

ISTQB

Exam Questions ISTQB-CTFL

ISTQB-Foundation Level Exam



NEW QUESTION 1

Consider the following testing levels:

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

Which of the following statements is true?

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

Answer: D

Explanation:

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

NEW QUESTION 2

Which of the following statements regarding inspection is NOT true?

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

Answer: B

Explanation:

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

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

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

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

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

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

NEW QUESTION 3

Which ONE of the following statements does NOT describe how testing contributes to higher quality?

- A. Properly designed tests that pass reduce the level of risk in a system.
- B. The testing of software demonstrates the absence of defects.
- C. Software testing identifies defects, which can be used to improve development activities.
- D. Performing a review of the requirement specifications before implementing the system can enhance quality.

Answer: B

Explanation:

? The testing of software does not demonstrate the absence of defects, but rather the presence of defects or the conformance of the software to the specified requirements¹. Testing can never prove that the software is defect-free, as it is impossible to test all possible scenarios, inputs, outputs, and behaviors of the software². Testing can only provide a level of confidence in the quality of the software, based on the coverage, effectiveness, and efficiency of the testing activities³.

? The other options are correct because: References =

? 1 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 10

? 2 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 11

? 3 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 12

? 4 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 13

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

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

? 7 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 14

? [8] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 15

? [9] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16

- ? [10] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 17
? [11] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 18
? [12] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 19

NEW QUESTION 4

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

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

Answer: B

Explanation:

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

NEW QUESTION 5

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 value of dynamic 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.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 6

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

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

Answer: A

Explanation:

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

NEW QUESTION 7

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

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

Answer: C

Explanation:

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

NEW QUESTION 8

Which of the following applications will be the MOST suitable for testing by Use Cases

- A. Accuracy and usability of a new Navigation system compared with previous system
- B. A billing system used to calculate monthly charge based on large number of subscribers parameters
- C. The ability of an Anti virus package to detect and quarantine a new threat
- D. Suitability and performance of a Multi media (audio video based) system to a new operating system

Answer: A

Explanation:

A new navigation system compared with a previous system is the most suitable application for testing by use cases, because it involves a high level of interaction between the user and the system, and the expected behavior and outcomes of the system are based on the user's needs and goals. Use cases can help to specify the functional requirements of the new navigation system, such as the ability to enter a destination, select a route, follow the directions, receive alerts, etc. Use cases can also help to compare the accuracy and usability of the new system with the previous system, by defining the success and failure scenarios, the preconditions and postconditions, and the alternative flows of each use case. Use cases can also help to design and execute test cases that cover the main and exceptional paths of each use case, and to verify the satisfaction of the user's expectations.

The other options are not the most suitable applications for testing by use cases, because they do not involve a high level of interaction between the user and the system, or the expected behavior and outcomes of the system are not based on the user's needs and goals. A billing system used to calculate monthly charge based on a large number of subscriber parameters is more suitable for testing by data-driven testing, which is a technique for testing the functionality and performance of a system or component by using a large set of input and output data. The ability of an antivirus package to detect and quarantine a new threat is more suitable for testing by exploratory testing, which is a technique for testing the functionality and security of a system or component by using an informal and flexible approach, based on the tester's experience and intuition. The suitability and performance of a multimedia (audio video based) system to a new operating system is more suitable for testing by compatibility testing, which is a technique for testing the functionality and performance of a system or component by using different hardware, software, or network environments. References = CTFL 4.0 Syllabus, Section 3.1.1, page 28-29; Section 4.1.1, page 44-45; Section 4.2.1, page 47-48.

NEW QUESTION 9

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

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

Answer: A

Explanation:

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

NEW QUESTION 10

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

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

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

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

Answer: A

Explanation:

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

NEW QUESTION 10

Which of 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 15

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

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

Answer: C

Explanation:

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

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

NEW QUESTION 18

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 22

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 25

A test manager defined the following test levels in her test plan; Component, System and Acceptance. Which Software Development lifecycle is the Test Manager most likely following?

- A. V-Model
- B. Agile
- C. Waterfall
- D. Prototyping

Answer: A

Explanation:

The test manager is most likely following the V-model for software development. The V-model is a software development model that defines four testing levels that correspond to four development phases: component testing corresponds to component design, integration testing corresponds to architectural design, system testing corresponds to system requirements specification, and acceptance testing corresponds to user requirements specification. The V-model also defines the test planning and test execution activities for each testing level. Agile is a software development model that follows an iterative and incremental approach, where testing is integrated into each iteration and adapts to changing requirements and feedback. Waterfall is a software development model that follows a sequential and linear approach, where testing is performed after the development phase is completed. Prototyping is a software development model that involves creating a simplified version of the software to elicit user feedback and validate requirements before developing the final product. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 18.

NEW QUESTION 28

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 30

Consider the following code
int premium=2500; if (age<30)
{
premium = premium +1500;
}

Which options suits for a correct combination of Boundary value and expected result. Assume first number as boundary followed by expected result.

