

## ISTQB-CTFL Dumps

### ISTQB-Foundation Level Exam

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



**NEW QUESTION 1**

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 2**

Who of the following has the best knowledge to decide what tests in a test project should be automated?

- A. The developer
- B. The customer
- C. The development manager
- D. The test leader

**Answer: D**

**Explanation:**

The test leader is the person who is responsible for planning, monitoring, and controlling the test activities and resources in a test project. The test leader should have the best knowledge of the test objectives, scope, risks, resources, schedule, and quality criteria. The test leader should also be aware of the test automation criteria, such as the execution frequency, the test support, the team education, the roles and responsibilities, and the devs and testers collaboration<sup>1</sup>. Based on these factors, the test leader can decide which tests are suitable for automation and which are not, and prioritize them accordingly. The test leader can also coordinate with the test automation engineers, the developers, and the stakeholders to ensure the alignment of the test automation strategy with the test project goals and expectations. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 2, Section 2.3.1, Page 152; ISTQB Glossary of Testing Terms v4.0, Page 403; ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 6, Section 6.1.1, Page 514; Top 8 Test Automation Criteria You Need To Fulfill - QAMIND<sup>1</sup>

**NEW QUESTION 3**

Given the following statements:

- \* 1. It can prevent defects by manual examination of the functional specification
- \* 2. It is effective since it can be performed very early in the software development life cycle
- \* 3. It can detect the failures in the running application
- \* 4. It can help eliminate defects in user stories 5. It can verify externally visible behaviors

Which set of statements represent values ONLY for static testing?

- A. 1,3, 4,5
- B. 2,4,5
- C. 1,2,4
- D. 1,2, 3, 4,5

**Answer: C**

**Explanation:**

Static testing involves reviewing and inspecting the code, requirements, or design documents without executing the code. It can prevent defects, is effective early in the software development life cycle, and can help eliminate defects in user stories. Option 1: "It can prevent defects by manual examination of the functional specification" - This is a value of static testing.

Option 2: "It is effective since it can be performed very early in the software development life cycle" - This is a value of static testing.

Option 3: "It can detect the failures in the running application" - This is a v (ISTQB not-for-profit association) namic testing, not static testing.

Option 4: "It can help eliminate defects in user stories" - This is a value of static testing. Option 5: "It can verify externally visible behaviors" - This is a value of dynamic testing, not static testing.

Therefore, the correct set of statements representing values only for static testing is 1, 2, 4, which corresponds to answer C<sup>6</sup>†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

**NEW QUESTION 4**

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 5**

Which of the following software development models BEST exemplifies a model that does NOT support the principle of early testing?

- A. The iterative development model
- B. The V-model
- C. The Waterfall model
- D. The incremental development model

**Answer: C**

**Explanation:**

The Waterfall model exemplifies a software development model that does not support the principle of early testing. In the Waterfall model, each phase must be completed before the next begins, which delays testing until after the completion of the earlier phases like requirements gathering and design. This can often result in finding defects later in the development cycle, making them more expensive and time-consuming to fix (ISTQB not-for-profit association) (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](https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf)

? ISTQB News Release on CTFL v4.0: <https://www.istqb.org/news/posts/istqb-releases-certified-tester-foundation-level-v40-ctfl/>

**NEW QUESTION 6**

A team's test strategy was to invest equal effort in testing each of a system's modules. After running one test cycle, it turned out that most of the critical bugs were detected in one of the system's modules.

Which testing principal suggests a change to the current test strategy for the next test cycle?

- A. Pesticide Paradox
- B. Early testing
- C. Absence-of-errors fallacy
- D. Defect clustering

**Answer: D**

**Explanation:**

Defect clustering is a testing principle that states that a small number of modules contain most of the defects detected during pre-release testing, or are responsible for most of the operational failures. Defect clustering can be explained by Pareto's principle (also known as the 80-20 rule), which states that approximately 80% of the problems are found in 20% of the modules. Defect clustering suggests a change to the current test strategy for the next test cycle, as it implies that more effort should be allocated to test the modules that have shown high defect density or criticality. Pesticide paradox is another testing principle that states that if the same tests are repeated over and over again, eventually they will no longer find any new defects. Pesticide paradox suggests a change to the current test strategy for the next test cycle, but not based on defect clustering, but rather on test diversity and coverage. Early testing is a testing principle that states that testing activities should start as early as possible in the software development life cycle and should be focused on defined objectives. Early testing does not suggest a change to the current test strategy for the next test cycle, but rather a proactive approach to prevent defects from occurring or propagating. Absence-of-errors fallacy is a testing principle that states that finding and fixing defects does not help if the system built is unusable and does not fulfill the users' needs and expectations. Absence-of-errors fallacy does not suggest a change to the current test strategy for the next test cycle, but rather a focus on quality attributes and user requirements. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 1, page 9-10.

**NEW QUESTION 7**

Which or the following would be a key difference between a peer review of code and static analysis of code using a tool?

- A. A peer reviews finds defects while static analysis finds failures.
- B. Static analysis targets the code technically whereas Peer review is applicable to further aspects.
- C. Peer reviews cannot find missing requirements whereas static analysis can
- D. A peer reviews find failures while static analysis finds defects.

