

ISTQB

Exam Questions ISTQB-CTFL

ISTQB-Foundation Level Exam



NEW QUESTION 1

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 2

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 3

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 4

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

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

Answer: C

Explanation:

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

NEW QUESTION 5

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 6

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 z = 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 7

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

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

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

Answer: D

Explanation:

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

NEW QUESTION 8

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 9

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 10

Which of the following statements contradicts the general principles of testing?

- A. Most defects are found in a small subset of a system's modules.
- B. If new defects are to be found we should run the same test set more often.
- C. Testing is better if it starts at the beginning of a project.
- D. How testing is done, is based on the situation in a particular project.

Answer: B

Explanation:

Statement B contradicts the general principles of testing, because running the same test set more often will not increase the chances of finding new defects, unless there are some changes in the system or environment that affect the test results. Running different test sets with different inputs, outputs or conditions would be more effective in finding new defects. Statements A, C and D are consistent with the general principles of testing. Statement A states that most defects are found in a small subset of a system's modules, which is true according to the defect clustering principle. Statement C states that testing is better if it starts at the beginning of a project, which is true according to the early testing principle. Statement D states that how testing is done, is based on the situation in a particular

project, which is true according to the context-dependent testing principle. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, pages 4-6.

NEW QUESTION 10

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 15

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

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

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

Answer: B

Explanation:

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

Given the state model with the following transitions:

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

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

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

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

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

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

Given these, Option B covers all the transitions

? Certified Tester Foundation Level v4.0

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

NEW QUESTION 17

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 18

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

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

Answer: A

Explanation:

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

NEW QUESTION 22

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 24

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 26

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 31

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 32

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 37

Which of the following statements about estimation of the test effort is WRONG?

A. Once the test effort is estimated, resources can be identified and a schedule can be drawn up.

B. Effort estimate can be inaccurate because the quality of the product under tests is not known.

C. Effort estimate depends on the budget of the project.

D. Experience based estimation is one of the estimation techniques.

Answer: C

Explanation:

? Effort estimate does not depend on the budget of the project, but rather on the scope, complexity, and quality of the software product and the testing activities¹.

Budget is a constraint that may affect the feasibility and accuracy of the effort estimate, but it is not a factor that determines the effort estimate. Effort estimate is the amount of work required to complete the testing activities, measured in terms of person-hours, person-days, or person-months².

? The other options are correct because: References =

? 1 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 154

? 2 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 155

? 3 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 156

? 4 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 157

? 5 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 158

? 6 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 159

? 7 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 16

? [8] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 160

? [9] ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 161

NEW QUESTION 42

"Statement Testing" is part of;

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

Answer: D

Explanation:

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

NEW QUESTION 47

Which of the following BEST defines risk level?

- A. Risk level is determined by calculating the absolute value of the sum of all potential issues that may occur on the project
- B. Risk level is calculated by adding the probabilities of all planned risks to a project
- C. Risk level is calculated by dividing the sum of all known risks by the sum of all unknown risks
- D. Risk level is determined by the likelihood of an event happening and the impact or harm from that event

Answer: D

Explanation:

Risk level is determined by the combination of two factors: the likelihood of an event occurring and the impact or harm that could result from that event. This approach allows risks to be prioritized based on their potential effect on the project or system. The likelihood represents the probability of the risk event occurring, while the impact represents the severity of the consequences if the event does happen. This concept is fundamental in risk-based testing and helps guide decision-making during the testing process. References:

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.1.

NEW QUESTION 50

Which of the following tasks is MOST LIKELY to be performed by the tester?

- A. Develop a test strategy and test policy for the organization
- B. Promote and advocate the test team within the organization
- C. Create the detailed test execution schedule
- D. Introduce suitable metrics for measuring test progress

Answer: C

Explanation:

Testers are typically involved in creating detailed test execution schedules, among other tasks such as designing tests, executing tests, and logging defects. Creating a test strategy and test policy, promoting and advocating the test team, and introducing metrics are typically responsibilities of test managers or senior roles.

