the question, the tester would be able to write down all of these items except for the test case identifier. A test case identifier is a unique identifier for a test case that is used to link it to other test artifacts, such as test plans, test scripts, test results or incident reports. However, since the tester is performing exploratory testing, there is no predefined test case that can be associated with the incident. Exploratory testing is an approach to testing that emphasizes learning, test design and test execution at the same time. Exploratory testing relies on the tester's skills, creativity and intuition to explore the software under test and discover defects. Exploratory testing does not use formal test cases or scripts, but rather uses test charters or missions that guide the tester's actions and objectives. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33; Chapter 5, page 47-48.

NEW QUESTION 61

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

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

Answer: A

Explanation:

Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential 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 Code Analysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs - CareerRide, ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

NEW QUESTION 65

Which of the following statements about reviews are TRUE?

- A. In walkthroughs the review meeting is typically led by the autho
- B. II Inspection is characterized by an open-ended review meetingIII Preparation before the review meeting is part of informal reviews IV Management rarely participates in technical review meetings
- C. II, III
- D. I, II
- E. I, IV
- F. III, IV

Answer: C

Explanation:

The following statements about reviews are true:

? I) In walkthroughs the review meeting is typically led by the author. A walkthrough is a type of review that has a predefined objective and agenda but no formal process or roles. A walkthrough is typically led by the author of the work product under review, who guides the participants through a scenario and solicits feedback.

? IV) Management rarely participates in technical review meetings. A technical review is a type of review that has a predefined objective and agenda but no formal process or roles. A technical review is typically performed by peers with technical expertise in order to evaluate technical aspects of a work product. Management rarely participates in technical review meetings, as they may not have sufficient technical knowledge or skills to contribute effectively. The following statements about reviews are false:

? II) Inspection is characterized by an open-ended review meeting. An inspection is a type of review that follows a defined process with formal entry and exit criteria and roles and responsibilities for participants. An inspection is characterized by a structured review meeting with a fixed duration and agenda.

? III) Preparation before the review meeting is part of informal reviews. Preparation before the review meeting is part of formal reviews, such as inspections or technical reviews. Preparation involves checking

NEW QUESTION 67

Which of the following activities are part of test planning?

- I) Setting the entry and exit criteria
- II) Determining the validity of bug reports
- III) Determining the number of resources required
- IV) Determining the expected result for test cases

- A. IV
- B. III
- C. I, III, IV
- D. I, II, IV

Answer: B

Explanation:

Test planning is a key activity in the testing process that involves defining the objectives, approach, resources, and schedule of intended test activities. Setting the entry and exit criteria (I) and determining the number of resources required (III) are integral parts of test planning. Determining the validity of bug reports (II) is more aligned with test analysis or test management activities post-execution, and determining the expected result for test cases (IV) is part of test design.

Therefore, options I and III (B) are the activities that belong to test planning.

NEW QUESTION 72

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 74

Which of the following BEST distinguishes the terms "validation" and "verification"?

- A. Verification is confirmation through the provision of objective evidence that the specified requirements have been met while validation is confirmation through the provision of objective evidence that the requirements for a specific intended use have been met
- B. Verification is confirmation through the provision of subjective evidence that the specified requirements have been met while validation is confirmation through the provision of subjective evidence that the designs for a specific intended use have been met
- C. Validation is confirmation through the provision of objective evidence that the specified requirements have been met while verification is confirmation through the provision of objective evidence that the requirements for a specific intended use have been met
- D. Validation is confirmation through the provision of subjective evidence that the specified requirements have been met while verification is confirmation through the provision of subjective evidence that the designs for a specific intended use have been met

Answer: A

Explanation:

In the context of software testing, the ISTQB Certified Tester Foundation Level (CTFL) v4.0 differentiates between "validation" and "verification" based on their respective focuses in the software development lifecycle. Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. In simpler terms, verification is about checking the product against the specified requirements to ensure it was built correctly. Validation, on the other hand, involves evaluating a system or component during or at the end of the development process to determine whether it meets specified requirements for its intended use. This means validation is about ensuring the product fulfills its intended use and meets the needs of the user.

References:

? ISTQB CTFL Syllabus v4.0: ISTQB Official Website

? ISTQB Foundation Level Resources v4.0: ASTQB Resources

NEW QUESTION 78

"Statement Testing" is part of;

- A. Specification Based testing
- B. Decision Testing
- C. Experience based testing
- D. Structured based testing

Answer: D

Explanation:

Statement Testing is a type of white-box testing technique where the test cases are designed based on the implementation of the software, specifically aiming to execute every statement in the code at least once. This falls under the category of structure-based testing (also known as white-box testing), where the internal structure of the system is used to design test cases. Therefore, option D is correct.

NEW QUESTION 82

Which of the following BEST defines risk level?

- A. Risk level is determined by calculating the absolute value of the sum of all potential issues that may occur on the project
- B. Risk level is calculated by adding the probabilities of all planned risks to a project
- C. Risk level is calculated by dividing the sum of all known risks by the sum of all unknown risks
- D. Risk level is determined by the likelihood of an event happening and the impact or harm from that event

Answer: D

Explanation:

Risk level is determined by the combination of two factors: the likelihood of an event occurring and the impact or harm that could result from that event. This approach allows risks to be prioritized based on their potential effect on the project or system. The likelihood represents the probability of the risk event occurring, while the impact represents the severity of the consequences if the event does happen. This concept is fundamental in risk-based testing and helps guide decision-making during the testing process. References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.1.

NEW QUESTION 83

Which of the following is the main benefit of a configuration management of testware?

- A. All testware is backed up with restore option, including incident reports and change request
- B. The testware can be traced to information in requirements tools and to the bug tracking system.
- C. All testware items are identified, version controlled, tracked for changes with relation to each other
- D. There is an easy way to assess the level to test coverage provided by the existing tests

Answer: C

Explanation:

Configuration management of testware is a critical aspect of maintaining the integrity and traceability of test assets throughout the testing lifecycle. The main benefit of configuration management is to ensure that all testware items, such as test cases, test scripts, test data, and test results, are systematically identified, version controlled, and tracked for changes in relation to each other.

Option C accurately describes this benefit. By applying configuration management principles to testware, teams can manage changes to test assets efficiently, ensuring that the testware remains consistent, up-to-date, and aligned with the version of the software under test. This control mechanism facilitates the reproducibility of tests, enhances the reliability of testing activities, and supports traceability from requirements through to defects.