**Answer: B**

**Explanation:**

The key difference between a peer review of code and static analysis of code using a tool lies in their approaches and scope. A peer review is a manual inspection of the code by peers or colleagues, focusing not only on the technical aspects of the code but also on other elements such as design, compliance with standards, and maintainability. Peer reviews can identify defects, suggest improvements, and ensure that the code adheres to best practices and team standards. On the other hand, static analysis is an automated process performed by tools designed to analyze the code without executing it. These tools can detect potential issues such as syntax errors, vulnerabilities, and code smells based on predefined rules and patterns. While static analysis is technically focused, it lacks the

broader perspective that human reviewers can provide, such as evaluating the code's maintainability or adherence to project-specific standards. Therefore, static analysis targets the code technically, whereas peer review encompasses a wider range of aspects, making option B the correct answer.

**NEW QUESTION 8**

Which of the following definitions is NOT true?

- A. Test data preparation tools fill databases, create files or data transmissions to set up test data to be used during the execution of tests.
- B. Test execution tools execute test objects using automated test scripts.
- C. Test Management tools monitor and report on how a system behaves during the testing activities.
- D. Test comparators determine differences between files, databases or test results.

**Answer: C**

**Explanation:**

Test Management tools are designed to support the planning, execution, and monitoring of the testing process. They provide features for managing test cases, test runs, tracking defects, and reporting on testing activities. However, the statement in option C describes Test Management tools as monitoring and reporting on the system's behavior during testing activities, which is not accurate. Test Management tools focus on the testing process itself rather than on the behavior of the system under test.

? Test data preparation tools (A) indeed create and manage test data for use during test execution.

? Test execution tools (B) automate the execution of test cases and the comparison of actual outcomes against expected results.

? Test comparators (D) are tools that compare actual outcomes with expected outcomes, highlighting discrepancies.

Therefore, option C is the correct answer as it inaccurately describes the function of Test Management tools.

**NEW QUESTION 9**

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

Which coverage metric will provide the needed information for this analysis?

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

**Answer: D**

**Explanation:**

Branch coverage is a type of structural coverage metric that measures the percentage of branches or decision outcomes that are executed by the test cases. A branch is a point in the code where the control flow can take two or more alternative paths based on a condition. For example, an if-else statement is a branch that can execute either the if-block or the else-block depending on the evaluation of the condition. Branch coverage ensures that each branch is taken at least once by the test cases, and thus reveals the behavior of the software under different scenarios. Branch coverage is also known as decision coverage or all-edges coverage.

Branch coverage is suitable for testing the cases where a specific action is not allowed, because it can verify that the test cases cover all the possible outcomes of the conditions that determine the action. For example, if the program has a condition that checks if the manufacturing stage can be stopped, then branch coverage can ensure that the test cases cover both the cases where the stage can be stopped and where it cannot be stopped. This way, branch coverage can help identify any missing or incorrect branches that may lead to undesired or unsafe actions.

The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types of coverage metrics, such as statement coverage, branch coverage, data flow coverage, etc. Code coverage does not specify which type of coverage metric is used for the analysis. Data flow coverage is a type of structural coverage metric that measures the percentage of data flow paths that are executed by the test cases. A data flow path is a sequence of statements that define, use, or kill a variable. Data flow coverage is useful for testing the correctness and completeness of the data manipulation in the software, but not for testing the conditions that determine the actions. Statement coverage is a type of structural coverage metric that measures the percentage of statements or lines of code that are executed by the test cases. Statement coverage ensures that each statement is executed at least once by the test cases, but it does not reveal the behavior of the software under different scenarios. Statement coverage is a weaker criterion than branch coverage, because it does not account for the branches or decision outcomes in the code. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

**NEW QUESTION 10**

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 10**

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 13**

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 made (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 D.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

**NEW QUESTION 15**

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 18**

Why should you choose a test technique?

- A. Because you need to match the way you test to the content of the product under test
- B. Because of the time constraints that usually accompany a test project
- C. Because this way you cover the full scope of the product's functionality
- D. Because choosing a test technique is a common practice in software testing

**Answer: A**

**Explanation:**

You should choose a test technique because you need to match the way you test to the content of the product under test. A test technique is a method or process for deriving and selecting test cases based on some criteria or rules. Different test techniques are suitable for different types of software products, depending on their characteristics, functionalities, requirements, specifications, risks, etc. Choosing a test technique helps to ensure that the test cases are relevant, effective, and efficient for the product under test. The other options are not correct reasons to choose a test technique. Time constraints are not a factor for choosing a test technique, but rather for prioritizing or optimizing testing activities. Covering the full scope of the product's functionality is not a guarantee of choosing a test technique, but rather a goal of testing. Choosing a test technique is not a common practice in software testing, but rather a professional skill and responsibility. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 31.

**NEW QUESTION 22**

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 27**

Given the following state model of sales order software: SEE ATTACHMENT

Which of the following sequences of transitions provides the highest level of transition coverage for the model (assuming you can start in any state)?