- A. * 29, 1500* 30. 2500
- B. * 29. 4000* 30. 2500
- C. * 29, 2500* 30. 1500
- D. * 30, 1500* 31,2500

Answer: B

Explanation:

In the given code snippet, the premium is increased by 1500 if the age is less than 30. Therefore, at the boundary value of age 29, the premium should be 2500 + 1500 = 4000, and at age 30, the premium should remain at its initial value of 2500, as the condition is no longer met. Option B correctly reflects this with 29, 4000 and 30, 2500 as the boundary value and the expected results, respectively.

NEW QUESTION 33

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 37

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 transitions. References:

? Certified Tester Foundation Level v4.0

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

NEW QUESTION 42

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

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

Answer: A

Explanation:

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

NEW QUESTION 47

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

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

Answer: A

Explanation:

Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential bugs¹². However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)³. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures. MTBF can only be estimated by dynamic testing, which involves executing the program under various conditions and collecting data on its failures⁴. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools. The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can be identified by static analysis tools. Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false⁵. Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables⁶. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption. References = What Is Static Analysis? Static Code Analysis Tools - Perforce Software, How Static Code Analysis Works | Perforce, Static Code Analysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs - CareerRide, ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

NEW QUESTION 48

Which of the following statements about reviews are TRUE?

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

Answer: C

Explanation:

The following statements about reviews are true:

- ? I) In walkthroughs the review meeting is typically led by the author. A walkthrough is a type of review that has a predefined objective and agenda but no formal process or roles. A walkthrough is typically led by the author of the work product under review, who guides the participants through a scenario and solicits feedback.
- ? IV) Management rarely participates in technical review meetings. A technical review is a type of review that has a predefined objective and agenda but no formal process or roles. A technical review is typically performed by peers with technical expertise in order to evaluate technical aspects of a work product. Management rarely participates in technical review meetings, as they may not have sufficient technical knowledge or skills to contribute effectively. The following statements about reviews are false:
- ? II) Inspection is characterized by an open-ended review meeting. An inspection is a type of review that follows a defined process with formal entry and exit criteria and roles and responsibilities for participants. An inspection is characterized by a structured review meeting with a fixed duration and agenda.
- ? III) Preparation before the review meeting is part of informal reviews. Preparation before the review meeting is part of formal reviews, such as inspections or technical reviews. Preparation involves checking

NEW QUESTION 50

Which of the following activities are part of test planning?

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

Answer: B

Explanation:

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

NEW QUESTION 51

Which of the following BEST matches the attributes with a level of testing?

- A. Stubs and drivers are often usedII The test environment should correspond to the production environment III Finding defects is not the main focusIV Testing can be based on use casesV Testing is normally performed by testersVI Testing for functional and non-functional characteristics
- B. Component - VI Integration - IV System - I Acceptance - 111
- C. Component - IV Integration - I System - VI Acceptance - V
- D. Component-I Integration - V System - II Acceptance - IV
- E. Component - V Integration - II System - IV Acceptance - VI

Answer: D

Explanation:

The relationship between impact analysis and regression testing in maintenance testing is that 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 53

Which of the following statements about test reports are TRUE?

- II. Test reports shall give stakeholders information as basis for decisions.
- III Test reports shall summarize what happened through a period of testing.
- IV. Test reports shall be approved by the development team, the test team and the customer

- A. Test reports shall include information about remaining risks.
- B. II, III, V
- C. I, II, IV
- D. I, III, v
- E. II, III, IV

Answer: A

Explanation:

Statements II, III and V are true about test reports. Test reports are documents that provide information on the results and status of testing activities for a given period or phase. Test reports should give stakeholders information as basis for decisions, such as whether to release the software product, whether to continue testing, whether to change the scope or priorities of testing, etc. Test reports should summarize what happened through a period of testing, such as what test cases were executed, what defects were found, what risks were identified, what issues were encountered, what achievements were made, etc. Test reports should include information about remaining risks, such as what defects are still open, what test cases are still pending, what functionalities are still untested, what uncertainties are still unresolved, etc. Statements I and IV are not true about test reports. Test reports do not need to be approved by the test team, the development team, or the customer, unless it is specified by the test policy or the test plan. Test reports only need to be reviewed and verified by the test leader or the test manager before being distributed to the intended recipients. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 141.

NEW QUESTION 54

"Statement Testing" is part of;

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

Answer: D

Explanation:

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

NEW QUESTION 57

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 60

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 tested¹. Product risks are related to the requirements, design, implementation, verification, and maintenance of the software product².
- ? 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 product³.
- ? 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 65

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 schedules^{6†}source.
- References:
- ? Certified Tester Foundation Level v4.0
- ? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 68

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 70

Which of the following should be included in a test status report?