Options A, B, and D describe other aspects of test management and testing processes but do not capture the core benefit of configuration management of testware, which is centered on the systematic control and tracking of testware items.

NEW QUESTION 86

Which of the following tasks is MOST LIKELY to be performed by the tester?

- A. Develop a test strategy and test policy for the organization
- B. Promote and advocate the test team within the organization
- C. Create the detailed test execution schedule
- D. Introduce suitable metrics for measuring test progress

Answer: C

Explanation:

Testers are typically involved in creating detailed test execution schedules, among other tasks such as designing tests, executing tests, and logging defects. Creating a test strategy and test policy, promoting and advocating the test team, and introducing metrics are typically responsibilities of test managers or senior roles.

In the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, the responsibilities of testers include creating test cases, setting up test (ISTQB not-for-profit association)nts, executing tests, and reporting defects, which align with creating detailed test execution schedules6†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 90

4 equivalence classes are given for integer values:

$0 < x < 100$

$100 \leq x \leq 200$

$200 < x < 500$

$x \geq 500$

Which of the following options represent correct set of data for valid equivalence class partitions?

- A. 50; 100; 200. 1000
- B. 0. 1.99, 100.200,201.499, 500;
- C. 0.50; 100; 150.200.350.500;
- D. 50; 100; 250; 1000

Answer: C

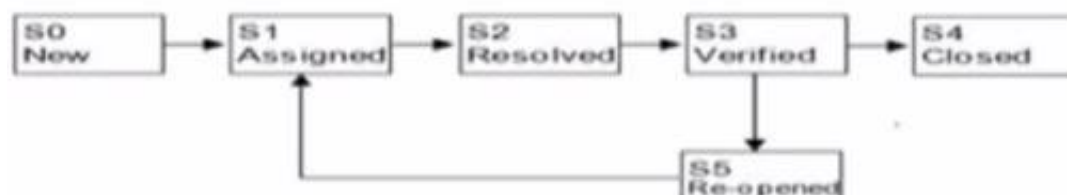
Explanation:

The correct set of data for valid equivalence class partitions should include one value from each equivalence class, and no value from outside the range. Option C satisfies this condition, as it has one value from each of the four equivalence classes (50, 100, 250, 500). Option A has two values from the same equivalence class (100 and 200), option B has values outside the range (0 and 0.99), and option D has two values from the same equivalence class (1000 and 500). Verified

References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 35.

NEW QUESTION 92

Which sequence of stated in the answer choices is correct in accordance with the following figure depicting the life-cycle of a defect?



- A. S0->S1->S2->S3->S5->S1
- B. S0->S1->S2->S3->S5->S1->S2->S3
- C. S0->S1->S2->S3->S4
- D. S0->S1 ->S2->S3->S5->S3->S4

Answer: D

Explanation:

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0, the life cycle of a defect typically follows a sequence from its discovery to its closure. In the provided figure, it starts with S0 (New), moves to S1 (Assigned), then to S2 (Resolved), followed by S3 (Verified). If the defect is not fixed, it can be Re-opened

(S5) and goes back for verification (S3). Once verified, it is Closed (S4). References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.4.3, Page 17.

NEW QUESTION 94

Which of the following is an example of black-box dynamic testing?

- A. Functional Testing
- B. Code inspection
- C. Checking memory leaks for a program by executing it
- D. Coverage analysis

Answer: A

Explanation:

Functional testing is an example of black-box dynamic testing. Black-box testing (also known as specification-based testing) is a type of testing that does not consider the internal structure or implementation of the system under test, but rather its external behavior or functionality. Dynamic testing is a type of testing that involves executing the system under test with various inputs and observing its outputs. Functional testing is a type of black-box dynamic testing that verifies that the system under test performs its intended functions according to its requirements or specifications. Functional testing can be performed at various levels and scopes depending on the objectives and criteria of testing. The other options are not examples of black-box dynamic testing. Code inspection is an example of white-box static testing. White-box testing (also known as structure-based testing) is a type of testing that considers the internal structure or implementation of the system under test. Static testing is a type of testing that does not involve executing the system under test, but rather analyzing it for defects, errors, or violations of standards. Code inspection is a type of white-box static testing that involves examining the source code of the system under test for quality, readability, maintainability, etc. Checking memory leaks for a program by executing it is an example of white-box dynamic testing. Memory leaks are defects that occur when a program fails to release memory that it has allocated but no longer needs. Checking memory leaks for a program by executing it requires knowledge and access to the internal structure or implementation of the program, such as memory allocation and deallocation mechanisms, pointers, references, etc. Coverage analysis is an example of white-box static testing. Coverage analysis is a technique that measures how much of the code or structure of the system under test has been exercised by a test suite. Coverage analysis requires knowledge and access to the internal structure or implementation of the system under test, such as statements, branches, paths, conditions, etc. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 7.

NEW QUESTION 95

Which of the following BEST explains a drawback of independent testing?

- A. Having the business organization participate as an independent test team can hurt the overall testing effort since business participants are often not trained nor experienced in testing
- B. Due to their differing backgrounds and perspectives, an independent test team may discover defects which the developers did not uncover
- C. An independent test team may be isolated from the rest of the development and project team
- D. An independent test team may possess specializations in specific test types such as usability or security which detract from the overall effectiveness of the test team

Answer: C

Explanation:

Independent testing offers several advantages, such as unbiased testing and detection of different defects. However, a drawback is that an independent test team may be isolated from the development team and project team. This can lead to communication gaps, reduced collaboration, and a lack of understanding of the project context.

According to the ISTQB Certified Tester Foundation Level (CTFL) syllabus v4.0, an independent test team may not have the same level of understanding of the system as the development team, leading to potential issues in communication and integration (ISTQB not-for-profit association).

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 99

The following open incident report provided: Date: 01.01.01

Description: When pressing the stop button the application status remain in "Attention" instead of "Ready".

Severity: High

Life Cycle: Integration

Which of the following details are missing in the given incident report?

