

Exam Questions ISTQB-CTFL

ISTQB-Foundation Level Exam

<https://www.2passeasy.com/dumps/ISTQB-CTFL/>



NEW QUESTION 1

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

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

Answer: D

Explanation:

The test leader is the person who is responsible for planning, monitoring, and controlling the test activities and resources in a test project. The test leader should have the best knowledge of the test objectives, scope, risks, resources, schedule, and quality criteria. The test leader should also be aware of the test automation criteria, such as the execution frequency, the test support, the team education, the roles and responsibilities, and the devs and testers collaboration¹. Based on these factors, the test leader can decide which tests are suitable for automation and which are not, and prioritize them accordingly. The test leader can also coordinate with the test automation engineers, the developers, and the stakeholders to ensure the alignment of the test automation strategy with the test project goals and expectations. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 2, Section 2.3.1, Page 152; ISTQB Glossary of Testing Terms v4.0, Page 403; ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 6, Section 6.1.1, Page 514; Top 8 Test Automation Criteria You Need To Fulfill - QAMIND¹

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

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

Answer: C

Explanation:

In the context of agile (iterative-incremental) development models, testing is integrated into the development process and occurs continuously throughout the

lifecycle of the project. Agile testing emphasizes adaptability and the need for feedback at various stages of development.

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

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

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

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

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

NEW QUESTION 4

Manager responsibilities in formal review includes ad except one of the following:

- A. Planning the review
- B. Determines if the review objectives have been met
- C. Decide on the execution of reviews
- D. Allocate time for review

Answer: B

Explanation:

A formal review is a type of review that follows a defined process with formal entry and exit criteria and roles and responsibilities for participants. A formal review can have various roles involved, such as manager, moderator, author, reviewer and scribe. The manager responsibilities in formal review include all except one of the following:

? Planning the review (correct responsibility)

? Determines if the review objectives have been met (incorrect responsibility)

? Decide on the execution of reviews (correct responsibility)

? Allocate time for review (correct responsibility) The responsibility of determining if the review objectives have been met belongs to the moderator role, not to the manager role. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 3, page 28-29.

NEW QUESTION 5

The four test levels used in ISTQB syllabus are:

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

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

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

Answer: D

Explanation:

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

The other options are incorrect because:

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

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

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

and likelihood of poor product quality.

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

NEW QUESTION 6

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

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

Answer: D

Explanation:

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

NEW QUESTION 7

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 8

Which of the following coverage criteria results in the highest coverage for state transition based test cases?

- A. Can't be determined
- B. Covering all transitions at least once
- C. Covering only start and end states
- D. Covering all states at least once

Answer: B

Explanation:

Covering all transitions at least once is the highest coverage criterion for state transition based test cases, because it ensures that every possible change of state is tested at least once. This means that all the events that trigger the transitions, as well as the actions and outputs that result from the transitions, are verified. Covering all transitions at least once also implies covering all states at least once, but not vice versa. Therefore, option D is not the highest coverage criterion. Option C is the lowest coverage criterion, because it only tests the initial and final states of the system or component, without checking the intermediate states or transitions. Option A is incorrect, because the coverage criteria for state transition based test cases can be determined and compared based on the number of transitions and states covered. References = CTFL 4.0 Syllabus, Section 4.2.3, page 49-50.

NEW QUESTION 9

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 10

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 10

A test score indicator for students produces a performance score based on a combination of the number of consecutive hours studied (below 4 hours, 4 to 8 hours, 9 to 12 hours or above 12 hours) and the average intensity of focus on the material during the study time (low, medium or high).

Given the following test cases: hours intensity score

T1 3 low55

T2 14 high 95

T3 9 low75

What is the minimum number of additional test cases that are needed to ensure full coverage of all valid INPUT equivalence partitions?

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

Answer: C

Explanation:

Considering the various valid input equivalence partitions of hours studied and intensity, three additional test cases are needed to fully cover all valid partitions. This would typically include testing combinations that vary both the number of hours and the intensity levels not covered by the initial test cases (ISTQB Main Web).References:

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

NEW QUESTION 14

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 17

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 20

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 25

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 27

ST is a Software Testing organization which utilizes a testing knowledge base. Access to ST knowledge base can be either full or limited. Access level is determined based on ST certification and testing experience as follows:

- * 1. If ST certified, with less than 5 years testing experience - allow limited access
- * 2. If ST certified, 5-10 years of testing experience - allow full access
- * 3. If not ST certified with 5-10 years of testing experience - allow limited access.