- A. Estimation details I
- B. Total number of open and closed defects III Actual effort spent I
- C. Defect reports
- D. Number of executed, failed, blocked tests
- E. III.V
- F. II, III
- G. I
- H. IV
- I. II, III.V

Answer: D

Explanation:

- The following should be included in a test status report: total number of open and closed defects, actual effort spent, and number of executed, failed, and blocked tests.
- A test status report is a document that provides information on the results and status of testing activities for a given period or phase. A test status report should include information that is relevant, accurate, and timely for the intended audience and purpose. Some of the information that should be included in a test status

report are: total number of open and closed defects, which can indicate the defect trend and defect density of the software product; actual effort spent, which can indicate the productivity and efficiency of the testing process; number of executed, failed, and blocked tests, which can indicate the test progress and test coverage of the software product. The following should not be included in a test status report: estimation details, defect reports, and impact analysis. Estimation details are not part of a test status report, but rather part of a test plan or a test estimation document. Estimation details provide information on the expected time, resources, and costs for testing activities, not on the actual results or status of testing activities. Defect reports are not part of a test status report, but rather separate documents that provide detailed information on individual defects found during testing. Defect reports include information such as defect description, defect severity, defect priority, defect status, defect resolution, etc. Defect reports can be referenced or summarized in a test status report, but not included in full. Impact analysis is not part of a test status report, but rather part of a risk assessment or prioritization process. Impact analysis provides information on the potential effects or consequences of a change or a defect on the software product or project. Impact analysis can be used to evaluate the amount or scope of testing to be performed, but not to report the results or status of testing activities. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 141.

NEW QUESTION 73

A bank software has a feature of locking down accounts that have and overdraft higher than \$10,000. Due to a bug in the code, the system locks down accounts once they reached \$1000 overdraft. Based on this context, which of the following is correct statement

- A. It is an example of a defect that it goes undetected, will cause harm to the end customer and to the company (the bank)
- B. It is an example of a defect, that if goes undetected, will cause harm to the end customer but not to the company (the bank)
- C. It is an example of a defect that if goes undetected, will not cause harm to the end customer, but will cause harm to the company (the bank)
- D. It is an example of a defect that if goes undetected, will not cause any real harm to either the end customer or to the company (the bank)

Answer: A

Explanation:

A defect is a flaw in a component or system that can cause the component or system to fail to perform its required function. A defect can cause harm to the end customer, the company, or both, depending on the impact and severity of the failure. In this case, the defect in the code causes the system to lock down accounts that have an overdraft higher than \$1000, instead of \$10,000. This can cause harm to the end customer, as they may lose access to their funds and face inconvenience or financial difficulties. This can also cause harm to the company, as they may lose customer trust and satisfaction, face legal issues or complaints, or incur additional costs for fixing the defect and restoring the accounts. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 3-4.

NEW QUESTION 78

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 80

The following open incident report provided: Date: 01.01.01

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

Severity: High

Life Cycle: Integration

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

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

Answer: B

Explanation:

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

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

NEW QUESTION 83

Which of the following sentences describe a product risk?

- A. The application might not be able to provide the expected responsiveness under a load of up-to 300 concurrent users
- B. Failure in acquiring an adequate and test automation tool
- C. A wrong configuration of the test environment that causes incidents related to the environment and not to the software under test
- D. The development team lacks knowledge of the technology on which the product is based

Answer: A

Explanation:

This question relates to identifying product risks, which are potential problems associated with the product itself, such as software functionality, reliability, usability, and performance. Option A describes a scenario where the application might not meet performance requirements under specific conditions (up to 300 concurrent users), which directly impacts the product's ability to perform its intended function. This is a classic example of a product risk, as it concerns the product's quality and its ability to meet user needs. Options B, C, and D, on the other hand, relate to project risks, which are concerns related to the management and execution of the project, such as tool acquisition, environment configuration, and team expertise, rather than the quality of the product itself.

NEW QUESTION 88

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 92

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 93

Which of the following is a key characteristic of informal reviews?

- A. Kick-off meeting
- B. Low cost
- C. Individual preparation
- D. Metrics analysis

Answer: B

Explanation:

A key characteristic of informal reviews is low cost. Informal reviews are a type of review that does not follow a formal process or have any formal documentation. Informal reviews are usually performed by individuals or small groups of peers or colleagues who have some knowledge or interest in the product under review. Informal reviews can be done at any time and for any purpose, such as checking for errors, clarifying doubts, sharing ideas, etc. Informal reviews have low cost, as they do not require much time, effort, or resources to conduct. The other options are not key characteristics of informal reviews. Kick-off meeting is a characteristic of formal reviews, such as inspections or walkthroughs. Kick-off meeting is a meeting that is held before the review process starts, where the roles and responsibilities of the participants are defined, the objectives and scope of the review are agreed, and the logistics and schedule of the review are planned. Individual preparation is a characteristic of formal reviews, such as inspections or walkthroughs. Individual preparation is an activity that is performed by the reviewers before the review meeting, where they examine the product under review and identify any issues or questions that need to be discussed or resolved during the review meeting. Metrics analysis is a characteristic of formal reviews, such as inspections or walkthroughs. Metrics analysis is an activity that is performed after the review process is completed, where the data and results of the review are collected and analyzed to measure the effectiveness and efficiency of the review, as well as to identify any improvement actions or lessons learned for future reviews. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 9.