- A. Identification or configuration of the applicationI
- B. The name of the developerII
- C. Recommendation of the developerIV The actions and/or conditions that came before the pressing of the button
- D. IV
- E. IV
- F. II
- G. II, III

Answer: B

Explanation:

In an incident report, essential details provide context and facilitate the investigation and resolution of the incident. The missing elements in the given incident report are:

I. Identification or configuration of the application: This detail is crucial as it specifies which version or configuration of the application is affected, helping in reproducing the issue. IV. The actions and/or conditions that came before pressing the button: Understanding the sequence of actions leading to the issue is vital for replicating and diagnosing the problem. The name of the developer (II) and the recommendation of the developer (III) are not typically included in an incident report as they do not contribute to identifying or resolving the incident. The focus is on the incident's details, reproduction steps, and the system's state rather than on personnel or proposed solutions at this stage. Therefore, option B, which includes both I and IV, is the correct answer.

NEW QUESTION 104

The following 4 equivalence classes are given:

$x \leq -100$
 $-100 < x < 100$
 $100 \leq x < 1000$
 $x \geq 1000$

Which of the following alternatives includes correct test values for x. based on equivalence partitioning?

- A. -100; 100:1000; 1001
- B. -500; 0; 100; 1000
- C. -99; 99:101; 1001
- D. -1000; -100; 100; 1000

Answer: D

Explanation:

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

Option D provides a value from each of these partitions:

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

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

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

? For ($x \geq 1000$), it gives 1500.

So, option D covers all four given equivalence classes with appropriate values. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents available at ISTQB and ASTQB.

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

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

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

NEW QUESTION 107

Which statement about use case testing is true?

- A. The test cases are designed to find defects in the data flow.
- B. The test cases are designed to be used by real users, not by professional testers
- C. The test cases are always designed by customers or end users.
- D. The test cases are designed to find defects in the process flow.

Answer: D

Explanation:

Use case testing is a technique that helps identify test cases that exercise the whole system on a transaction by transaction basis from start to finish. Use cases are descriptions of how users interact with the system to achieve a specific goal. Use case testing is not focused on data flow, but rather on process flow. Use case testing can be performed by professional testers, customers or end users, depending on the context. Use case testing does not require the test cases to be designed by customers or end users, but rather by anyone who has access to the use case specifications. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 4, page 36.

NEW QUESTION 110

Which of the types of test tools noted below BEST describes tools that support reviews?

- A. Tools to assess data quality
- B. Tools to support usability testing
- C. Tools to support specialized testing needs
- D. Tools to support static testing

Answer: D

Explanation:

Static testing refers to testing that doesn't involve executing code. It includes activities like reviews, inspections, and static analysis. Tools that support static testing help with activities such as analyzing source code, checking coding standards, and aiding in document reviews. These tools can automate or facilitate various aspects of static testing processes, such as highlighting potential issues in code or documents without executing the software.

References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 6.1.1.

NEW QUESTION 115

Which of the following statements about static analysis are FALSE?

- A. Static analysis can be used Instead of dynamic testing.I
- B. Static analysis can uncover defects like security vulnerabilities.II
- C. Static analysis can be used to check conformance to specifications and standard
- D. Static analysis typically detects failures prior to component testing.
- E. II
- F. I
- G. III
- H. II
- I. IV
- J. I, IV

Answer: D

Explanation:

Static analysis involves analyzing the software's code, design, and structure without executing the program. It can uncover various types of defects, including security vulnerabilities (II) and non-conformance to specifications and standards (III). However, static analysis cannot replace dynamic testing (I), which involves executing the software to observe its behavior under various conditions. Dynamic testing can identify failures that static analysis cannot, such as those related to runtime issues and interaction between different parts of the software. Statement IV is false because static analysis does not detect failures; it detects defects. Failures are observed when the software is executed, which is beyond the scope of static analysis.

NEW QUESTION 119

A test manager decided to skip static testing since he believes bugs can be found easily by doing dynamic testing. Was this decision right or wrong?

- A. The decision was wrong
- B. Ensuring quality mandates that static testing is performed after performing the dynamic testing.
- C. The decision was right
- D. Static testing is usually redundant if a product is planned to go through a full-cycle of dynamic testing.
- E. The decision was right
- F. Most of the bugs are easier to identify during the dynamic testing.
- G. The decision was wrong
- H. Static testing can find defects early in the development process, reducing the overall cost of testing and development

Answer: D

Explanation:

Static testing is a form of testing that does not involve executing the software or system under test. It includes activities such as reviews, inspections, walkthroughs, and analysis of documents, code, and models. Static testing can find defects early in the development process, before they become more expensive and difficult to fix in later stages. Static testing can also improve the quality of the software or system by preventing defects from being introduced in the first place. Static testing can complement dynamic testing, which involves executing the software or system under test and checking the results against expected outcomes. Dynamic testing can find defects that static testing may miss, such as performance, usability, or integration issues. However, dynamic testing alone is not sufficient to ensure quality, as it may not cover all possible scenarios, inputs, or paths. Therefore, a test manager who decides to skip static testing is making a wrong decision, as he or she is ignoring the benefits of static testing and relying solely on dynamic testing, which may not be effective or efficient enough to find and prevent defects. References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.1.1, page 14; ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 36; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 3, page 9.

NEW QUESTION 123

A system computes prices for bus tickets. The price depends on

- the passenger type (baby, child, adult, senior citizen, student, military)
- the travelling type (as single or in a group)
- the distance (zone 1. 2. 3)
- the kind of transport (ordinary, express)

Which of the following test techniques is the most appropriate one for testing the price computation?

- A. Statement coverage
- B. State transition testing
- C. Equivalence partitioning
- D. Use case testing

Answer: C

Explanation:

Equivalence partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component. Equivalence partitioning can be used to reduce the number of test cases by selecting one representative value from each partition. Equivalence partitioning is suitable for testing the price computation, as it can identify different partitions based on the passenger type, the travelling type, the distance and the kind of transport. Equivalence partitioning is not statement coverage, which is a technique that measures how many executable statements in a source code are executed by a test suite. Statement coverage is not appropriate for testing the price computation, as it does not consider the input data or output results. Equivalence partitioning is not state transition testing, which is a technique that models how a system transitions from one state to another depending on events or conditions. State transition testing is not relevant for testing the price computation, as it does not involve any states or transitions. Equivalence partitioning is not use case testing, which is a technique that tests how users interact with a system to achieve a specific goal. Use case testing is not applicable for testing the price computation, as it does not focus on a single function or component. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION 127

The following sentences refer to the 'Standard for Software Test Documentation' specification (IEEE 829).