- A. IN PRODUCTION -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED -> PLACED
- B. IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION
- C. PLACED -> IN PRODUCTION -> SHIPPED -> CANCELLED -> PLACED
- D. PLACED -> CANCELLED -> PLACED -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED

**Answer: B**

**Explanation:**

State transition testing is a black-box testing technique where test cases are designed to cover states and transitions of a state machine.

Given the state model with the following transitions:

- ? PLACED -> IN PRODUCTION
- ? IN PRODUCTION -> CANCELLED
- ? IN PRODUCTION -> SHIPPED
- ? SHIPPED -> INVOICED
- ? INVOICED -> CANCELLED
- ? CANCELLED -> PLACED

To cover all transitions at least once, we need to create a sequence that covers all six transitions.

Option A: IN PRODUCTION -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED -> PLACED- Misses SHIPPED -> INVOICED and INVOICED -> CANCELLED transitions.

Option B: IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION- Covers all transitions.

Option C: PLACED -> IN PRODUCTION -> SHIPPED -> CANCELLED (ISTQB not-for-profit association) (Udemy)sses INVOICED -> CANCELLED transition.

Option D: PLACED -> CANCELLED -> PLACED -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED- Misses SHIPPED -> INVOICED and INVOICED -> CANCELLED transitions.

Given these, Option B covers all the transitions6†source9†source. References:

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

**NEW QUESTION 28**

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 32**

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 34**

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 bugs<sup>12</sup>. 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)<sup>3</sup>. 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 failures<sup>4</sup>. 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 false<sup>5</sup>. 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 variables<sup>6</sup>. 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 36**

In which of the following test documents would you expect to find test exit criteria described<sup>9</sup>

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

**Answer: D**

**Explanation:**

Test exit criteria are the conditions that must be fulfilled before concluding a particular testing phase. These criteria act as a checkpoint to assess whether we have achieved the testing objectives and are done with testing<sup>1</sup>. Test exit criteria are typically defined in the test plan document, which is one of the outputs of the test planning phase. The test plan document describes the scope, approach, resources, and schedule of the testing activities. It also identifies the test items, the features to be tested, the testing tasks, the risks, and the test deliverables<sup>2</sup>. According to the ISTQB® Certified Tester Foundation Level Syllabus v4.0, the test plan document should include the following information related to the test exit criteria<sup>3</sup>:

? The criteria for evaluating test completion, such as the percentage of test cases

executed, the percentage of test coverage achieved, the number and severity of defects found and fixed, the quality and reliability of the software product, and the stakeholder satisfaction.

? The criteria for evaluating test process improvement, such as the adherence to the

test strategy, the efficiency and effectiveness of the testing activities, the lessons learned and best practices identified, and the recommendations for future

improvements.

Therefore, the test plan document is the most appropriate test document to find the test exit criteria described. The other options, such as test design specification, project plan, and requirements specification, are not directly related to the test exit criteria. The test design specification describes the test cases and test procedures for a specific test level or test type<sup>3</sup>. The project plan describes the overall objectives, scope, assumptions, risks, and deliverables of the software project<sup>4</sup>. The requirements specification describes the functional and non-functional requirements of the software product<sup>5</sup>. None of these documents specify the conditions for ending the testing process or evaluating the testing outcomes. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Entry and Exit Criteria in Software Testing | Baeldung on Computer Science, Entry And Exit Criteria In Software Testing - Rishabh Software, Entry and Exit Criteria in Software Testing Life Cycle - STLC [2022 Updated] - Testsigma Blog, ISTQB® releases Certified Tester Foundation Level v4.0 (CTFL).

### NEW QUESTION 39

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 41

Which of the following is NOT an objective of testing?

- A. Finding defects
- B. Providing information for decision-making
- C. Gaining confidence about the level of quality of the software
- D. Analyzing and removing the cause of failures

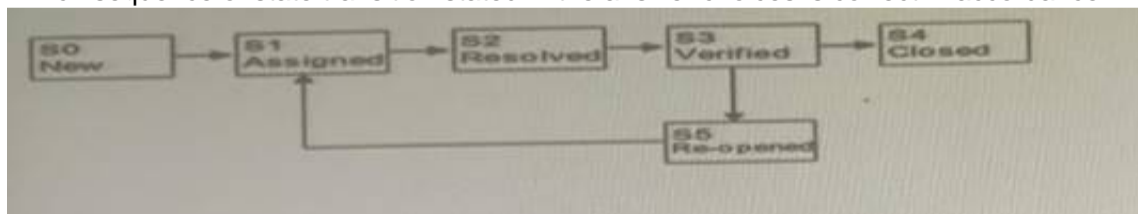
**Answer: D**

#### Explanation:

Analyzing and removing the cause of failures is not an objective of testing, but rather a task of development or maintenance. A failure is an event or behavior that deviates from the expected or specified result of a system under test. A failure is caused by an error (also known as a mistake or a fault) in the software code, design, or specification. Analyzing and removing the cause of failures is a process of locating and fixing errors in the software code, design, or specification, which is also known as debugging or defect resolution. Analyzing and removing the cause of failures does not aim to find or report defects, but rather to correct or prevent them. The other options are objectives of testing. Finding defects is one of the main objectives of testing, as it helps to improve the quality and reliability of the software product. Providing information for decision-making is another objective of testing, as it helps to support decision making and risk management. Gaining confidence about the level of quality of the software is another objective of testing, as it helps to assure that the software product meets its requirements and customer or user needs and expectations. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 3.