In the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, the responsibilities of testers include creating test cases, setting up test (ISTQB not-for-profit association)nts, executing tests, and reporting defects, which align with creating detailed test execution schedules^{6†}source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 53

Which of the following BEST describes checklist-based testing?

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

Answer: B

Explanation:

Checklist-based testing involves using checklists that contain items, such as potential test conditions, that should be tested. These checklists are often based on insights into what is important to the user, potential areas where software might fail, and specific aspects that need to be tested. It provides a structured yet flexible

approach to testing, ensuring key areas are covered while allowing testers to use their experience and understanding of the system. Checklist-based testing is not limited to non-functional testing but can be applied to various types of testing, including functional testing. References: ? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4.5.

NEW QUESTION 54

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

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

Answer: D

Explanation:

The following should be included in a test status report: total number of open and closed defects, actual effort spent, and number of executed, failed, and blocked tests.

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

NEW QUESTION 56

4 equivalence classes are given for integer values:

- $0 < x < 100$
- $100 \leq x \leq 200$
- $200 < x < 500$
- $x \geq 500$

Which of the following options represent correct set of data for valid equivalence class partitions?

- A. 50; 100; 200. 1000
- B. 0. 1.99, 100.200,201.499, 500;
- C. 0.50; 100; 150.200.350.500;
- D. 50; 100; 250; 1000

Answer: C

Explanation:

The correct set of data for valid equivalence class partitions should include one value from each equivalence class, and no value from outside the range. Option C satisfies this condition, as it has one value from each of the four equivalence classes (50, 100, 250, 500). Option A has two values from the same equivalence class (100 and 200), option B has values outside the range (0 and 0.99), and option D has two values from the same equivalence class (1000 and 500). Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 35.

NEW QUESTION 60

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

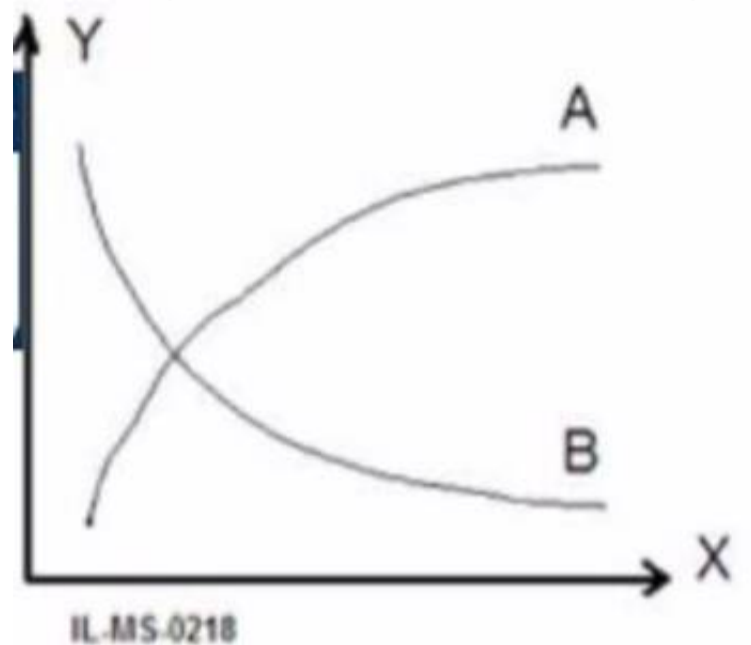
Explanation:

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 61

The following chart represents metrics related to testing of a project that was completed. Indicate what is represented by tie lines A, B and the axes X.Y



A)

X - Time

Y - Cost

A - Cost of test (per week)

B - Cost of finding a single bug (per week)

B)

X - Time

Y - Number of defects

A - Number of open defects

B - Number of closed defects

C)

X - Time

Y - Percent

A - % of functional tests in the test suite

B - % of non-functional tests in the test suite

D)

X - Time

Y - Count

A - Total number of executed tests

B - Number of open bugs

A. Option A

B. Option B

C. Option C

D. Option D

Answer: D

Explanation:

Option D correctly explains what is represented by the lines A, B and the axes X, Y in a testing metrics chart. According to option D:

? X-axis represents Time

? Y-axis represents Count

? Line A represents Number of open bugs

? Line B represents Total number of executed tests

This information is essential in understanding and analyzing the testing metrics of a completed project.