NEW QUESTION 95

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 99

A software application incorrectly provided customers discounts of 50% off their total purchases if the purchases exceeded \$100. It was discovered through an audit that the discount should have been only 5% off these purchases. A root cause analysis uncovered that the requirements incorrectly stated 50% instead of 5% in this scenario.

Which of the following MOST accurately reflects this scenario?

- A. The audit finding is the root cause, the incorrect calculation of 50% is the defect, and the incorrect requirement is the effect
- B. The incorrect customer discount is the effect and the reason for the requirement error is the root cause
- C. The incorrect discount is the root cause, requiring a root cause analysis which led to investigating the software code, design, and requirements
- D. A defect in the code is determined to be the root cause of the incorrect calculation

Answer: B

Explanation:

According to the ISTQB CTFL syllabus, a defect is a deviation from the expected result which in this scenario is the incorrect discount applied to the customers. The root cause, as per the ISTQB definition, is the originating cause of a defect, which in this case is the incorrect requirement stating 50% instead of 5%. Therefore, the incorrect requirement is the root cause and the customer receiving the wrong discount is the effect of this root cause. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.3 "Defects, Root Causes, and Effects".

NEW QUESTION 103

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

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

What would be your FIRST recommendation to the company?

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

Answer: B

Explanation:

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

NEW QUESTION 105

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

- A. 999.1000.23232.99999.100000
- B. 999.1000.50000.100000.100001

- C. 999.100000
- D. 1000,50000,99999

Answer: B

Explanation:

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

NEW QUESTION 107

Which of the following BEST describes a test summary report for executive-level employees

- A. The report is detailed and includes a status summary of defects by priority or budget
- B. The report is detailed and includes specific information on defects and trends
- C. The report is high-level and includes a status summary of defects by priority or budget
- D. The report is high-level and includes specific information on defects and trends

Answer: C

Explanation:

For executive-level employees, a test summary report should be concise and focus on high-level information. It typically includes a summary of defects categorized by priority or budget. Executives are generally interested in the overall status and the impact on critical business objectives rather than detailed technical information. The report should provide an overview of the most important aspects of testing, such as key issues, test progress, and any risks or concerns that could affect project outcomes. References:
? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 5.3.1.

NEW QUESTION 108

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 110

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

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

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

Answer: C

Explanation:

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

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

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

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

References:

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

NEW QUESTION 111

You are testing the download process of a mobile phone application.

For which of the following capabilities of the system do you need to design a nonfunctional test?

- A. It was easy to locate, download and install the application
- B. The application was correctly downloaded
- C. The application created an installation log file in a given folder

D. The application installed only after the user's approval

Answer: A

Explanation:

This question asks for a non-functional aspect of testing the download process of a mobile application. Option A, "It was easy to locate, download and install the application," refers to usability, which is a non-functional quality attribute. Non-functional testing involves testing the system's attributes, such as usability, performance, reliability, etc., rather than specific behaviors or functions. Options B, "The application was correctly downloaded," C, "The application created an installation log file in a given folder," and D, "The application installed only after the user's approval," describe functional aspects, focusing on what the software does rather than how it performs or is experienced by the user.

NEW QUESTION 114

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 118

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

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

Answer: D

Explanation:

Equivalence partitioning is a black-box test design technique where inputs to the software or system are divided into groups that are expected to exhibit similar behavior. For an integer field that should accept values from 1 to 15, the valid equivalence class is 1 through 15. The invalid equivalence classes are numbers less than 1 and numbers more than 15. Therefore, option D, "Less than 1, 1 through 15, more than 15," correctly identifies the valid equivalence class along with the two invalid classes, covering all possible input scenarios for the field. Options A, B, and C either do not accurately capture the valid range or incorrectly specify the range boundaries.

NEW QUESTION 123

A company runs a pilot project for evaluation of a test automation tool. Which of the following is NOT a valid object of this pilot project?

- A. Get familiar with the functionality and options of the tool
- B. Check how the tool fits to the existing test processes
- C. Train all testers on using the tool
- D. Decide upon standards for tool implementation

Answer: C

Explanation:

? A pilot project is a small-scale experiment or trial that is conducted to evaluate the feasibility, effectiveness, and suitability of a test automation tool before implementing it on a larger scale1.

? The objectives of a pilot project may vary depending on the context and scope of the test automation initiative, but some common ones are2:

? Therefore, option C is not a valid objective of a pilot project, as it is not necessary to train all testers on using the tool at this stage. Training all testers on using the tool would be more appropriate after the tool has been selected and approved for full-scale implementation, and after the standards and guidelines have been established. Training all testers on using the tool during the pilot project would be inefficient, costly, and premature, as the tool may not be suitable or effective for the intended purpose, or may be replaced by another tool later.