### NEW QUESTION 43

Which sequence of state transition 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->S4
- B. S0->S1->S2->S3->S5^>S1
- C. S0->S1->S2->S3->S5->S1->S2->S3
- D. S0->S1->S2->S3->S5->S3->S4

**Answer: C**

#### Explanation:

The figure depicts the life-cycle of a defect using state transition testing. State transition testing is a technique that models how a system transitions from one state to another depending on events or conditions. The figure shows six states (S0 to S5) and seven transitions (T0 to T6). The correct sequence of state transitions that follows the figure is S0->S1->S2->S3->S5->S1->S2->S3. This sequence represents the following scenario:

? S0: The defect is not yet detected (initial state).

? T0: The defect is detected by testing (event).

? S1: The defect is reported and registered (state).  
? T1: The defect is assigned to a developer for fixing (event).  
? S2: The defect is being fixed by the developer (state).  
? T2: The developer fixes the defect and delivers a new version (event).  
? S3: The defect is verified by testing (state).  
? T5: The testing fails to confirm that the defect is fixed (event).  
? S5: The defect is rejected by testing (state).  
? T6: The defect is reassigned to a developer for fixing (event).  
? S1: The defect is reported and registered (state).  
? T1: The defect is assigned to a developer for fixing (event).  
? S2: The defect is being fixed by the developer (state).  
? T2: The developer fixes the defect and delivers a new version (event).  
? S3: The defect is verified by testing (state). The other sequences are incorrect, as they do not follow the transitions shown in the figure. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 40-41.

**NEW QUESTION 46**

I When closing the test activities, all the testware resources can be uninstalled and released  
II All the testware should be subject to Configuration Management  
III. The testware. at the end of the project, should be transferred to the organization responsible for maintenance  
IV The developers are responsible for the correct installation of the testware

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

**Answer: A**

**Explanation:**

Testware is a term that refers to all artifacts produced during the testing process, such as test plans, test cases, test scripts, test data, test results, defect reports, etc. The following statements about testware are correct:

? II) All the testware should be subject to Configuration Management. Configuration management is a process that establishes and maintains consistency among work products throughout their life cycle. Configuration management applies to all testware, as it helps ensure their quality and consistency, track their changes and defects, control their versions and access rights, and link them to other artifacts.

? III) The testware at the end of the project should be transferred to the organization responsible for maintenance. Maintenance testing is testing performed on a software product after delivery to correct defects or improve performance or other attributes. Maintenance testing requires testware from previous testing activities or phases, such as test cases, test data, test results, etc. Therefore, the testware at the end of the project should be transferred to the organization responsible for maintenance testing, such as support team or maintenance team. The following statements about testware are incorrect:

? I) When closing the test activities, all the testware resources can be uninstalled and released. This statement is incorrect, as some testware resources may still be needed for future testing activities or phases, such as maintenance testing or regression testing. Therefore, when closing the test activities, some testware resources should be archived and stored for future use, while others can be uninstalled and released.

? IV) The developers are responsible for the correct installation of the testware. This statement is incorrect, as the testers are responsible for the correct installation of the testware. The testers should ensure that they have access to all necessary testware resources and that they are installed and configured properly before starting the test execution. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 6, page 58-61.

**NEW QUESTION 48**

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

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

**Answer: D**

**Explanation:**

? A product risk is a risk that affects the quality or timeliness of the software product being developed or tested<sup>1</sup>. Product risks are related to the requirements, design, implementation, verification, and maintenance of the software product<sup>2</sup>.

? The risk of the sub-contractor failing to meet his commitment is a product risk, as it could cause a delay in the completion of the testing required for the current cycle, which in turn could affect the release date of the product. The release date is an important aspect of the product quality, as it reflects the customer satisfaction and the market competitiveness of the product<sup>3</sup>.

? The other options are not correct because: References =

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

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

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

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

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

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

**NEW QUESTION 50**

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 54**

Which of the following BEST describes checklist-based testing?

- A. Checklist-based testing includes formal tests from detailed lists of test conditions, allowing much repeatability
- B. Checklist-based testing may involve a list of tests based on what is important to the user as well as an understanding of why and how software fails
- C. Checklist-based testing, while popular, provides little consistency and few guidelines concerning test case development
- D. Checklist-based testing is restricted to non-functional testing, including usability, performance, and security test

**Answer: B**

**Explanation:**

Checklist-based testing involves using checklists that contain items, such as potential test conditions, that should be tested. These checklists are often based on insights into what is important to the user, potential areas where software might fail, and specific aspects that need to be tested. It provides a structured yet flexible approach to testing, ensuring key areas are covered while allowing testers to use their experience and understanding of the system. Checklist-based testing is not limited to non-functional testing but can be applied to various types of testing, including functional testing.

