

ISTQB

Exam Questions ISTQB-CTFL

ISTQB-Foundation Level Exam



NEW QUESTION 1

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

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

Answer: B

Explanation:

A correct reason to apply test automation is when there are a lot of repetitive testing tasks. Test automation is the use of software tools or scripts to perform or support testing activities, such as test case execution, test result comparison, test data generation, etc. Test automation can be beneficial when there are a lot of repetitive testing tasks that need to be performed frequently or consistently, such as regression testing, performance testing, load testing, etc. Test automation can help to save time and effort, increase reliability and accuracy, and improve coverage and efficiency of testing. The other options are not correct reasons to apply test automation. When a new test automation tool is launched is not a reason to apply test automation, but rather a factor for choosing a test automation tool. When it is easy to automate is not a reason to apply test automation, but rather a factor for evaluating the feasibility of test automation. When it is cheap to buy test automation tools is not a reason to apply test automation, but rather a factor for estimating the cost and benefit of test automation. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 10.

NEW QUESTION 2

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

- The bank card is valid
- The PIN code is correct
- Money is available in the user's account

The following are some possible interactions between the user and the ATM:

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

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

Answer: B

Explanation:

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

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

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

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

NEW QUESTION 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

Given the following review types and review characteristics:

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

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

Answer: C

Explanation:

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

NEW QUESTION 5

Which of the following BEST describes a benefit of test automation?

- A. More subjective assessment
- B. Reduction in repetitive manual work
- C. Availability of the test automation tool vendor
- D. Negligible effort to maintain the test assets generated by the tool

Answer: B

Explanation:

Test automation provides numerous benefits to software testing, and one of the key advantages is the reduction of repetitive manual work. This benefit is explicitly covered in the ISTQB Foundation Level Syllabus (v4.0).

Test automation allows testers to automate repetitive tasks such as regression testing, freeing up their time to focus on more complex and exploratory testing. This leads to improved efficiency and helps in avoiding human errors associated with repetitive tasks. Option A: "More subjective assessment" contradicts the benefit of automation as it focuses on objectivity.

Option C: "Availability of the test automation tool vendor" is not a direct benefit of test automation, although vendor support can be valuable.

Option D: "Negligible effort to maintain the test assets" is misleading as maintaining automated tests often requires effort and attention to changes in the system under test. Therefore, the correct answer is B (ISTQB not-for-profit association) (ISTQB). References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 6

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

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

Answer: C

Explanation:

The Waterfall model exemplifies a software development model that does not support the principle of early testing. In the Waterfall model, each phase must be completed before the next begins, which delays testing until after the completion of the earlier phases like requirements gathering and design. This can often result in finding defects later in the development cycle, making them more expensive and time-consuming to fix (ISTQB not-for-profit association) (ISTQB not-for-profit association). References:

? ISTQB® Certified Tester Foundation Level Syllabus v4.0: https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf
? ISTQB News Release on CTFL v4.0: <https://www.istqb.org/news/posts/istqb-releases-certified-tester-foundation-level-v40-ctfl/>

NEW QUESTION 7

Which of the following is a typical product risk?

- A. Poor usability of the software
- B. A problem in the code developed by a 3rd party
- C. Low quality of the configuration data, test data and tests
- D. Problem in defining the right requirements

Answer: A

Explanation:

A typical product risk involves issues directly related to the software product's functionality, performance, usability, reliability, etc. Option A, "Poor usability of the software," directly impacts the end-user's interaction with the software and is a quality attribute of the product itself, making it a product risk. Options B, "A problem in the code developed by a 3rd party," C, "Low quality of the configuration data, test data and tests," and D, "Problem in defining the right requirements," can be considered either product or project risks depending on the context, but option A is the most directly associated with a typical product risk concerning the quality and usability of the software.

NEW QUESTION 8

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

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

Answer: D

Explanation:

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

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

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

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

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 9

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 10

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^{6†source9†source}. References:

? Certified Tester Foundation Level v4.0

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

NEW QUESTION 10

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 11

A software module to be used in a mission critical application incorporates an algorithm for secure transmission of data. Which review type is most appropriate to ensure high quality and technical correctness of the algorithm?