References:

? 1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82

? 2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 83

? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 84

? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 85

NEW QUESTION 126

In which of the following cases you would NOT execute maintenance testing?

- A. Retirement of the software or system
- B. Modifications to a released software or system
- C. Migration of the system data to a replacement system
- D. Update to the Maintainability requirements during the development phase

Answer: D

Explanation:

Maintenance testing is testing performed on a software product after delivery to correct defects or improve performance or other attributes. Maintenance testing can be triggered by various situations, such as modifications to a released software or system, migration of the system data to a replacement system, or retirement of the software or system. Maintenance testing is not executed when there is an update to the maintainability requirements during the development phase, as this is not a maintenance situation but rather a change request that should be handled by the development process. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 18-19.

NEW QUESTION 127

The following incident report that was generated during test of a web application. What would you suggest as the most important report improvement?

Defect detected date: 15 8.2010 Defect detected by: Joe Smith Test level System test

Test case: Area 5/TC 98 Build version: 2011-16.2

Defect description After having filled out all required fields in screen 1, t click ENTER to continue to screen 2 Nothing happens, no system response at all.

- A. Add information about which web browser was used
- B. Add information about which developer should fix the bug
- C. Add the time stamp when the incident happened
- D. Add an impact analysis

Answer: A

Explanation:

The most important report improvement for the given incident report would be to add information about which web browser was used when the defect was detected. This information is relevant for reproducing and debugging the defect, as different web browsers may have different behaviors or compatibility issues with the web application. The other options are less important or irrelevant for the incident report. The developer who should fix the bug can be assigned by the project manager or the defect tracking system, not by the tester who reports the defect. The time stamp when the incident happened is not very useful, as it does not indicate the cause or the frequency of the defect. The impact analysis is not part of the incident report, but rather of the risk assessment or prioritization process. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 140.

NEW QUESTION 130

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

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

Answer: B

Explanation:

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

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

NEW QUESTION 134

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 136

Which of the following are the phases of the ISTQB fundamental test process?

- A. Test planning and control, Test analysis and design, Test implementation and execution, Evaluating exit criteria and reporting
- B. Test closure activities
- C. Test planning, Test analysis and design
- D. Test implementation and control
- E. Checking test coverage and reporting, Test closure activities
- F. Test planning and control, Test specification and design
- G. Test implementation and execution, Evaluating test coverage and reporting, Retesting and regression testing, Test closure activities
- H. Test planning
- I. Test specification and design
- J. Test implementation and execution
- K. Evaluating exit criteria and reporting
- L. Retesting and test closure activities

Answer: A

Explanation:

The ISTQB fundamental test process consists of five main phases, as described in the ISTQB Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15:

? Test planning and control: This phase involves defining the test objectives, scope, strategy, resources, schedule, risks, and metrics, as well as monitoring and controlling the test activities and results throughout the test process.

? Test analysis and design: This phase involves analyzing the test basis (such as requirements, specifications, or user stories) to identify test conditions (such as features, functions, or scenarios) that need to be tested, and designing test cases and test procedures (such as inputs, expected outcomes, and execution steps) to cover the test conditions. This phase also involves evaluating the testability of the test basis and the test items (such as software or system components), and selecting and implementing test techniques (such as equivalence partitioning, boundary value analysis, or state transition testing) to achieve the test objectives and optimize the test coverage and efficiency.

? Test implementation and execution: This phase involves preparing the test environment (such as hardware, software, data, or tools) and testware (such as test cases, test procedures, test data, or test scripts) for test execution, and executing the test procedures or scripts according to the test plan and schedule. This phase also involves logging the outcome of test execution, comparing the actual results with the expected results, and reporting any discrepancies as incidents (such as defects, errors, or failures).

? Evaluating exit criteria and reporting: This phase involves checking if the planned test activities have been completed and the exit criteria (such as quality, coverage, or risk levels) have been met, and reporting the test results and outcomes to the stakeholders. This phase also involves making recommendations for the release or acceptance decision based on the test results and outcomes, and identifying any residual risks (such as known defects or untested areas) that need to be addressed or mitigated.

? Test closure activities: This phase involves finalizing and archiving the testware and test environment for future reuse, and evaluating the test process and the test project against the test objectives and the test plan. This phase also involves identifying any lessons learned and best practices, and communicating the findings and suggestions for improvement to the relevant parties.

References = ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 37-38;

ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 88, page 32.