References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4.5.

**NEW QUESTION 58**

Which of the following statements BEST describes how test cases are derived from a use case?

- A. Test cases are derived based on non-functional requirements such as usability
- B. Test cases are created using white-box test techniques to execute scenarios of use cases
- C. Test cases are derived based on pair testing between a user and a tester to find defects
- D. Test cases are designed to cover various user behaviors, including basic, exceptional or alternative and error behaviors associated with human users or systems

**Answer: D**

**Explanation:**

Use cases describe a system's behavior as it responds to a request from a user. They typically consist of various scenarios, such as basic flow, alternative flow, and exceptional flow, which represent possible behaviors when a user interacts with the system. When deriving test cases from use cases, it is important to cover these different types of user behaviors. Test cases should be designed to verify how the system behaves during each of these scenarios. This ensures that the system operates correctly for normal and error conditions encountered by human users or systems interacting with the application. Thus, test cases derived from use cases aim to cover basic, exceptional, and alternative flows, ensuring comprehensive coverage.

References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.2.4.

**NEW QUESTION 59**

Mark the correct sentences:

- \* Defects are a result of environmental conditions and are also referred to as "Failures"
- \* A human mistake may produce a defect
- \* A system may totally fail to operate correctly when a failure exists in it
- \* When a defect exists in a system it may result in a failure
- \* Defects occur only as a result of technology changes

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

**Answer: A**

**Explanation:**

? The question is about marking the correct sentences among the given statements related to defects, failures, and mistakes. According to the ISTQB glossary, the definitions of these terms are1:

? Therefore, out of the five given statements, only two are correct, namely:

? The other three statements are incorrect, namely: References:

? 1: ISTQB Glossary of Testing Terms 4.0, 2023, available at ISTQB) and ASTQB).

**NEW QUESTION 60**

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 61**

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 62**

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 65**

Which of the following statements about test estimation approaches is CORRECT?

- A. The Wideband Delphi estimation technique is an example of the risk-based approach
- B. The Wideband Delphi estimation technique is an example of the expert-based approach
- C. Burndown charts used in Agile development is an example of the risk-based approach
- D. Burndown charts used in Agile development is an example of the expert-based approach

**Answer:** B

**Explanation:**

There are two main approaches to test estimation:

? Expert-based approach:

? Metrics-based approach:

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, the expert-based approach relies on experts' experience and knowledge, which aligns with the Wideband Delphi technique.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

**NEW QUESTION 69**

The testers in company A were part of the development team. Due to an organizational change they moved to be part of the support team. What are the advantages and the disadvantages of this change?

- A. Advantage: More independence in deciding what and how to test, Disadvantage: Isolation from me development team knowledge
- B. Advantage: being closer to customer perspective, Disadvantage less independence in perspectives
- C. Advantage: pulled to support tasks and having less time for testing, Disadvantage less chances to move a tester to development
- D. Advantage: increased chances to move a tester to development; Disadvantage: pulled to support tasks and having less time for testing

**Answer: B**

**Explanation:**

Being part of the support team means that the testers are closer to the customer perspective, which is an advantage for testing, as they can better understand the user needs and expectations, and identify more realistic scenarios and risks. However, being part of the support team also means that they have less independence in deciding what and how to test, as they may be influenced by the customer's preferences or requests, which could compromise the objectivity and effectiveness of testing. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 6.

**NEW QUESTION 74**

Which of the following statements is not correct?

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

**Answer: A**

**Explanation:**

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

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

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

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

**NEW QUESTION 79**

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 80**

Which of the following statements is the BEST example of non-functional testing?

- A. Tests which capture the time it takes to save a file
- B. Tests which calculate overtime pay for those employees entitled to such
- C. Tests related to "what" the system should do
- D. Tests based on the internal structure of a component or system

**Answer: A**

**Explanation:**

Non-functional testing refers to testing aspects that do not relate to specific behaviors or functions of the software but to attributes such as performance, usability, reliability, etc. Tests that capture the time it takes to save a file directly relate to the performance of the system, thus falling under non-functional testing. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.2.5 "Functional and Non- functional Testing".

**NEW QUESTION 85**

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 88**

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 92**

Which of the following exemplifies how a software bug can cause harm to a company?

- A. "Print" prints the last page twice for a file with 1000 pages  
B. The timeout on the login page of a web site is 9 minutes, while the requirement was for 10 minutes  
C. When uninstalling the application, the uninstall dialog has a spelling mistake  
D. When calculating the final price in a shopping list, the price of the last item is not added

**Answer: D**

**Explanation:**

A software bug can cause harm to a company by directly affecting its operations, reputation, user satisfaction, and financials. Option D, "When calculating the final price in a shopping list, the price of the last item is not added," describes a defect that directly impacts the core functionality of a financial transaction, potentially leading to financial loss and customer dissatisfaction. This can have severe implications for the company's credibility and revenue. Options A, B, and C describe bugs that, while potentially annoying, do not have the same direct impact on the company's core operations and financial integrity as option D.