Which sentence is correct?

- A. Any deviation from this standard should be approved by management, marketing & development
- B. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization
- C. The key to high quality test documentation regimes is strict adherence to this standard
- D. This test plan outline is relevant for military project
- E. For consumer market projects there is a different specification with fewer items.

Answer: B

Explanation:

The IEEE 829 standard is a widely used specification for test documentation, but it is not mandatory or universal. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization. The standard does not require any approval from management, marketing or development for any deviation, nor does it depend on the type of project (military or consumer market). The standard also does not guarantee high quality test documentation regimes, as it only provides a general outline and format, not the actual content or quality criteria. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 16.

NEW QUESTION 129

A mid-size software product development company has analyzed data related to defects detected in its product and found out that defects fixed in earlier builds are getting re-opened after a few months.

The company management now seeks your advice in order to reverse this trend and prevent re-opening of defects fixed earlier.

What would be your FIRST recommendation to the company?

- A. Automate existing test suits so that lesser time is spent on execution of each test, and more tests can be executed in the available time thus leading to a lower probability of defects slipping by
- B. Verify existing regression test suite are adequate, and augment it, if required, in order to ensure that defects fixed earlier get re-tested in each subsequent build
- C. Analyze the product modules containing maximum defects, and get them thoroughly tested and defects fixed as a one-time activity
- D. If required, train the teams responsible for development and testing of the modules containing maximum number of defects, and if this does not help, replace them with more knowledgeable people

Answer: B

Explanation:

Regression testing is a type of testing that verifies that previously tested software still performs correctly after changes. Regression testing can help prevent re-opening of defects fixed earlier by ensuring that they do not cause any new failures or side effects. The first recommendation to the company is to verify existing regression test suite are adequate, and augment it, if required, in order to ensure that defects fixed earlier get re-tested in each subsequent build. This can help improve the coverage and effectiveness of regression testing and detect any regression defects as soon as possible. Automating existing test suites may also help reduce the time and effort required for regression testing, but this is not the first recommendation, as automation may not be feasible or cost-effective for all test cases. Analyzing the product modules containing maximum defects and getting them thoroughly tested and defects fixed as a one-time activity may also help reduce the defect density and improve the quality of those modules, but this is not the first recommendation, as it does not address the root cause of re-opening defects fixed earlier. Training or replacing the teams responsible for development and testing of the modules containing maximum number of defects may also help improve their skills or performance, but this is not the first recommendation, as it may not be necessary or appropriate for all teams. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 19; Chapter 4, page 45.

NEW QUESTION 131

A system has valid input numbers ranging between 1000 and 99999 (both inclusive). Which of the following inputs are a result of designing tests for all valid equivalence classes and their boundaries?

- A. 999.1000.23232.99999.100000
- B. 999.1000.50000.100000.100001
- C. 999.100000
- D. 1000,50000,99999

Answer: B

Explanation:

A correct list of boundary values for the P input should include the minimum and maximum values of the valid range (15 and 350), as well as the values just below and above the boundaries (14 and 351). Boundary value analysis is a test design technique that involves testing the values at or near the boundaries of an input domain or output range, as these values are more likely to cause errors than values in the middle. Option B satisfies this condition, as it has all four boundary values (14, 15, 350, 351). Option A has two values from the same equivalence class (1000 and 99999), option C has two values outside the range (999 and 100000), and option D has no boundary values at all. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 34.

NEW QUESTION 134

Given the following priorities and dependencies for these test cases: SEE ATTACHMENT

Which of the following test execution schedules BEST takes into account the priorities and technical and logical dependencies?

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

Answer: C

Explanation:

When scheduling test cases, priorities and dependencies must be considered. The best execution order will respect both the logical dependencies and the priorities assigned to each test case.

Given the options, the correct order considering the priorities and dependencies is:

- ? TC1 (Priority 1)
- ? **TC (ISTQB not-for-profit association)ity 2, dependent on TC1)
- ? TC2 (Priority 3, dependent on TC1)
- ? TC4 (Priority 4)
- ? TC5 (Priority 5)
- ? TC6 (Priority 6, dependent on TC4)

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, understanding dependencies and scheduling tests accordingly is crucial for effective test execution.

References:

- ? Certified Tester Foundation Level v4.0
- ? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 136

What is 'Component Testing'?

- A. Integration Testing
- B. Functional testing
- C. Experience-based testing
- D. A test level

Answer: D

Explanation:

Component testing is a test level. A test level is a group of test activities that are organized and managed together based on some common characteristics or objectives. A test level can be defined based on various factors, such as the scope and target of testing, the phase and model of development, the stakeholders and roles involved in testing, etc. Component testing (also known as unit testing or module testing) is a test level that focuses on verifying the functionality and quality of individual software components (such as modules, classes, functions, methods, etc.). Component testing can be performed by developers or testers using various techniques and tools depending on the type and complexity of the components. The other options are not test levels. Integration testing is another test level that focuses on verifying the functionality and quality of groups of software components that interact with each other or with external systems. Functional testing is a type of black-box dynamic testing that verifies that the system under test performs its intended functions according to its requirements or specifications. Experience-based testing is a category of test design techniques that rely on the tester's knowledge and intuition to derive and select test cases based on their experience with similar systems, technologies, domains, risks, etc. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 19.

NEW QUESTION 141

Decision table testing is being performed on transactions in a bank's ATM (Automated Teller Machine) system. Two test cases have already been generated for rules 1 and 4. which are shown below:

SEE ATTACHMENT 1

Given the following additional test cases: SEE ATTACHMENT 2

Which two of the additional test cases would achieve full coverage of the full decision table (when combined with the test cases that have already been generated for rules 1 and 4)?

- A. DT1, DT4
- B. DT3, DT4
- C. DT2, DT3
- D. DT1.DT2

Answer: C

Explanation:

Decision table testing is used to analyze combinations of inputs to determine the appropriate outputs, often based on specific rules or conditions.

For the problem statement:

? Rule 1: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = True, Correct PIN = True)

? Rule 4: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = True, Correct

PIN = False)

The additional test cases are:

? DT1: (Withdrawal = Allowed, Balance = Insufficient, Fast Cash = True, Correct PIN = True)

? DT2: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = False, Correct PIN = True)

? DT3: (Withdrawal = Allowed, Balance = Insufficient, Fast Cash = True, Correct PIN = False)

? DT4: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = False, Correct PIN = False)

From the given test cases, DT2 covers the scenario where Fast Cash is False, which is not covered in the initial cases. DT3 covers the case where Balance is Insufficient and PIN is incorrect.