NEW QUESTION 141

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

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

Answer: B

Explanation:

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

NEW QUESTION 146

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

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

Answer: A

Explanation:

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

NEW QUESTION 147

A Static analysis tool analyzes a given program's CONTROL FLOW among other things. Which of the following options represents the most likely outcome of the control flow analysis:

- A. Identification of unreachable code

- B. Report on adherence to the coding standards
- C. Number of comment lines
- D. Number of source code lines

Answer: A

Explanation:

A static analysis tool is a tool that analyzes a given program's source code or executable code without executing it. A static analysis tool can perform various types of analysis on a program's code, such as syntax checking, data flow analysis, control flow analysis, complexity measurement, coding standards compliance checking, etc. Control flow analysis is a type of analysis that examines how a program's statements are executed in different paths or branches. One of the most likely outcomes of control flow analysis is identification of unreachable code, which is code that can never be executed due to logical errors or design flaws. Unreachable code can reduce readability and maintainability of the code, as well as increase complexity and size. The other options are not outcomes of control flow analysis, but rather outcomes of other types of analysis. Report on adherence to coding standards is an outcome of coding standards compliance checking. Number of comment lines and number of source code lines are outcomes of complexity measurement. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 8.

NEW QUESTION 151

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

```
IF Gender = Boy
  If Age > 3 AND Age < 5 Shoe Size = 1
  ELSE IF Age >=5 AND Age < 7
  Shoe Size = 2 ENDIF
ELSE
  IF Age > 3 AND Age < 5
  Shoe Size = 0
  ELSE IF Age >=5 AND Age < 7
  Shoe Size = 1 ENDIF ENDIF
```

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

Answer: B

Explanation:

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

- ? Gender = Boy, Age <= 3 (Shoe Size assignment is not reached, but the condition is evaluated)
- ? Gender = Boy, Age > 3 AND Age < 5 (Shoe Size = 1)
- ? Gender = Boy, Age >= 5 AND Age < 7 (Shoe Size = 2)
- ? Gender != Boy, Age <= 3 (Again, Shoe Size assignment is not reached, but the condition is evaluated)
- ? Gender != Boy, Age > 3 AND Age < 5 (Shoe Size = 0)
- ? Gender != Boy, Age >= 5 AND Age < 7 (Shoe Size = 1)

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

NEW QUESTION 155

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 159

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 163

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

Conditions

Preferred Guest Card holder	YES	YES	NO	NO
Room Type	Silver	Platinum	Silver	Platinum

Conditions	YES	NO	NO	NO
48 Preferred upgrade to Gold Luxury	YES	NO	NO	NO
Preferred upgrade to Silver	N/A	YES	N/A	NO

What is the expected result for each of the following test cases?

Customer A: Preference Guest Card holder, holding a Silver room Customer B: Non Preferred Guest Card holder, holding a Platinum room

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

Answer: C

Explanation:

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

NEW QUESTION 168

Which of the following is NOT a product risk?

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

Answer: C

Explanation:

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

NEW QUESTION 169

Which of the following statements is LEAST likely to be describing component testing?

- A. It identifies defects in modules and classes.
- B. Simulators and stubs may be required.
- C. It mainly tests interfaces and interaction between components.
- D. It may be applied using a test-first approach.

Answer: C

Explanation:

Component testing (also known as unit testing or module testing) is a level of testing that focuses on verifying the functionality and quality of individual software components (such as modules, classes, functions, methods, etc.). Component testing mainly tests interfaces and interaction between components, as well as internal logic and data structures of the components. Component testing may be applied using a test-first approach (such as test-driven development or behavior-driven development), where tests are written before the code is implemented. Component testing does not identify defects in modules and classes, as this is a result of component testing, not an objective. Simulators and stubs may be required for component testing, as they can simulate or replace missing or incomplete components or external systems that are needed for testing. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 19.

NEW QUESTION 170

Can "cost" be regarded as Exit criteria?

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

Answer: A

Explanation:

Cost can be regarded as an exit criterion for testing, because it is a factor that affects the profitability and feasibility of the software product. Testing is an investment that aims to improve the quality and reliability of the software product, but it also consumes resources, such as time, money, and human effort. Therefore, testing should be planned and executed in a way that balances the cost and benefit of testing activities. Having cost as an exit criterion helps to avoid spending too much money on testing, which may result in an unprofitable product or a loss of competitive advantage. Cost can also help to prioritize and focus the testing efforts on the most critical and valuable features and functions of the software product. However, cost should not be the only exit criterion for testing, as it may not reflect the true quality and risk level of the software product. Other exit criteria, such as defect rate, test coverage, user satisfaction, etc., should also be considered and defined in the test plan.

The other options are incorrect, because they either deny the importance of cost as an exit criterion, or they make false or unrealistic assumptions about the cost of testing. Option B is incorrect, because the financial value of product quality can be estimated, for example, by using cost-benefit analysis, return on investment, or cost of quality models. Option C is incorrect, because going by cost as an exit criterion does not necessarily constrain the testing project or help achieve the desired quality level. Cost is a relative and variable factor that depends on the scope, complexity, and context of the software product and the testing project. Option D is incorrect, because the cost of testing can be measured effectively, for example, by using metrics, such as test effort, test resources, test tools, test environment, etc.

NEW QUESTION 171

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 172

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 173

Which of the following statements about independent testing is WRONG?

- A. Independent testing is necessary because developers don't know any testing.
- B. Independent testing is best suited for the system test level.