**NEW QUESTION 93**

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<sup>6†</sup>source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

#### NEW QUESTION 94

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 97

Consider the following statements about risk-based testing.

I) Risk-based testing has the objective to reduce the level of protect risks.

II) Tests should be prioritized to find tie critical detects as early as possible.

III) Non-testing activities may also help to reduce risk

IV) Risks have to be reassessed on a regular basis.

V) The project stakeholders can give useful input to determine the risks

A. I III IV and V are tru

B. II is false.

C. II, III IV and V are correc

D. I is false.

E. I, II and IV are tru

F. III and V are false.

G. II, III and V are tru

H. 1 ants Iv are false.

**Answer: B**

#### Explanation:

The following statements about risk-based testing are correct:

? II) Tests should be prioritized to find tie critical detects as early as possible. Risk- based testing involves prioritizing tests based on risk level, which reflects both the likelihood and impact of defects or failures. Tests with higher risk level should be executed earlier than tests with lower risk level, in order to find and fix critical defects as soon as possible.

? III) Non-testing activities may also help to reduce risk. Risk-based testing does not

only involve testing activities, but also other activities that can help mitigate risks, such as reviews, inspections, audits, simulations or prototyping.

#### NEW QUESTION 100

A system has a self-diagnostics module that starts executing after the system is reset. The diagnostics are running 12 different tests on the systems memory hardware. The following is one of the requirements set for the diagnostics module:

'The time taking the diagnostics tests to execute shall be less than 2 seconds' Which of the following is a failure related to the specified requirement?

A. The diagnostic tests fail to start after a system reset

B. The diagnostic tests take too much time to execute

C. The diagnostic tests that measure the speed of the memory, fail

D. The diagnostic tests fail due to incorrect implementation of the test code

**Answer:** B

**Explanation:**

A failure is an event in which a component or system does not perform a required function within specified limits<sup>1</sup>. A requirement is a condition or capability needed by a user to solve a problem or achieve an objective<sup>2</sup>. In this case, the requirement is that the diagnostics tests should execute in less than 2 seconds. Therefore, any event that violates this requirement is a failure. The only option that clearly violates this requirement is B. The diagnostic tests take too much time to execute. If the diagnostic tests take more than 2 seconds to complete, then they do not meet the specified limit and thus fail. The other options are not necessarily failures related to the specified requirement. Option A. The diagnostic tests fail to start after a system reset is a failure, but not related to the time limit. It is related to the functionality of the self-diagnostics module. Option C. The diagnostic tests that measure the speed of the memory, fail is also a failure, but not related to the time limit. It is related to the accuracy of the memory tests. Option D. The diagnostic tests fail due to incorrect implementation of the test code is also a failure, but not related to the time limit. It is related to the quality of the test code. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Requirements Engineering Fundamentals.

**NEW QUESTION 103**

Given the following state model of sales order software: SEE ATTACHMENT

Which of the following sequences of transitions provides the highest level of transition coverage for the model (assuming you can start in any state)?

- A. IN PRODUCTION -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED -> PLACED
- B. IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION
- C. PLACED -> IN PRODUCTION -> SHIPPED -> CANCELLED -> PLACED
- D. PLACED -> CANCELLED -> PLACED -> CANCELLED -> PLACED -> IN PRODUCTION -> CANCELLED

**Answer:** B

**Explanation:**

To achieve the highest level of transition coverage, one must consider all the possible transitions between the states in the given state model of the sales order software. The transitions in the sequence provided in Option B - "IN PRODUCTION -> SHIPPED -> INVOICED -> CANCELLED -> PLACED -> IN PRODUCTION" cover all the states and transitions effectively. This covers the transitions from IN PRODUCTION to SHIPPED, SHIPPED to INVOICED, INVOICED to CANCELLED, CANCELLED to PLACED, and PLACED to IN PRODUCTION, thereby maximizing the transition coverage. References: ? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.3.5.

**NEW QUESTION 106**

Which of the following is NOT an experience-based technique?

- A. Boundary value analysis.
- B. Error guessing
- C. Exploratory testing
- D. Fault attack

**Answer:** A

**Explanation:**

Boundary value analysis is not an experience-based technique, but rather a specification-based technique (also known as black-box technique). Experience-based techniques are 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. Some examples of experience-based techniques are error guessing, exploratory testing, fault attack, checklist-based testing, etc. Specification-based techniques are techniques that rely on the tester's analysis and interpretation of the requirements or specifications of the system under test to derive and select test cases based on some criteria or rules. Some examples of specification-based techniques are equivalence partitioning, boundary value analysis, decision table testing, state transition testing, etc. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 31.

**NEW QUESTION 111**

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 113**

Which of the following statements about re-testing and regression testing are TRUE? I Re-testing should be performed after a defect is fixed.

II Regression testing should always be performed after a defect is fixed.

III. Re-testing and regression testing may be performed at any test level.

IV Regression testing may include functional, non-functional and structural testing.