- A. Walkthrough
- B. Informal Review
- C. Technical Review
- D. Management Review

Answer: C

Explanation:

A technical review is a type of formal review that involves a team of technical experts who evaluate a software product against a set of predefined quality criteria. A technical review is suitable for ensuring high quality and technical correctness of complex or critical software components, such as algorithms, architectures or designs. A technical review is not a walkthrough, which is an informal review led by the author of the work product. A technical review is not an informal review, which is a review that does not follow a defined process and has no formal entry or exit criteria. A technical review is not a management review, which is a type of formal review that focuses on business aspects and project progress. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 3, page 29-30.

NEW QUESTION 14

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 16

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

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

Answer: D

Explanation:

Test exit criteria are the conditions that must be fulfilled before concluding a particular testing phase. These criteria act as a checkpoint to assess whether we have achieved the testing objectives and are done with testing¹. Test exit criteria are typically defined in the test plan document, which is one of the outputs of the test planning phase. The test plan document describes the scope, approach, resources, and schedule of the testing activities. It also identifies the test items, the features to be tested, the testing tasks, the risks, and the test deliverables². According to the ISTQB® Certified Tester Foundation Level Syllabus v4.0, the test plan document should include the following information related to the test exit criteria³:

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

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

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

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

Therefore, the test plan document is the most appropriate test document to find the test exit criteria described. The other options, such as test design specification, project plan, and requirements specification, are not directly related to the test exit criteria. The test design specification describes the test cases and test procedures for a specific test level or test type³. The project plan describes the overall objectives, scope, assumptions, risks, and deliverables of the software project⁴. The requirements specification describes the functional and non-functional requirements of the software product⁵. None of these documents specify the conditions for ending the testing process or evaluating the testing

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

NEW QUESTION 21

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

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

Answer: C

Explanation:

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

NEW QUESTION 26

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 29

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

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

Answer: A

Explanation:

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

References:

? ISTQB CTFL Syllabus v4.0: ISTQB Official Website

? ISTQB Foundation Level Resources v4.0: ASTQB Resources

NEW QUESTION 30

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 35

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

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

Answer: C

Explanation:

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

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

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

NEW QUESTION 36

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 andtimely 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 40

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 43

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



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

Answer: D

Explanation:

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0, the life cycle of a defect typically follows a sequence from its discovery to its closure. In the provided figure, it starts with S0 (New), moves to S1 (Assigned), then to S2 (Resolved), followed by S3 (Verified). If the defect is not fixed, it can be Re-opened (S5) and goes back for verification (S3). Once verified, it is Closed (S4). References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.4.3, Page 17.

NEW QUESTION 46

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 49

Which of the following statements about static analysis are FALSE?

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

Answer: D

Explanation:

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

NEW QUESTION 54

A software application incorrectly provided customers discounts of 50% off their total purchases if the purchases exceeded S100. 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 59

Which of the following is NOT a deciding factor m determining the extent of testing required?

- A. Level of risk of the product or features
- B. Budget to do testing
- C. A particular tester involved in testing
- D. Time available to do testing

Answer: C

Explanation:

The extent of testing required for a software product depends on various factors, such as the level of risk, the budget, and the time available. The level of risk reflects the potential impact of failures on the stakeholders and the environment. The budget determines how much resources can be allocated for testing. The time available defines the schedule and deadlines for testing activities. The particular tester involved in testing is not a deciding factor for the extent of testing required, as testing should be based on objective criteria and not on personal preferences or abilities. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 14-15.

NEW QUESTION 61

A mid-size software product development company has analyzed data related to defects detected in its product and found out that detects 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, andmore tests can be executed m 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, tram 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 64

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 execution6†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 65

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

For which to the following capabilities to the system 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 70

Decision table testing is being performed on tran-sactions 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 71

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

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

Answer: C

Explanation:

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

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

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

NEW QUESTION 76

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

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

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

Answer: B

Explanation:

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

the time limit. It is related to the quality of the test code. References = ISTQB® Certified Tester Foundation Level Syllabus v4.0, Requirements Engineering Fundamentals.

NEW QUESTION 81

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 86

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

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

Answer: C

Explanation:

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

least once (ISTQB Main Web).References:

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

NEW QUESTION 89

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 93

Which of the following statements describes regression testing?