- C. A certain degree of independence makes the tester more effective at finding defects.
- D. Independent test teams may find other types of defects than developers who are familiar with the system's structure.

Answer: A

Explanation:

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

NEW QUESTION 175

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 176

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 181

Your manager asked you when testing will be complete. In order to answer this question, you'll most likely use:

- A. Test progress reports
- B. Your colleagues advice
- C. A conversion spreadsheet
- D. A Test Oracle

Answer: A

Explanation:

When a manager asks when testing will be complete, the most appropriate and informative resource to provide an answer is test progress reports (Option A). Test progress reports contain detailed information on the status of testing activities, including what has been accomplished, what remains to be done, the results of the tests conducted, and any issues or risks that might impact the completion of testing. These reports allow for an informed assessment of the testing progress and estimation of when testing might be completed. Options B, C, and D do not provide the structured, detailed, and specific information required to accurately answer

the manager's question about the completion of testing.

NEW QUESTION 185

The ISTQB glossary defines Quality Assurance as: "Part of quality management focused on providing confidence that quality requirements will be fulfilled. Which of the following is not one of the Quality Assurance activity?"

- A. Requirements elicitation
- B. Defect analysis
- C. Functional Testing
- D. Performance Testing

Answer: C

Explanation:

Quality Assurance (QA) activities are focused on providing confidence that quality requirements will be fulfilled through planned and systematic processes. These activities are preventive in nature, aimed at ensuring quality is built into the product from the beginning.

? Requirements elicitation (A) is part of the requirements engineering process and is concerned with gathering the needs and conditions to meet for a new or altered product.

? Defect analysis (B) can be part of QA activities as it involves analyzing defects to prevent them in future development cycles.

? Functional Testing (C) and Performance Testing (D) are types of dynamic testing, which are actually Quality Control activities rather than Quality Assurance.

They are concerned with the identification of defects in the product, not with the processes to prevent defects.

Since the question asks for an activity that is NOT part of Quality Assurance, options A and B are incorrect because they can be part of QA activities. Between C and D, while both are dynamic testing activities, Functional Testing (C) is more directly related to verifying the functionality against specified requirements, which is more aligned with Quality Control. Therefore, C is the best answer.

NEW QUESTION 190

Which of the following would be the LEAST likely to be used as the basis for a test exit criteria?

- A. Test schedules
- B. Cost of testing performed so far
- C. Confidence of testers in tested code
- D. Number of unfixed defects

Answer: A

Explanation:

Test exit criteria are the conditions or requirements that must be met before testing can be concluded. Test exit criteria are usually defined in the test plan and agreed by the stakeholders. Test exit criteria can be based on various factors, such as test coverage, defect status, quality level, risk level, etc. Test schedules would be the least likely to be used as the basis for test exit criteria, because test schedules are not directly related to the quality or performance of the software product, but rather to the time or resources allocated for testing. Test schedules can be used as the basis for test entry criteria, which are the conditions or requirements that must be met before testing can start. The other options are more likely to be used as the basis for test exit criteria. Cost of testing performed so far can be used as a basis for test exit criteria, because it can indicate the return on investment or the cost-benefit ratio of testing. Confidence of testers in tested code can be used as a basis for test exit criteria, because it can reflect the level of satisfaction or assurance of the testers about the quality or reliability of the software product. Number of unfixed defects can be used as a basis for test exit criteria, because it can indicate the level of risk or impact of the remaining defects on the software product. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 13.

NEW QUESTION 195

You need to test the login page of a web site. The page contains fields for user name and password. Which test design techniques are most appropriate for this case?

- A. Decision table testing, state transition testing.
- B. Equivalence partitioning, Boundary value analysis.
- C. Exploratory testing, statement coverage.
- D. Decision coverage, fault attack.

Answer: B

Explanation:

Equivalence partitioning and boundary value analysis are test design techniques that are most appropriate for testing the login page of a web site. The page contains fields for user name and password, which are input values that can be divided into partitions of equivalent data. 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. Boundary value analysis is a technique that tests boundary values between partitions of equivalent data. Boundary values are values at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Boundary value analysis can be used to detect defects caused by incorrect handling of boundary conditions. For example, for testing the user name field, we can identify two equivalence partitions: valid user name (existing and correct) and invalid user name (non-existing or incorrect). The boundary values for these partitions are the minimum and maximum length of user name allowed by the system.

Decision table testing and state transition testing are not suitable for testing the login page of a web site, as they are more applicable for testing components that have multiple inputs and outputs that depend on logical combinations of conditions or events. Decision table testing is a technique that shows combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects). State transition testing is a technique that models how a system transitions from one state to another depending on events or conditions.

Exploratory testing and statement coverage are not suitable for testing the login page of a web site, as they are more applicable for testing components that require learning, creativity and intuition or structural analysis. 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. Statement coverage is a type of structural testing that measures how many statements in a program have been executed by a test suite. Statement coverage can be used to assess the adequacy or completeness of a test suite.