Combining Rules 1 and 4 with DT2 and DT3 covers all the scenarios. References:

? Certified Tester Foundation Level v4.0

? 10 Sample Exams ISTQB Foundation Level (CTFL) v4.0

NEW QUESTION 143

Which of the following is MOST likely to be an example of a PROJECT risk?

- A. A computation is not always performed correctly in some situations
- B. A system architecture may not support some non-functional requirements
- C. Team members' skills may not be sufficient for the assigned work
- D. Specific modules do not adequately meet their intended functions according to the user

Answer: C

Explanation:

A project risk relates to potential issues that could affect the overall success of the project. Among the options provided, the risk that "Team members' skills may not be sufficient for the assigned work" is clearly a project risk because it pertains to the potential failure of the project due to inadequate skillsets among the team.

This risk affects the entire project's ability to meet its objectives. References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.2.

NEW QUESTION 146

Which of the following is a valid collection of equivalence classes for the following problem: An integer field shall contain values from and including 1 to and including 15.

- A. Less than 0.1 through 14. 15 and more
- B. Less than 1.1 through 14. more than 15
- C. negative number
- D. 1 through 15. above 15
- E. Less than 1.1 through 15. more than 15

Answer: D

Explanation:

Equivalence partitioning is a black-box test design technique where inputs to the software or system are divided into groups that are expected to exhibit similar

behavior. For an integer field that should accept values from 1 to 15, the valid equivalence class is 1 through 15. The invalid equivalence classes are numbers less than 1 and numbers more than 15. Therefore, option D, "Less than 1, 1 through 15, more than 15," correctly identifies the valid equivalence class along with the two invalid classes, covering all possible input scenarios for the field. Options A, B, and C either do not accurately capture the valid range or incorrectly specify the range boundaries.

NEW QUESTION 150

Which statement best describes the key difference between a mindset for test activities and a mindset for development activities?

- A. A tester possesses professional pessimism while a developer is concerned with validating the product
- B. A tester is concerned with finding defects while a developer is interested in designing solutions
- C. A tester is interested in building solutions while a developer is concerned with verifying the product
- D. A tester is concerned with verifying the product while a developer possesses professional pessimism

Answer: B

Explanation:

The key difference between the mindsets for test activities and development activities lies in the objectives: a tester is primarily concerned with finding defects to ensure product quality, while a developer focuses on designing and building solutions. This distinct focus helps ensure thorough quality checks and balances within the software development life cycle (ISTQB not-for-profit association).References:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf

NEW QUESTION 151

Which of the following provides the BEST description of statement coverage?

- A. A white-box test technique which covers the decision results which determine the next statement to be executed
- B. A black-box test technique which uses a state table to derive test cases
- C. A white-box test technique which focuses on the percentage of executable statements that has been executed by a test suite
- D. An experience-based test technique in which test cases are based on the tester's knowledge of past failures

Answer: C

Explanation:

Statement coverage is a white-box test technique which focuses on executing all possible statements in the code at least once during testing. This helps in determining the percentage of executable statements that have been executed by the test suite, aiming to ensure that all parts of the program have been tested at least once (ISTQB Main Web).References:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION 156

Which of the following activities does NOT belong to a typical technical review?

- A. Pre-meeting preparation by reviewers
- B. Using checklists during the meeting
- C. Inviting end-users to the meeting
- D. Preparation of a review report

Answer: C

Explanation:

Technical reviews are structured meetings that aim to examine various aspects of a product or project to identify any defects or improvements. Options A (Pre-meeting preparation by reviewers), B (Using checklists during the meeting), and D (Preparation of a review report) are typical activities in a technical review process. Inviting end-users to the meeting (C), however, is generally not part of a typical technical review, as these reviews are usually more focused on the technical aspects and are conducted by peers or experts within the development or testing teams rather than end-users.

NEW QUESTION 159

Which of the following CORRECTLY matches the roles and responsibilities in a formal review?

- A. Facilitator - Fixes defects in the work product under review
- B. Scribe - Collates potential defects found during the individual review activity
- C. Review Leader - Creates the work product under review
- D. Author - Identify potential defects in the work product under review

Answer: B

Explanation:

In formal reviews, the scribe's role is to collate potential defects and other findings during the review process. This position is crucial as it ensures all observations and defects are recorded accurately, facilitating efficient analysis and resolution of issues identified during the review.References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2.4 "Roles and Responsibilities in a Formal Review".

NEW QUESTION 162

Which of the following is a possible reason for introducing a defect in software code?

- A. Rushing to meet a tight deadline to turn code over for testing
- B. Improper unit testing
- C. Improper system testing
- D. Focus on static testing over dynamic testing

Answer:

A

Explanation:

The ISTQB CTFL syllabus identifies several causes for defects in software. One prominent reason, as highlighted in the curriculum, is the pressure and rush to meet tight deadlines, which can lead to insufficiently reviewed or tested code being moved into further stages of testing or production. This scenario describes rushing to meet a deadline as a potential cause for defects because it may compromise the thoroughness of code development and testing. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.1 "Why is Testing Necessary?".

NEW QUESTION 165

Given the following requirement: Requirement ID: 2 8

Requirement Description Additional Entrance Fee Detailed Description

An additional fee of \$3 is charged during the weekend, but

- 1) Visitors aged under 7 are not charged.
- 2) Visitors aged 7 to 13 inclusive get a 20% discount off the additional fee.
- 3) Visitors aged greater than 65 get a 50% discount off the additional fee. Age should be an integer of 0 or above.

Weekend means Friday to Sunday inclusive. Which of the following statements is NOT correct?

- A. Thursday is a valid input boundary value.
- B. A minimum of 6 valid test cases are derived from boundary value analysis based on input age.
- C. \$3.01 is a valid output boundary value.
- D. 7 and 13 are boundary values for the equivalence partition including age 10.