What would be the results for:

- A - ST certified. 12 years of testing experience
- B - Not ST certified. 7 years of testing experience
- C - Not ST certified. 3 years of testing experience

- A. A - unknown B - limited access C- unknown
- B. A - full access B - limited access C - unknown
- C. A - full access B - limited access C - limited access
- D. A - unknown B - full access C - unknown

Answer: B

Explanation:

The correct answer can be derived by applying the given rules to each case:

? A is ST certified and has 12 years of testing experience, which is more than 10 years. Therefore, A does not match any of the rules and the result is unknown.
? B is not ST certified and has 7 years of testing experience, which is between 5 and 10 years. Therefore, B matches rule 3 and the result is limited access.
? C is not ST certified and has 3 years of testing experience, which is less than 5 years. Therefore, C does not match any of the rules and the result is unknown.
Verified References: This question does not require any external references, as it is based on logical reasoning.

NEW QUESTION 31

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 32

Which are the MAIN goals of risk management in a software project?

- A. To increase the success probability for the project regardless of costs.
- B. To increase focus on preventative processes and to increase satisfaction for the testers.
- C. To control contractual problems and minimize the impacts of company policies.
- D. To reduce the probability of undesired situations and to reduce the effect of potential impact.

Answer: D

Explanation:

Risk management is a process that identifies, analyzes, evaluates and mitigates risks in a software project. Risks are factors that may negatively affect the quality, schedule, budget or scope of a software project. The main goals of risk management in a software project are to reduce the probability of undesired situations and to reduce the effect of potential impact. This can be achieved by applying various strategies, such as avoidance, transfer, reduction or acceptance. Risk management does not aim to increase the success probability for the project regardless of costs, as this may not be feasible or realistic. Risk management does not aim to increase focus on preventative processes and to increase satisfaction for the testers, as these are secondary or indirect outcomes. Risk management does not aim to control contractual problems and minimize the impacts of company policies, as these are specific types of risks that may not apply to all projects. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 2, page 14-15.

NEW QUESTION 37

Why should you choose a test technique?

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

Answer: A

Explanation:

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

NEW QUESTION 39

Which test approach will best fit a new project, with little documentation and high probability for bugs?

- A. Exploratory testing
- B. Requirements based testing
- C. Metric based approach
- D. Regression testing

Answer: A

Explanation:

Exploratory testing is an approach to testing that emphasizes learning, test design and test execution at the same time. Exploratory testing relies on the tester's skills, creativity and intuition to explore the software under test and discover defects. Exploratory testing is suitable for a new project with little documentation and high probability for bugs, as it can help uncover unknown requirements, assumptions and risks. Exploratory testing is not requirements based testing, which is an approach to testing that derives test cases from documented requirements or specifications. Requirements based testing is not feasible for a new project with little documentation, as it requires clear and complete requirements to be available. Exploratory testing is not metric based approach, which is an approach to testing that uses quantitative measures to monitor and control the testing process and evaluate the quality of the software product. Metric based approach is not effective

for a new project with high probability for bugs, as it may not capture all aspects of quality and may lead to false confidence or unrealistic expectations. Exploratory testing is not regression testing, which is an approach to testing that verifies that previously tested software still performs correctly after changes. Regression testing is not relevant for a new project with no previous versions or baselines. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 5, page 47-48.

NEW QUESTION 44

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

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

Answer: D

Explanation:

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

NEW QUESTION 45

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

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

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

Answer: B

Explanation:

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

? Identifier: A unique identifier for the incident report

? Summary: A concise summary of the incident

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

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

? Priority: The level of urgency for resolving the incident

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

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

NEW QUESTION 47

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 49

Which of the following statements about reviews are TRUE?

- A. In walkthroughs the review meeting is typically led by the author
B. II Inspection is characterized by an open-ended review meeting
III 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 53

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 55

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

- A. Stubs and drivers are often used
II The test environment should correspond to the production environment
III Finding defects is not the main focus
IV Testing can be based on use cases
V Testing is normally performed by testers
VI Testing for functional and non-functional characteristics
B. Component - VI Integration - IV System - I Acceptance - II
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 56

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 61

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 65

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

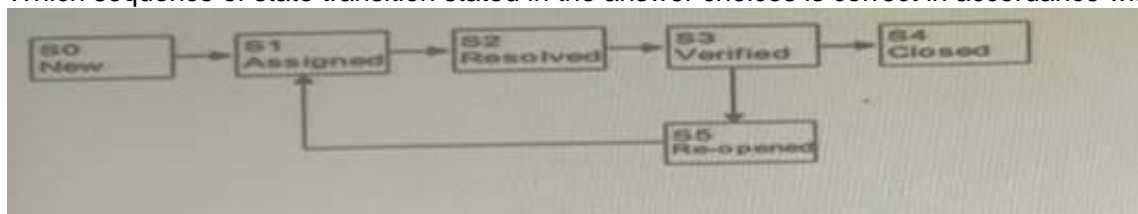
Answer: D

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 69

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



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

Answer: C

Explanation:

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

that follows the figure is S0->S1->S2->S3->S5->S1->S2->S3. This sequence represents the following scenario:

? S0: The defect is not yet detected (initial state).
? T0: The defect is detected by testing (event).
? S1: The defect is reported and registered (state).
? T1: The defect is assigned to a developer for fixing (event).
? S2: The defect is being fixed by the developer (state).
? T2: The developer fixes the defect and delivers a new version (event).
? S3: The defect is verified by testing (state).
? T5: The testing fails to confirm that the defect is fixed (event).
? S5: The defect is rejected by testing (state).
? T6: The defect is reassigned to a developer for fixing (event).
? S1: The defect is reported and registered (state).
? T1: The defect is assigned to a developer for fixing (event).
? S2: The defect is being fixed by the developer (state).
? T2: The developer fixes the defect and delivers a new version (event).
? S3: The defect is verified by testing (state). The other sequences are incorrect, as they do not follow the transitions shown in the figure. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION 70

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

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

Answer: A

Explanation:

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

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

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

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

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

NEW QUESTION 73

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 74

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 75

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

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

Answer: D

Explanation:

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

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

NEW QUESTION 78

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

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

Answer: A

Explanation:

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

NEW QUESTION 82

The following open incident report provided: Date: 01.01.01

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

Severity: High

Life Cycle: Integration

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

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

Answer: B

Explanation:

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

- I. Identification or configuration of the application: This detail is crucial as it specifies which version or configuration of the application is affected, helping in

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

NEW QUESTION 83

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

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

Answer: B

Explanation:

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

NEW QUESTION 86

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 89

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

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

Answer: D

Explanation:

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

References:

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

NEW QUESTION 92

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

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

Answer: A

Explanation:

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

NEW QUESTION 96

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 97

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 102

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 105

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, 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, 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 108

Which of the following BEST matches the descriptions with the different categories of test techniques?

- * 1. Test cases are based on the test basis which may include the requirements, use cases and user stories
 - * 2. Test cases are based on the test basis which may include the software architecture or code
 - * 3. Test cases can show deviations from the requirements
 - * 4. These test techniques are applicable to both functional and non-functional testing
 - * 5. Tests are based on knowledge of developers, users and other stakeholders
- Black - Black-box test techniques
White * White-box test techniques
Experience - Experience-based test techniques

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

Answer: A

Explanation:

Matching the descriptions with the test techniques:

- ? Black-box test techniques use the external description of the software, including requirements, use cases, and user stories.
- ? White-box test techniques use the internal structure of the software system, including software architecture and code.
- ? Black-box test techniques can reveal deviations from the requirements as they validate the external behavior of the software.
- ? Both black-box and white-box test techniques are applicable to functional and non-functional testing.
- ? Experience-based test techniques rely on the knowledge and intuition of developers, users, and other stakeholders (ISTQB Main Web).References:
? ISTQB® Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION 110

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 111

Which of the following is an INCORRECT statement about the benefit of traceability between the test basis and test work products?

- A. Traceability may be required by IT governance rules
- B. Traceability may help evaluate the extent of test coverage
- C. Traceability may allow testing to be auditable
- D. Traceability may make it harder to understand the impact of changes

Answer: D

Explanation:

The statement "Traceability may make it harder to understand the impact of changes" is incorrect. Traceability in testing actually facilitates understanding the impact of changes by linking test cases to requirements. This linkage helps ensure that any changes in the requirements are adequately reflected and verified in the test cases, thus supporting effective management of changes and maintaining the integrity of the system or product being developed (ISTQB not-for-profit association) (ISTQB Main Web).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 Official Website - CTFL Certification: <https://www.istqb.org/certifications/certified-tester-foundation-level/>

NEW QUESTION 115

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 117

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 120

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, I 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 125

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 129

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 130

Which of the following activities is NOT a pan of the fundamental testing process?

- A. Archiving automation code
- B. Test status reporting
- C. Test process improvement
- D. Build release and maintenance

Answer: D

Explanation:

The fundamental testing process includes activities that are directly related to the planning, preparation, execution, and evaluation of tests, as well as the closure activities of the testing phase. Option D, "Build release and maintenance," falls outside the scope of the fundamental testing process as it relates more to software development and operations rather than specific testing activities. Options A, "Archiving automation code," B, "Test status reporting," and C, "Test process improvement," are all activities that can be part of or associated with the fundamental testing process. Archiving automation code is part of test closure, test status reporting is part of test monitoring and control, and test process improvement can be an outcome of test closure activities.