- A. Retesting of a fixed defectI
- B. Testing of an already tested programII
- C. Testing of new functionality in a programI
- D. Regression testing applies only to functional testingV Tests that do not have to be repeatable, because They are only used once
- E. II, IV, V
- F. I, III, IV
- G. II
- H. I, IV

Answer: C

Explanation:

Regression testing is the re-running of functional and non-functional tests to ensure that previously developed and tested software still performs as expected after a change1 It does not involve retesting of a fixed defect, testing of new functionality, or applying only to functional testing. Tests that are used for regression testing should be repeatable, because they are used to verify the stability of the software after each change2 References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 4, Section 4.2.2, Page 291; ISTQB Glossary of Testing Terms v4.0, Page 292

NEW QUESTION 94

Can "cost" be regarded as Exit criteria?

- A. Ye
- B. Spending too much money on testing 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 help 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 95

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 97

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 98

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 100

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

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

Answer: C

Explanation:

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

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

NEW QUESTION 102

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 104

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 108

Which of the following is correct with regards to debugging?

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

Answer: A

Explanation:

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

NEW QUESTION 110

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

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

Answer: D

Explanation:

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

NEW QUESTION 114

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

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

Answer: B

Explanation:

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

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

NEW QUESTION 119

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

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

Answer: B

Explanation:

Structural testing is a type of testing that measures its effectiveness by tracking which lines of code were executed by the tests. Structural testing, also known as white-box testing or glass-box testing, is based on the internal structure, design, or implementation of the software. Structural testing aims to verify that the software meets the specified quality attributes, such as performance, security, reliability, or maintainability, by exercising the code paths, branches, statements, conditions, or data flows. Structural testing uses various coverage metrics, such as function coverage, line coverage, branch coverage, or statement coverage, to determine how much of the code has been tested and to identify any untested or unreachable parts of the code. Structural testing can be applied at any level of testing, such as unit testing, integration testing, system testing, or acceptance testing, but it is more commonly used at lower levels, where the testers have access to the source code.

The other options are not correct because they are not types of testing that measure their effectiveness by tracking which lines of code were executed by the tests.

Acceptance testing is a type of testing that verifies that the software meets the acceptance criteria and the user requirements. Acceptance testing is usually performed by the end-users or customers, who may not have access to the source code or the technical details of the software. Acceptance testing is more concerned with the functionality, usability, or suitability of the software, rather than its internal structure or implementation. Integration testing is a type of testing that verifies that the software components or subsystems work together as expected. Integration testing is usually performed by the developers or testers, who may use both structural and functional testing techniques to check the interfaces, interactions, or dependencies between the components or subsystems. Integration testing is more concerned with the integration logic, data flow, or communication of the software, rather than its individual lines of code. Exploratory testing is a type of testing that involves simultaneous learning, test design, and test execution. Exploratory testing is usually performed by the testers, who use their creativity, intuition, or experience to explore the software and discover any defects, risks, or opportunities for improvement. Exploratory testing is more concerned with the behavior, quality, or value of the software, rather than its internal structure or implementation. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54; Chapter 1: Fundamentals of Testing, Section 1.4: Testing Throughout the Software Development Lifecycle, Pages 11-13; Chapter 3: Static Testing, Section 3.4: Exploratory Testing, Pages 40-41.

NEW QUESTION 122

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

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

Answer: A

Explanation:

If a program got 100% decision coverage in a test, then it is guaranteed that every executable statement is covered. Decision coverage (also known as branch coverage) is a type of structural coverage (also known as white-box coverage) that measures how many decision outcomes have been exercised by a test suite. A decision outcome is a possible result of a decision point (such as an if-then-else statement) in a program's code. Decision coverage requires that each decision point has both true and false outcomes executed at least once by a test suite. Decision coverage implies statement coverage, which is another type of structural coverage that measures how many executable statements have been executed by a test suite. Statement coverage requires that each executable statement is

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

NEW QUESTION 124

A bank offers a savings account with various interest rates based on the current balance in the account. The balance ranges and respective interest rates are:

Up to \$100.00 = 2%

\$100.01 to \$500.00 = 4%

\$500.01 to \$1,000.00 = 5% Above \$1,000.00 = 7%

Using two-point boundary value analysis, which of the following sets of test inputs provides the relatively highest level of boundary coverage?

- A. \$5.00, \$100.00, \$499.99, \$1,000.00, \$1,000.01
- B. \$100.00, \$100.01, \$100.02, \$500.00, \$999.99
- C. \$100.00, \$500.00, \$1,000.00, \$1,000.01
- D. \$5.00, \$100.00, \$500.00, \$1,000.01

Answer: B

Explanation:

Boundary Value Analysis (BVA) is a software testing technique in which tests are designed to include values at the boundaries. The concept is to focus on the boundaries since errors tend to occur at the edges of input ranges rather than in the middle.

