



Download BCS CTFL4 Exam Dumps Free

Shared by Wilkins on 17-06-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

In which one of the following test techniques are test cases derived from the analysis of the software architecture?

Options:

- A- Black-box test techniques.
- B- Experience-based test techniques.
- C- Checklist-based test techniques.
- D- White-box test techniques.

Answer:

D

Explanation:

White-box test techniques are test design techniques where the test cases are derived from the internal structure of the software, including its architecture, code, and logical flow. These techniques involve the tester having knowledge of the internal workings of the software to create test cases that ensure all possible paths and conditions are tested. This is in contrast to black-box test techniques, which focus on input-output behavior without considering the internal structure.

Reference: ISTQB CTFL Syllabus V4.0, Section 4.3

Question 2

Question Type: MultipleChoice

Atypical generic skill required for the role of tester is the ability to

Options:

- A- Take on the role of developer to meet challenging project deadlines
- B- Assume leadership aimed at imposing decisions on the rest of the team.
- C- Use tools to make the execution of repetitive testing tasks more efficient.
- D- Determine the corrective actions to get a test project on track in case of deviations from the

test plan

Answer:

C

Explanation:

A key skill for testers is the ability to use various tools to automate repetitive tasks, enhancing the efficiency and effectiveness of testing processes. This includes tools for test execution, test management, and defect tracking. The ISTQB CTFL Syllabus v4.0 emphasizes the importance of using tools to improve productivity and reduce manual effort in repetitive testing tasks, making this a critical skill for testers.

Question 3

Question Type: MultipleChoice

Which option best statements best describe Behavior-Driven Development (BDD)?

Options:

- A- A collaborative approach that allows every stakeholder to contribute to how the software component must behave.
- B- Expresses the behavior of an application with test cases written in Given When Then format.
- C- Is used to develop code guided by automated test cases.
- D- A psychological technique in which the team's behavior in agile teams is evaluated.

Answer:

A

Explanation:

Behavior-Driven Development (BDD) is a collaborative approach that enhances communication among project stakeholders, including developers, testers, and business analysts. It involves defining how software should behave through examples written in a common language understandable by all stakeholders, often using the Given-When-Then format.

Question 4

Question Type: MultipleChoice

Which of the following statements about the shift-left approach is FALSE?

Options:

- A- The shift-left approach can only be implemented with test automation
- B- The shift-left approach in testing is compatible with DevOps practices.
- C- The shift-left approach can involve security vulnerabilities
- D- The shift-left approach can be supported by static analysis tools.

Answer:

A

Explanation:

In a formal review process, the recorder's role is typically responsible for documenting the findings of the review team, including action items, decisions, and recommendations. This ensures that there is an accurate record of what was discussed and agreed upon, facilitating follow-up and continuous improvement. Therefore, statement C is correct as per the ISTQB CTFL syllabus.

Question 5

Question Type: MultipleChoice

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

Options:

- A- Tests should be written in executable format before the code is written and should act as executable specifications that drive coding
- B- Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level
- C- Test objectives should be the same for all test levels, although the number of tests designed at

various levels can vary significantly

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

Answer:

D

Explanation:

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

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

Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level: This is a specific testing practice that is associated with sequential software development models, such as the waterfall model, the V-model, or the W-model, in which the software development and testing activities are performed in a linear and sequential order, with well-defined phases, deliverables, and dependencies. Test levels are the stages of testing that correspond to the levels of integration of the software system, such as component testing,

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

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

[ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle1](#)

[ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles1](#)

[ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches1](#)

[ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles1](#)

[ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1](#)

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

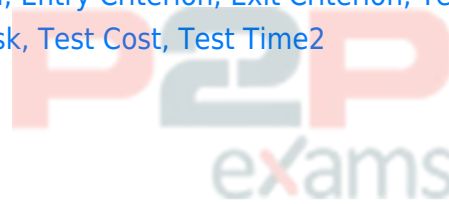
ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design1

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

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

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

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



Question 6

Question Type: MultipleChoice

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?

Options:

- 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. Reference= ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

Question 7

Question Type: MultipleChoice

A company wants to reward each of its salespeople with an annual bonus that represents the sum of all the bonuses accumulated for every single sale made by that salesperson. The bonus for a single sale can take on the following four values: 3%, 5%, 7% and 10% (the percentage refers to the amount of the single sale). These values are determined on the basis of the type of customer (classified as "Basic" or "Premium") to which such sale was made, and on the amount of such sale classified into the following three groups G1, G2 and G3:

- * [G1]: less than 300 euros
- * [G2]: between 300 and 2000 euros
- * [G3]: greater than 2000 euros

Which of the following is the minimum number of test cases needed to cover the full decision table associated with this scenario?

Options:

- A- 12
- B- 6
- C- 4
- D- 3

Answer:

B

Explanation:

The minimum number of test cases needed to cover the full decision table associated with this scenario is 6. This is because the decision table has 4 conditions (type of customer and amount of sale) and 4 actions (bonus percentage). The conditions have 2 possible values each (Basic or Premium, and G1, G2 or G3), so the total number of combinations is $2 \times 2 \times 2 \times 2 = 16$. However, not all combinations are valid, as some of them are contradictory or impossible. For example, a sale cannot be both less than 300 euros and greater than 2000 euros at the same time. Therefore, we need to eliminate the invalid combinations and keep only the valid ones. The valid combinations are:

Type of customer

Amount of sale

Bonus percentage

Basic

G1

3%

Basic

G2

5%

Basic

G3

7%

Premium

G1

5%

Premium

G2

7%

Premium

G3

10%



These 6 combinations cover all the possible values of the conditions and actions, and they are the minimum number of test cases needed to cover the full decision table.

Question 8

Question Type: MultipleChoice

A test status report SHOULD:

Options:

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

Answer:

A

Explanation:

A test status report is a document that provides a snapshot of the testing activities and their progress during a particular period. It should include information about any impediments encountered during the test execution and the actions taken to resolve them, which helps stakeholders understand the challenges and how they were addressed .

Option B describes an activity related to test completion rather than ongoing status reporting. Option C is incorrect because the structure and contents of the report may vary based on the audience's needs. Option D, while important, is not the primary purpose of a test status report, which focuses more on the current status and impediments.



To Get Premium Files for CTFL4 Visit

<https://www.p2pexams.com/products/ctfl4>

For More Free Questions Visit

<https://www.p2pexams.com/bcs/pdf/ctfl4>

20%
DISCOUNT

P2P
exams