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 2.5.1, Page 35.

NEW QUESTION 64

The following 4 equivalence classes are given:

$x \leq -100$

$-100 < x < 100$

$100 \leq x < 1000$

$x \geq 1000$

Which of the following alternatives includes correct test values for x. based on equivalence partitioning?

- A. -100; 100;1000; 1001
- B. -500; 0; 100; 1000
- C. -99; 99;101; 1001
- D. -1000; -100; 100; 1000

Answer: D

Explanation:

? The question is about selecting the correct test values for x based on equivalence partitioning. Equivalence partitioning is a software test design technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In this case, the given equivalence classes are:

Option D provides a value from each of these partitions:

? For $(x \leq -100)$, it gives -1000.

? For $(-100 < x < 100)$, it gives -100 and 100.

? For $(100 \leq x < 1000)$, it gives 500.

? For $(x \geq 1000)$, it gives 1500.

So, option D covers all four given equivalence classes with appropriate values. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents available at ISTQB and ASTQB.

? 1: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 38

? 2: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 39

? : ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 40

NEW QUESTION 67

The testers in company A were part of the development team. Due to an organizational change they moved to be part of the support team.

What are the advantages and the disadvantages of this change?

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

Answer: B

Explanation:

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

NEW QUESTION 71

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 76

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 77

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 81

The following sentences refer to the 'Standard for Software Test Documentation' specification (IEEE 829).

Which sentence is correct?

- A. Any deviation from this standard should be approved by management, marketing & development
- B. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization
- C. The key to high quality test documentation regimes is strict adherence to this standard
- D. This test plan outline is relevant for military project
- E. For consumer market projects there is a different specification with fewer items.

Answer: B

Explanation:

The IEEE 829 standard is a widely used specification for test documentation, but it is not mandatory or universal. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization. The standard does not require any approval from management, marketing or development for any deviation, nor does it depend on the type of project (military or consumer market). The standard also does not guarantee high quality test documentation regimes, as it only provides a general outline and format, not the actual content or quality criteria. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, page 16.

NEW QUESTION 84

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

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

What would be your FIRST recommendation to the company?

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

Answer: B

Explanation:

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

Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 19; Chapter 4, page 45.

NEW QUESTION 85

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

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

Answer: C

Explanation:

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

? ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 5.3.1.

NEW QUESTION 90

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

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

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

Answer: C

Explanation:

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

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

? TC1 (Priority 1)

? **TC (ISTQB not-for-profit association)ity 2, dependent on TC1)

? TC2 (Priority 3, dependent on TC1)

? TC4 (Priority 4)

? TC5 (Priority 5)

? TC6 (Priority 6, dependent on TC4)

According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, understanding dependencies and scheduling tests accordingly is crucial for effective test execution6†source.

References:

? Certified Tester Foundation Level v4.0

? ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION 93

Consider the following statements about risk-based testing.

- I) Risk-based testing has the objective to reduce the level of protect risks.
- II) Tests should be prioritized to find tie critical detects as early as possible.
- III) Non-testing activities may also help to reduce risk
- IV) Risks have to be reassessed on a regular basis.
- V) The project stakeholders can give useful input to determine the risks

- A. I III IV and V are tru
- B. II is false.
- C. II, III IV and V are correc
- D. I is false.
- E. I, II and IV are tru
- F. III and V are false.
- G. II, III and V are tru
- H. 1 ants Iv are false.

Answer: B

Explanation:

The following statements about risk-based testing are correct:

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

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

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

NEW QUESTION 97

Which of the following activities does NOT belong to a typical technical review?