Given the problem statement:

? Up to \$100.00 = 2%

? \$100.01 to \$500.00 = 4%

? \$500.01 to \$1,000.00 = 5%

? Above \$1,000.00 = 7%

Two-point boundary value analysis means testing the two boundaries of each range. For each range:

? The boundaries for "Up to \$100.00" would be \$100.00 and \$100.01.

? The boundaries for "\$100.01 to \$500.00" would be \$100.00 and \$500.00.

? The boundaries for "\$500.01 to \$1,000.00" would be \$500.00 and \$1,000.00.

? The boundaries for "Above \$1,000.00" would be \$1,000.00 and \$1,000.01. Now, let's examine the options:

? A. \$5.00, \$100.00, \$499.99, \$1,000.00, \$1,000.01

? B. \$100.00, \$100.01, \$100.02, \$500.00, \$999.99

? C. \$100.00, \$500.00, \$1,000.00, \$1,000.01

? D. \$5.00, \$100.00, \$500.00, \$1,000.01

Given the options, B provides the highest boundary coverage (ISTQB not-for-profit association) (Udemy).

References:

? Certified Tester Foundation Level v4.0

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

NEW QUESTION 126

Which of the following statements is CORRECT?

- A. Test cases are made up of input values, expected results and actual results developed to cover test objectives
- B. Test cases describe items or events to test that are derived from the test basis during the test analysis activity
- C. Test cases are sequences of actions for test execution specified during the test implementation activity
- D. Test cases are derived during the test design activity to cover test objectives or test conditions

Answer: C

Explanation:

A test case is a set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition. A test case is a sequence of actions for test execution that can be followed by a tester or a test automation tool. A test case is specified during the test implementation activity, which is the activity that prepares the testware needed for test execution. A test case does not include actual results, as these are obtained during test execution and compared with the expected results. A test case does not describe items or events to test, as these are derived from the test basis during the test analysis activity. A test case is not derived during the test design activity, as this is the activity that specifies the test conditions or objectives that need to be tested. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 3, page 23-24; Chapter 4, page 34.

NEW QUESTION 131

Which of the following are valid testing principles?

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

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

Answer: A

Explanation:

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

NEW QUESTION 133

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

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

Answer: A

Explanation:

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

? Skill and staff shortages: This factor refers to the lack of adequate or qualified human resources to perform the project tasks. This may result in delays, errors, rework, or low productivity.

? Problems in defining the right requirements: This factor refers to the difficulties or ambiguities in eliciting, analyzing, specifying, validating, or managing the requirements of the project. This may result in misalignment, inconsistencies, gaps, or changes in the requirements, affecting the project scope and quality.

? Contractual issues: This factor refers to the challenges or disputes that may arise from the contractual agreements between the project parties, such as clients, suppliers, vendors, or subcontractors. This may result in legal, financial, or ethical risks, affecting the project delivery and satisfaction.

The other options are not correct because they list factors that contribute to PRODUCT risks, not project risks. Product risks are the uncertainties or threats that may affect the quality or functionality of the software product or system. Some of the factors that contribute to product risks are:

? Poor software quality characteristics: This factor refers to the lack of adherence or compliance to the quality attributes or criteria of the software product or system, such as reliability, usability, security, performance, or maintainability. This may result in defects, failures, or dissatisfaction of the users or stakeholders.

? Software does not perform its intended functions: This factor refers to the deviation or discrepancy between the expected and actual behavior or output of the software product or system. This may result in errors, faults, or malfunctions of the software product or system.

References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 1: Fundamentals of Testing, Section 1.5: Risks and Testing, Pages 14-16.

NEW QUESTION 137

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

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

Answer: D

Explanation:

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

NEW QUESTION 139

What is test oracle?

- A. The source of test objectives
- B. The source for the actual results
- C. The source of expected results
- D. The source of input conditions

Answer: C

Explanation:

A test oracle is a mechanism or principle that can be used to determine whether the observed behavior or output of a system under test is correct or not¹. A test oracle can be based on various sources of expected results, such as specifications, user expectations, previous versions, comparable systems, etc². References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.2.1, Page 91; ISTQB Glossary of Testing Terms, Version 4.0, Page 332.

NEW QUESTION 141

.....

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](#)