- A. Re-testing should be included in the debugging activity.
- B. I, II
- C. IV
- D. II, V
- E. I, III

F. I  
G. IV, V

**Answer:** A

**Explanation:**

The following statements about re-testing and regression testing are true:

? I) Re-testing should be performed after a defect is fixed. Re-testing is a type of testing that verifies that a defect has been successfully resolved by executing a test case that previously failed due to that defect. Re-testing should be performed after a defect is fixed and delivered to ensure that it does not cause any new failures or side effects.

? III) Re-testing and regression testing may be performed at any test level. Re-testing and regression testing are not limited to a specific test level, but can be applied at any level depending on the context and objectives. For example, re-testing and regression testing can be performed at unit level, integration level, system level or acceptance level.

? IV) Regression testing may include functional, non-functional and structural testing. Regression testing is a type of testing that verifies that previously tested software still performs correctly after changes. Regression testing may include various types of testing depending on the scope and purpose of the changes. For example, regression testing may include functional testing to check if the software meets its requirements, non-functional testing to check if the software meets its quality attributes, or structural testing to check if the software meets its design or code standards. The following statement about re-testing and regression testing is false:

? II) Regression testing should always be performed after a defect is fixed.

Regression testing is not always necessary after a defect is fixed, as some defects may have a low impact or low likelihood of affecting other parts of the software.

Regression testing should be performed after a defect is fixed only if there is a risk of introducing new defects or causing existing defects due to the changes made to fix the defect. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 2, page 19; Chapter 4, page 45.

**NEW QUESTION 118**

How can testing contribute to higher quality?

- A. Testing help to measure the quality of software.
- B. Testing ensures that remaining defects are documented.
- C. Testing removes errors in the software.
- D. Testing eliminates the risk with software.

**Answer:** A

**Explanation:**

Testing can contribute to higher quality by helping to measure the quality of software. Quality is defined as the degree to which a component or system satisfies specified requirements and customer or user needs and expectations. Testing is a process of evaluating a component or system by applying inputs and observing outputs, and comparing them with expected results. Testing can help to measure the quality of software by providing information on its functionality, performance, usability, security, reliability, etc. Testing can also help to identify and report defects in software, which can lead to improvement actions and quality assurance activities. The other options are not accurate descriptions of how testing can contribute to higher quality. Testing does not ensure that remaining defects are documented, but rather that detected defects are reported. Testing does not remove errors in software, but rather finds defects in software behavior or quality. Testing does not eliminate the risk with software, but rather assesses and manages the risk with software. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 3.

**NEW QUESTION 123**

Which of the following BEST describes exploratory testing?

- A. Exploratory testing is a suitable test technique which may replace both black-box and white-box test techniques
- B. Exploratory testing is a valid and useful black-box test technique since it focuses on test cases related to the architecture and design of a system
- C. Exploratory testing requires both solid specifications and much project time available for test execution
- D. Exploratory testing may be used within defined time periods, during which the tester may follow a test charter as a guide

**Answer:** D

**Explanation:**

Exploratory testing involves simultaneous test design and execution and is guided by a test charter, which outlines what needs to be tested, how it should be tested, and what to look for. This technique is typically conducted within predefined time periods, known as time-boxing, which allows testers to explore a system, understand its functionalities, and identify potential issues without detailed documentation or prior test case planning. The key aspects of exploratory testing include flexibility, adaptability, and the ability to respond to system behavior during testing. References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4.2.

**NEW QUESTION 128**

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 130**

You are testing an e-commerce system that sporadically fails to properly manage customers' shopping carts. You have stressed the urgency of this situation to the development manager and development team and they recognize the priority of resolving the underlying defect. The development team is waiting for more information, which you will include in your defect report. Given the following items of information they are included in a typical defect report:

- \* 1. The expected results
- \* 2. The actual results
- \* 3. The urgency and priority to fix this
- \* 4. The date and author of the defect report
- \* 5. A description of the defect in order to reproduce, including screenshots and database dumps

Which of these items will be MOST useful to the developers to help them identify and remove the defect causing this failure?

- A. 1, 2, 5
- B. 1, 2, 3, 4, 5
- C. 1, 2, 4
- D. 3, 4

**Answer:** A

**Explanation:**

When developers are trying to identify and remove a defect, they need clear information on what went wrong and what was expected. The items that will be most useful to developers in this context are the expected results (item 1), the actual results (item 2), and a description of the defect including steps to reproduce, screenshots, and database dumps (item 5). This information helps developers understand the nature of the defect and provides the necessary details to reproduce and diagnose the issue effectively. References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 5.5.1.

**NEW QUESTION 135**

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 136**

An Incident Management tool implements the following defect states; Open, Assigned, Solved,

Closed Consider the following defect report: Id T000561

Test Object "Warehouse Management" application Tester name; John Bishop

Date: 10th. April 2010 Test Case MRT558I

Status OPEN Severity Serious Priority

Problem- After inputting the Total Quantity item = 450 in the SV034 screen, the system shows an unexpected Error message=47

Correction: Developer name: Closing date:

Which of the following is a valid criticism of this report?