Answer: A

Explanation:

Boundary value analysis is a technique that tests boundary values between partitions of equivalent data. Boundary values are values at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Boundary value analysis can be applied to both input and output values. Based on the given requirement, we can identify two input values: age and weekend. Age should be an integer of 0 or above, and weekend means Friday to Sunday inclusive. The following statement is not correct:

? A) Thursday is a valid input boundary value. This statement is not correct, as Thursday is not a boundary value for the input weekend. The boundary values for the input weekend are Friday and Sunday, as they are at the edge of the equivalence partition that represents weekend days. The following statements are correct:

? B) A minimum of 6 valid test cases are derived from boundary value analysis based on input age. This statement is correct, as we can derive six valid test cases based on input age by using the minimum and maximum values for each equivalence partition defined by the requirement. The equivalence partitions for input age are: under 7 (0 to 6), 7 to 13 inclusive (7 to 13), and greater than 65 (66 and above). The minimum and maximum values for each partition are: 0 and 6, 7 and 13, and 66 and any value above it.

? C) \$3.01 is a valid output boundary value. This statement is correct, as \$3.01 is a boundary value for the output additional fee. The additional fee can have four possible values depending on the input age: \$0 (for visitors aged under 7), \$2.40 (for visitors aged 7 to 13 inclusive with a 20% discount), \$1.50 (for visitors aged greater than 65 with a 50% discount), and \$3 (for visitors aged between 14 and 65). The boundary values for the output additional fee are \$0 and \$3, as they are at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Therefore, \$3.01 is a valid output boundary value, as it is at the smallest incremental distance above \$3.

? D) 7 and 13 are boundary values for the equivalence partition including age 10.

This statement is correct, as 7 and 13 are boundary values for the equivalence partition that represents visitors aged 7 to 13 inclusive. This partition includes age 10, which is an internal value within the partition. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION 168

For the following pseudo-code determine number of tests required for 100% statement coverage

IF Gender = Boy

If Age > 3 AND Age < 5 Shoe Size = 1

ELSE IF Age >= 5 AND Age < 7

Shoe Size = 2 ENDIF

ELSE

IF Age > 3 AND Age < 5

Shoe Size = 0

ELSE IF Age >= 5 AND Age < 7

Shoe Size = 1 ENDIF ENDIF

- A. 6
- B. 4
- C. 2
- D. 6

Answer: B

Explanation:

To achieve 100% statement coverage, we need to design test cases that ensure every statement in the given pseudo-code is executed at least once. Analyzing the pseudo-code, we notice that there are conditions based on two variables: Gender and Age. To cover all statements, we need to consider the paths that lead to each assignment of the Shoe Size variable.

? Gender = Boy, Age <= 3 (Shoe Size assignment is not reached, but the condition is evaluated)

? Gender = Boy, Age > 3 AND Age < 5 (Shoe Size = 1)

? Gender = Boy, Age >= 5 AND Age < 7 (Shoe Size = 2)

? Gender != Boy, Age <= 3 (Again, Shoe Size assignment is not reached, but the condition is evaluated)

? Gender != Boy, Age > 3 AND Age < 5 (Shoe Size = 0)

? Gender != Boy, Age >= 5 AND Age < 7 (Shoe Size = 1)

However, upon closer inspection, we see that tests 1 and 4 do not contribute to statement coverage as they do not lead to a Shoe Size assignment. Therefore, we only need 4 test cases to achieve 100% statement coverage, making option B the correct answer.

NEW QUESTION 172

Which of the following statements about decision tables are TRUE?

- A. Generally, decision tables are generated for low risk test items.I
- B. Test cases derived from decision tables can be used for component tests.II
- C. Several test cases can be selected for each column of the decision table.I
- D. The conditions in the decision table represent negative tests generally.
- E. III
- F. I, IV
- G. I
- H. IV
- I. I
- J. III

Answer: D

Explanation:

A decision table is a technique that shows combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects). A decision table consists of four quadrants: conditions (inputs), actions (outputs), condition entries (values) and action entries (results). The following statements about decision tables are true:

? II. Test cases derived from decision tables can be used for component tests.

Decision tables can be used to test components that have multiple inputs and outputs that depend on logical combinations of conditions. Decision tables can help cover all possible combinations or scenarios in a systematic way.

? III. Several test cases can be selected for each column of the decision table. A column of a decision table represents a unique combination of condition entries and action entries. Several test cases can be selected for each column by varying other input values or expected results that are not part of the decision table. The following statements about decision tables are false:

? I. Generally, decision tables are generated for low risk test items. Decision tables are not related to risk level, but rather to complexity level. Decision tables are generated for test items that have complex logic or multiple conditions and actions that need to be tested.

? IV. The conditions in the decision table represent negative tests generally. The conditions in the decision table represent both positive and negative tests, depending on whether they are valid or invalid inputs for the test item. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 4, page 42-43.

NEW QUESTION 174

Which of the following is NOT a product risk?

- A. Poor software usability
- B. Failure-prone software is delivered
- C. Problems in defining the right requirements
- D. Software does not perform the intended functions

Answer: C

Explanation:

Problems in defining the right requirements is not a product risk, but rather a project risk. A product risk is a risk that affects the quality or performance of the software product itself, such as poor usability, failure-prone functionality, security vulnerabilities, compatibility issues, etc. A project risk is a risk that affects the management or delivery of the software project itself, such as unrealistic schedule, insufficient resources, unclear scope, changing requirements, etc. The other options are examples of product risks, as they relate to the software product's characteristics or features. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 12.

NEW QUESTION 175

The following part of a business process flow is specified; REPEAT (book a bill) UNTIL (User presses Cancel). How many test cases are necessary in order to achieve 100% branch coverage of the process flow?

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

Answer: C

Explanation:

To achieve 100% branch coverage of the process flow, we need to test both the true and false outcomes of the condition (User presses Cancel). Branch coverage is a type of structural testing that measures how many decision outcomes in a program have been executed by a test suite. Branch coverage can be used to assess the adequacy or completeness of a test suite.

To test the true outcome of the condition, we need a test case that simulates the user pressing Cancel after booking a bill. This test case will exit the loop and end the process flow.

To test the false outcome of the condition, we need a test case that simulates the user not pressing Cancel after booking a bill. This test case will repeat the loop and book another bill.

Therefore, we need at least two test cases to achieve 100% branch coverage of the process flow. One test case for each possible outcome of the condition. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION 177

Which of the following statements about independent testing is WRONG?