Decision coverage and fault attack are not suitable for testing the login page of a web site, as they are more applicable for testing components that have complex logic or potential errors. Decision coverage is a type of structural testing that measures how many decision outcomes in a program have been executed by a test suite. Decision coverage can be used to assess the adequacy or completeness of a test suite. Fault attack is a type of functional testing that deliberately introduces faults into a system in order to provoke failures or errors. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus -

Springer], Chapter 4, page 34-46; Chapter 5, page 47-48.

NEW QUESTION 198

Why it is essential that defects found in a review be reported objectively?

- A. In order to facilitate easy entry of detected defects in a OTS (Defect Tracking System)
- B. In order to allow the author of reviewed work product(S) to take the feedback positively as an effort at improving the product (S) and not as a personal assault
- C. In order to allow the review moderator to easily understand them, and assign them to the right developer for fixing
- D. In order to allow augmentation of existing checklists used for reviewing the work product (S)

Answer: B

Explanation:

The purpose of a review is to find defects and improve the quality of the work product, not to criticize or blame the author. Reporting defects objectively means describing them factually and constructively, without using negative or emotional language that could offend the author or damage their motivation. This way, the author can take the feedback positively as an effort at improving the product and not as a personal assault. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 138.

NEW QUESTION 202

You are responsible for applying the correct technique for a review of the requirements document for a project to develop a new software application. You identify the reviewers and the required roles, including the meeting leader, who is the requirements document author, and a separate role for a scribe. Additionally, you decide to take a relatively informal approach to the requirements review. The goal of the review is to find defects in the requirements document, such as omissions, Inconsistencies, and duplications. Another goal of the review is to improve the software application's usability and accessibility by considering the various stakeholders' viewpoints.

Which of the following statements BEST describes this scenario?

- A. This scenario is using a pair review type and a perspective-based review technique
- B. This scenario is using a walkthrough review type and a checklist-based review technique
- C. This scenario is using a walkthrough review type and a perspective-based review technique
- D. This scenario is using a pair review type and a checklist-based review technique

Answer: C

Explanation:

This scenario is using a walkthrough review type and a perspective-based review technique. In a walkthrough, the author of the document leads the meeting and it typically includes a meeting leader and a scribe, as described. This type of review is informal, focuses on discussion, and often involves scenario-based reading of the document to understand different user perspectives (ISTQB Main Web).References:
 ? ISTQB® Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION 205

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

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

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

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

Answer: D

Explanation:

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

- ? Valid ZIP code format and valid ZIP code value (e.g., 12345)
- ? Valid ZIP code format and invalid ZIP code value (e.g., 99999)
- ? Invalid ZIP code format and valid ZIP code value (e.g., 1234)
- ? Invalid ZIP code format and invalid ZIP code value (e.g., ABCDE) Therefore, four test cases are required to test this field, one for each equivalence class.

Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION 209

A QA manager of a start-up company needs to implement within a week a low cost incident management tool. Which of the following is the best option?

- A. Document incidents on a large board in the lab
- B. Purchase and deploy an incident management tool
- C. Manage the incidents through E-mails and phone calls
- D. Manage the incidents in a spreadsheet posted on the intranet

Answer: D

Explanation:

An incident is any event that occurs during testing that requires investigation. An incident management tool is a software tool that supports recording and tracking incidents throughout their life cycle. A QA manager of a start-up company needs to implement within a week a low cost incident management tool. The best option for this case is to manage the incidents in a spreadsheet posted on the intranet. This option has several advantages over other options:

- ? It is low cost, as it does not require purchasing any additional software or hardware.
- ? It is easy to implement within a week, as it does not require installing or configuring any complex software or hardware.

? It is accessible and transparent, as it can be viewed and updated by anyone who has access to the intranet.

? It is structured and organized, as it can store and display various information about incidents, such as identifier, summary, description, severity, priority, status, resolution, etc. The other options are not suitable for this case, as they have several disadvantages over the chosen option:

? Documenting incidents on a large board in the lab is not a good option, as it is not accessible or transparent to anyone who is not physically present in the lab. It is also not structured or organized, as it may not store or display all the necessary information about incidents.

? Purchasing and deploying an incident management tool is not a good option, as it is not low cost or easy to implement within a week. It may require spending a significant amount of money and time on acquiring, installing and configuring the software or hardware.

? Managing the incidents through emails and phone calls is not a good option, as it is not structured or organized. It may lead to confusion, inconsistency or loss of information about incidents. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33.

NEW QUESTION 210

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questions and Answers in PDF Format

ISTQB-CTFL Practice Exam Features:

- * ISTQB-CTFL Questions and Answers Updated Frequently
- * ISTQB-CTFL Practice Questions Verified by Expert Senior Certified Staff
- * ISTQB-CTFL Most Realistic Questions that Guarantee you a Pass on Your First Try
- * ISTQB-CTFL Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The ISTQB-CTFL Practice Test Here](#)