NEW QUESTION 133

Which of the following statements about decision tables are TRUE?

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

Answer: D

Explanation:

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

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

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

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

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

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

NEW QUESTION 136

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 139

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 141

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 146

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 149

The ISTQB glossary defines Quality Assurance as: "Plan or 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 154

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 156

Which of the following is true about Oracles?

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

Answer: A

Explanation:

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

NEW QUESTION 157

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 161

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 165

A software calculates the annual car tax using three inputs:

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

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

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

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

Answer: B

Explanation:

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

NEW QUESTION 170

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 174

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 176

When should component integration tests be carried out?

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

Answer: D

Explanation:

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

NEW QUESTION 179

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 181

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 182

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 187

Given the following examples of entry and exit criteria:

- * 1. A defined level of code coverage has been achieved
- * 2. The test automation tool has been installed and properly configured
- * 3. The number of unresolved defects is within the predefined limit
- * 4. The performance test environment has been set-up and is available
- * 5. The user stories have proper acceptance criteria defined
- * 6. The testing budget has been spent and the project sponsor bears the risk of not testing any further

Which of the following BEST categorizes them as entry and exit criteria:

- A. Entry criteria - 2, 4, 5 Entry criteria -1, 3, 4
- B. Entry criteria - 2, 4 Entry criteria - 2, 4, 5, 6
- C. Exit criteria -1,3,6 Exit criteria - 2, 5, 6
- D. Exit criteria -1,3,5,6 Exit criteria -1,3

Answer: A

Explanation:

Entry and exit criteria are used to determine when to start and stop testing, respectively.

? Entry Criteria:

? Exit Criteria:

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, these criteria help in effectively managing the testing process⁶†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 190

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

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

Answer: B

Explanation:

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

NEW QUESTION 191

Which of the following statements is an example of testing contributing to higher quality?

- A. A test leader writes a test summary report
- B. A project manager asks to a test leader to estimate the test effort
- C. A tester installs a test ten in the lest environment
- D. A tester finds a bug which is resolved prior to release

Answer: D

Explanation:

? The question is about identifying an example of testing contributing to higher quality. Quality is the degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations¹. Testing is the process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects².

? Therefore, testing contributes to higher quality by verifying and validating that the software products and related work products meet the specified requirements, are fit for purpose and have no defects, or at least have a reduced number of defects. Testing also provides information about the quality of the software products and related work products to the stakeholders, who can make informed decisions based on the test results³.

? Out of the four given statements, only option D is an example of testing contributing to higher quality, as it shows that testing has detected a defect (a flaw in a component or system that can cause the component or system to fail to perform its required function⁴) and that the defect has been resolved (fixed and confirmed) prior to release (delivery of the software product to the customer or end user). This means that testing has prevented a potential failure (an event in which a component or system does not perform a required function within specified limits) from occurring in the operational environment, and thus has improved the quality of the software product.

? Option A is not an example of testing contributing to higher quality, as it is a reporting activity that summarizes the test results and evaluates the test objectives, but does not directly affect the quality of the software product or related work products. A test summary report is a document that records and communicates the outcomes of testing activities, including test completion criteria, test results, incident reports, test summary and evaluation, and lessons learned.

? Option B is not an example of testing contributing to higher quality, as it is a planning activity that estimates the resources and time needed for testing activities,

but does not directly affect the quality of the software product or related work products. A test effort estimate is an approximation of the amount of work and/or the duration of time required to perform testing activities.

? Option C is not an example of testing contributing to higher quality, as it is a preparation activity that sets up the test environment (an environment containing hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test), but does not directly affect the quality of the software product or related work products. A test environment installation is a process of installing and configuring the test environment according to the test environment specification. References:

- ? 1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 10
- ? 2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 11
- ? 3: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 12
- ? 4: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 13
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 13
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 77
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 78
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 79
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 80
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 81
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82
- ? : 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
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 86
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 87
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 88
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 89
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 90
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 91
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 92
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 93
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 94
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 95
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 96
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 97
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 98
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 99
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 100
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 101
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 102
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 103
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 104
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 105
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 106
- ? : ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 107

NEW QUESTION 194

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual ISTQB-CTFL Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the ISTQB-CTFL Product From:

<https://www.2passeasy.com/dumps/ISTQB-CTFL/>

Money Back Guarantee

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 FirstTry
- * ISTQB-CTFL Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year