- A. Independent testing is necessary because developers don't know any testing.
- B. Independent testing is best suited for the system test level.
- C. A certain degree of independence makes the tester more effective at finding defects.
- D. Independent test teams may find other types of defects than developers who are familiar with the system's structure.

Answer: A

Explanation:

Independent testing is testing performed by a person or group that is independent of the development team. Independent testing can have various degrees of

independence, ranging from testers who are part of the same organization as developers to testers who are external contractors or consultants. Independent testing can have various benefits, such as reducing bias, increasing objectivity, improving quality, or providing different perspectives. Independent testing is not necessary because developers don't know any testing, as this is a wrong and disrespectful statement. Developers can perform various types of testing, such as unit testing, component testing, or integration testing. However, independent testing can complement developer testing by providing additional levels of verification and validation, such as system testing, acceptance testing, or non-functional testing. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 16-17.

NEW QUESTION 181

Which ONE of the following statements about state transition testing is correct?

- A. The state transition diagram explicitly shows all invalid transitions.
- B. The size of the state table depends on the number of possible transitions between the states
- C. Usually it is not possible to create tests to cover all transitions and all states
- D. All transitions between states are explicitly shown in the state table.

Answer: D

Explanation:

State transition testing is a black-box testing technique used to analyze the behavior of a system by examining the transitions between different states in response to events. In state transition testing, a state table or diagram is used to represent the states of a system and the transitions between these states triggered by events.

Option D is correct because in state transition testing, all transitions between states should be explicitly shown in the state table. This includes valid transitions that the system is expected to make under normal operation and, where relevant, invalid transitions that should be tested to ensure the system handles unexpected or erroneous inputs gracefully. The state table provides a comprehensive view of how the system should behave, making it possible to create tests that cover all defined transitions.

NEW QUESTION 184

Which of the following is NOT a common objective of testing?

- A. Finding defects in the software
- B. Preventing defects
- C. Debugging the software to find the reason for defects
- D. Providing information on the status of the system

Answer: C

Explanation:

Debugging the software to find the reason for defects is not a common objective of testing, but rather a task of development or maintenance. Debugging is a process of locating and fixing errors in the software code, while testing is a process of finding and reporting defects in the software behavior or quality. Testing does not aim to fix defects, but rather to provide information on their existence and impact. The other options are common objectives of testing. Finding defects in the software is one of the main objectives of testing, as it helps to improve the quality and reliability of the software. Preventing defects is another objective of testing, as it helps to avoid rework and reduce costs and risks. Providing information on the status of the system is another objective of testing, as it helps to support decision making and risk management. Verified

References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 3.

NEW QUESTION 189

Which of the following is true about Oracles?

- A. Sometimes old version of a product can be used as an Oracle
- B. Oracles help in reproducing the irreproducible bugs
- C. Oracles are derived from the design
- D. Oracles can be generated automatically using data generators

Answer: A

Explanation:

An oracle is a mechanism or source that can provide the expected result for a given test input or situation. Sometimes old version of a product can be used as an oracle, if it is assumed that the old version behaves correctly for the test cases that are executed on the new version. This is also known as back-to-back testing. Oracles do not help in reproducing the irreproducible bugs, as they only provide the expected results, not the actual results. Oracles are not derived from the design, but from the requirements or specifications. Oracles cannot be generated automatically using data generators, as data generators only provide test inputs, not test outputs. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 9.

NEW QUESTION 190

For a mandatory input field "ZIP code" the following rules are given:

- 1 - The valid ZIP code format is 5 numeric digits.
- 2 - The code has to exist in the post office's official ZIP code list

Using equivalence classes partitioning, how many test cases are required to test this field?

- A. 8
- B. 3
- C. 6
- D. 4

Answer: D

Explanation:

Equivalence classes partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component. Equivalence classes partitioning can be used to reduce the number of test cases by selecting one representative value from each partition. For the ZIP code field, there are four equivalence classes based on the given rules:

? Valid ZIP code format and valid ZIP code value (e.g., 12345)
? Valid ZIP code format and invalid ZIP code value (e.g., 99999)
? Invalid ZIP code format and valid ZIP code value (e.g., 1234)
? Invalid ZIP code format and invalid ZIP code value (e.g., ABCDE) Therefore, four test cases are required to test this field, one for each equivalence class.
Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION 195

Which of the following is correct with regards to debugging?

- A. Debugging identifies the cause of a failure
- B. Debugging is often performed by test engineers
- C. Debugging is considered part of the testing activities
- D. Debugging is intended to find as many defects as possible in the code

Answer: A

Explanation:

Debugging is the process of finding, analyzing and removing the causes of failures in software. Debugging is not considered part of testing, but rather a development activity that can involve testing. Debugging is not intended to find as many defects as possible, but rather to fix the specific failure that was observed. Debugging is usually performed by developers, not by test engineers. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 1, page 6.

NEW QUESTION 196

A software calculates the annual car tax using three inputs:

- E; the emission level of the vehicle
- P: the power of the vehicle
- T the type of the vehicle

The input value for P can be integer positive values between 15 and 350.

Which of the following answers contains a correct list of a boundary values for the P input?

- A. 14,351
- B. 14,15,350,351
- C. 15,350
- D. 5.175.500

Answer: B

Explanation:

A correct list of boundary values for the P input should include the minimum and maximum values of the valid range (15 and 350), as well as the values just below and above the boundaries (14 and 351). Boundary value analysis is a test design technique that involves testing the values at or near the boundaries of an input domain or output range, as these values are more likely to cause errors than values in the middle. Option B satisfies this condition, as it has all four boundary values (14, 15, 350, 351). Option A has only two boundary values (14 and 351), option C has only two boundary values (15 and 350), and option D has no boundary values at all. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 34.

NEW QUESTION 199

When should component integration tests be carried out?

- A. Integration tests should always be done after system tests
- B. Integration tests should be done at the customer's site, after acceptance tests
- C. Integration tests can be done before or after system tests
- D. Integration tests should always be done before system tests

Answer: D

Explanation:

Component integration tests are designed to verify the interactions and interfaces between integrated components. These tests should be carried out after component testing (where individual components are tested in isolation) but before system testing (where the entire system is tested as a whole). This ensures that any issues arising from the integration of components are identified and resolved early in the testing process, making option D the correct answer.

NEW QUESTION 203

Which of the following is a CORRECT statement about how a tester should communicate about defects, test results, and other test information?