- A. The Priority, the Correction description and the Developer name are missing
- B. The version of the application is missing
- C. There is no link to the applicable requirement (traceability)
- D. The description is not highlighting the source of the problem

**Answer:** B

**Explanation:**

A valid criticism of this report is that the version of the application is missing. The version of the application is an important piece of information that should be included in a defect report, as it helps to identify which release or build of the software product contains the defect. The version of the application can also help to reproduce and debug the defect, as different versions may have different behaviors or features. The other options are not valid criticisms of this report. The priority, the correction description and the developer name are not missing, but rather not applicable for this report. The priority is a measure of how urgently a defect needs to be fixed, which can be assigned by the project manager or the defect tracking system, not by the tester who reports the defect. The correction description and the developer name are information that are added after the defect has been resolved, not when it has been reported. There is no link to the applicable requirement (traceability) is not a valid criticism of this report, because traceability is not a mandatory attribute of a defect report, but rather an optional one. Traceability is a relationship between two or more entities (such as requirements, test cases, defects, etc.) that shows how they are related or dependent on each other. Traceability can help to verify that the requirements are met by the test cases and defects, but it is not essential for reporting a defect. The description is not highlighting the source of the problem is not a valid criticism of this report, because highlighting the source of the problem is not a responsibility of the tester who reports the defect, but rather of the developer who fixes the defect. The description should provide enough information to describe what happened when the defect occurred, such as input values, expected results, actual results, error messages, screenshots, etc., but it does not need to explain why or how it happened. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 140.

**NEW QUESTION 140**

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 145**

Which of the following is an example of tasks most associated with the test design activity?

- A. The project manager updates the project schedule as key test tasks are completed
- B. Test data, derived from production data, is loaded into the test environment
- C. Every day
- D. the tester notes the status of his/her test cases in preparation for daily reports
- E. The identification of test execution and test automation tools

**Answer:** D

**Explanation:**

Test design is one of the fundamental activities in software testing, focusing on how tests will be structured and what strategies will be employed. As per the ISTQB syllabus, identifying appropriate test execution and automation tools is a task closely associated with test design activities because it directly influences how tests will be conducted, managed, and executed efficiently. This activity involves deciding on the tools that will best support the testing goals and requirements. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4 "Test Design Techniques".

**NEW QUESTION 147**

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 151**

A program got 100% decision coverage in a test. Which of the following statements is then guaranteed to be true?

- A. Every executable statement is covered.
- B. Every output equivalence class has been tested.
- C. Every input equivalence class has been tested.
- D. The "dead" code has not been covered.

**Answer:** A

**Explanation:**

If a program got 100% decision coverage in a test, then it is guaranteed that every executable statement is covered. Decision coverage (also known as branch coverage) is a type of structural coverage (also known as white-box coverage) that measures how many decision outcomes have been exercised by a test suite. A decision outcome is a possible result of a decision point (such as an if-then-else statement) in a program's code. Decision coverage requires that each decision point has both true and false outcomes executed at least once by a test suite. Decision coverage implies statement coverage, which is another type of structural coverage that measures how many executable statements have been executed by a test suite. Statement coverage requires that each executable statement is executed at least once by a test suite. Therefore, if a program got 100% decision coverage in a test, then it also got 100% statement coverage in a test, which means that every executable statement is covered. The other options are not guaranteed to be true if a program got 100% decision coverage in a test. Every

output equivalence class has been tested and every input equivalence class has been tested are not guaranteed to be true if a program got 100% decision coverage in a test, because equivalence classes are based on functional requirements or specifications, not on code structure or logic. Equivalence classes are used in specification-based testing (also known as black-box testing), which is a type of testing that does not consider the internal structure or implementation of the system under test. Decision coverage is used in structure-based testing (also known as white-box testing), which is a type of testing that considers the internal structure or implementation of the system under test. Therefore, achieving 100% decision coverage does not imply achieving 100% equivalence class coverage. The “dead” code has not been covered is not guaranteed to be true if a program got 100% decision coverage in a test, because dead code (also known as unreachable code) is code that can never be executed due to logical errors or design flaws. Dead code can reduce readability and maintainability of the code, as well as increase complexity and size. Decision coverage does not account for dead code, as it only considers the decision outcomes that are possible to execute. Therefore, achieving 100% decision coverage does not imply that the dead code has not been covered. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 36.

**NEW QUESTION 156**

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 157**

Which of the following tools is most likely to detect defects in functions or methods in source code?

- A. configuration management tool
- B. unit test framework tool
- C. test design tool
- D. monitoring tool

**Answer:** B

**Explanation:**

A unit test framework tool is a tool that supports the creation, execution, and reporting of unit tests, which are tests that verify the functionality and quality of individual software components (such as functions or methods) in source code. A unit test framework tool can help to detect defects in functions or methods in source code by providing features such as test case generation, test case execution, test result comparison, test coverage measurement, etc. Some examples of unit test framework tools are JUnit, NUnit, TestNG, etc. The other options are not tools that are likely to detect defects in functions or methods in source code. A configuration management tool is a tool that supports the management and control of different versions and variants of software products or components. A test design tool is a tool that supports the design and generation of test cases based on some criteria or rules. A monitoring tool is a tool that monitors the behavior or performance of a system or component under test. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 10.

**NEW QUESTION 161**

.....

## 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>