- A. Testers should include personal opinions and judgements in defect reports and review findings
- B. Testers should emphasize the benefits of testing, such as increased quality and reduced risk
- C. Testers should reject all questions about their test findings and information
- D. Testers should take a command-and-control approach with the project team

Answer: B

Explanation:

Communication from testers about defects, test results, and other test information should emphasize the benefits of testing such as increased quality and reduced risk. This positive framing helps in reinforcing the value of testing and ensuring stakeholders understand the contribution of testing to the overall project success (ISTQB not-for-profit association).References:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf

NEW QUESTION 204

Which of the following does MOT describe a reason why testing is necessary?

- A. The customer decided that 100% branch coverage shall be achieved
- B. The acquisition of test automation tools was based on the assumption that it will be used in all projects
- C. For avionics and pharmaceutical systems software testing is mandated by standards
- D. The risks associated with delivering the system are far higher than the cost of testing

Answer: B

Explanation:

Testing is necessary for various reasons, such as:

- ? To detect defects and failures that may affect the quality, performance, reliability or security of a software product or system
 - ? To verify that a software product or system meets its specified requirements, expectations and standards
 - ? To validate that a software product or system fulfills its intended purpose and satisfies its stakeholders' needs
 - ? To provide information and feedback about the status and risks of a software product or system
 - ? To comply with regulations or contractual obligations that mandate testing for certain types of software products or systems
- The following statements describe some reasons why testing is necessary:
- ? A) The customer decided that 100% branch coverage shall be achieved. This is a reason why testing is necessary, as it reflects a contractual obligation or a quality standard that requires testing to measure and achieve a certain level of code coverage.
 - ? C) For avionics and pharmaceutical systems software testing is mandated by standards. This is a reason why testing is necessary, as it reflects a regulation or a compliance requirement that mandates testing for certain types of software products or systems that have high safety or security risks.
 - ? D) The risks associated with delivering the system are far higher than the cost of testing. This is a reason why testing is necessary, as it reflects a risk-based approach that considers testing as an investment to reduce the probability and impact of potential failures or defects. The following statement does not describe a reason why testing is necessary:
 - ? B) The acquisition of test automation tools was based on the assumption that it will be used in all projects. This is not a reason why testing is necessary, as it reflects a business decision or a resource allocation that does not justify the need or purpose of testing. Test automation tools are not always suitable or beneficial for all projects, and testing can be performed with or without test automation tools. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 5-6.

NEW QUESTION 209

Which of the following are valid testing principles?

- I) Exhaustive testing is in general impossible.
- II) Exhaustive testing should be executed for code intended to be reused.
- III) Testing may guarantee that a program is correct.
- IV) Testing cannot guarantee that a program is correct.
- V) Defects cluster together in certain areas of the product.

- A. I, IV, V
- B. II, IV
- C. I, V
- D. I, III

Answer: A

Explanation:

Statements I, IV and V are valid testing principles according to the ISTQB syllabus. Statement I states that exhaustive testing is in general impossible, because it would require testing all possible inputs, outputs and combinations of states, which is usually impractical or impossible. Statement IV states that testing cannot guarantee that a program is correct, because testing can only show the presence of defects, not their absence. Statement V states that defects cluster together in certain areas of the product, which means that some modules or functions are more likely to contain defects than others. Statements II and III are invalid testing principles. Statement II states that exhaustive testing should be executed for code intended to be reused, which contradicts statement I. Statement III states that testing may guarantee that a program is correct, which contradicts statement IV. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, pages 4-5.

NEW QUESTION 211

Which of the following lists factors That contribute to PROJECT risks?

- A. skill and staff shortages; problems in defining the right requirements, contractual issues.
- B. skill and staff shortages; software does not perform its intended functions; problems in defining the right requirements.
- C. problems in defining the right requirements; contractual issues; poor software quality characteristics.
- D. poor software quality characteristics; software does not perform its intended functions.

Answer: A

Explanation:

Project risks are the uncertainties or threats that may affect the project objectives, such as scope, schedule, cost, and quality. According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, some of the factors that contribute to project risks are:

- ? Skill and staff shortages: This factor refers to the lack of adequate or qualified human resources to perform the project tasks. This may result in delays, errors, rework, or low productivity.
 - ? Problems in defining the right requirements: This factor refers to the difficulties or ambiguities in eliciting, analyzing, specifying, validating, or managing the requirements of the project. This may result in misalignment, inconsistencies, gaps, or changes in the requirements, affecting the project scope and quality.
 - ? Contractual issues: This factor refers to the challenges or disputes that may arise from the contractual agreements between the project parties, such as clients, suppliers, vendors, or subcontractors. This may result in legal, financial, or ethical risks, affecting the project delivery and satisfaction.
- The other options are not correct because they list factors that contribute to PRODUCT risks, not project risks. Product risks are the uncertainties or threats that may affect the quality or functionality of the software product or system. Some of the factors that contribute to product risks are:
- ? Poor software quality characteristics: This factor refers to the lack of adherence or compliance to the quality attributes or criteria of the software product or system, such as reliability, usability, security, performance, or maintainability. This may result in defects, failures, or dissatisfaction of the users or stakeholders.
 - ? Software does not perform its intended functions: This factor refers to the deviation or discrepancy between the expected and actual behavior or output of the software product or system. This may result in errors, faults, or malfunctions of the software product or system.
- References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 1: Fundamentals of Testing, Section 1.5: Risks and Testing, Pages 14-16.

NEW QUESTION 216

Software was found to take much more time than the stated requirement of less than one second to save a file. Upon investigation it was found that there was an unnecessary check inside a loop which was slowing down the file-save operation. The software not being able to meet the desired response time is an example of

- A. It is not a defect
- B. Defect
- C. Error
- D. Failure

Answer: D

Explanation:

A failure is an event in which a component or system does not perform a required function within specified limits. A failure is observable by the software users or other stakeholders. A failure is caused by one or more defects in the software. In this case, the software not being able to meet the desired response time is an example of a failure, as it deviates from the stated requirement and affects the user experience. It is not a defect, which is a flaw in the software that causes the failure. It is not an error, which is a human action that produces an incorrect result. It is not a non-defect, as it clearly violates a specified requirement. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 4.

NEW QUESTION 220

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your ISTQB-CTFL Exam with Our Prep Materials Via below:

<https://www.certleader.com/ISTQB-CTFL-dumps.html>