- A. Pre-meeting preparation by reviewers
- B. Using checklists during the meeting
- C. Inviting end-users to the meeting

D. Preparation of a review report

Answer: C

Explanation:

Technical reviews are structured meetings that aim to examine various aspects of a product or project to identify any defects or improvements. Options A (Pre-meeting preparation by reviewers), B (Using checklists during the meeting), and D (Preparation of a review report) are typical activities in a technical review process. Inviting end-users to the meeting (C), however, is generally not part of a typical technical review, as these reviews are usually more focused on the technical aspects and are conducted by peers or experts within the development or testing teams rather than end-users.

NEW QUESTION 99

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

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

Answer: A

Explanation:

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

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

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

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

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

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

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

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

NEW QUESTION 100

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

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

Answer: B

Explanation:

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

NEW QUESTION 105

Which of the following activities is NOT a part 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 106

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

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

Answer: A

Explanation:

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

NEW QUESTION 109

Given the following requirement: Requirement ID: 2 8

Requirement Description Additional Entrance Fee Detailed Description

An additional fee of S3 is charged during the weekend, but

- 1) Visitors aged under 7 are not charged.
- 2) Visitors aged 7 to 13 inclusive get a 20% discount off the additional fee.
- 3) Visitors aged greater than 65 get a 50% discount off the additional fee. Age should be an integer of 0 or above.

Weekend means Friday to Sunday inclusive. Which of the following statements is NOT correct?

- A. Thursday is a valid input boundary value.
- B. A minimum of 6 valid test cases are derived from boundary value analysis based on input age.
- C. \$3.01 is a valid output boundary value.
- D. 7 and 13 are boundary values for the equivalence partition including age 10.

Answer: A

Explanation:

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 applied to both input and output values. Based on the given requirement, we can identify two input values: age and weekend. Age should be an integer of 0 or above, and weekend means Friday to Sunday inclusive. The following statement is not correct:

? A) Thursday is a valid input boundary value. This statement is not correct, as Thursday is not a boundary value for the input weekend. The boundary values for the input weekend are Friday and Sunday, as they are at the edge of the equivalence partition that represents weekend days. The following statements are correct:

? B) A minimum of 6 valid test cases are derived from boundary value analysis based on input age. This statement is correct, as we can derive six valid test cases based on input age by using the minimum and maximum values for each equivalence partition defined by the requirement. The equivalence partitions for input age are: under 7 (0 to 6), 7 to 13 inclusive (7 to 13), and greater than 65 (66 and above). The minimum and maximum values for each partition are: 0 and 6, 7 and 13, and 66 and any value above it.

? C) \$3.01 is a valid output boundary value. This statement is correct, as \$3.01 is a boundary value for the output additional fee. The additional fee can have four possible values depending on the input age: \$0 (for visitors aged under 7), \$2.40 (for visitors aged 7 to 13 inclusive with a 20% discount), \$1.50 (for visitors aged greater than 65 with a 50% discount), and \$3 (for visitors aged between 14 and 65). The boundary values for the output additional fee are \$0 and \$3, as they are at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge. Therefore, \$3.01 is a valid output boundary value, as it is at the smallest incremental distance above \$3.

? D) 7 and 13 are boundary values for the equivalence partition including age 10.

This statement is correct, as 7 and 13 are boundary values for the equivalence partition that represents visitors aged 7 to 13 inclusive. This partition includes age 10, which is an internal value within the partition. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION 114

Can "cost" be regarded as Exit criteria?

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

Answer: A

Explanation:

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

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

NEW QUESTION 118

The following part of a business process flow is specified; REPEAT (book a bill) UNTIL (User presses Cancel). How many test cases are necessary in order to achieve 100% branch coverage of the process flow?

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

Answer: C

Explanation:

To achieve 100% branch coverage of the process flow, we need to test both the true and false outcomes of the condition (User presses Cancel). Branch coverage is a type of structural testing that measures how many decision outcomes in a program have been executed by a test suite. Branch coverage can be used to assess the adequacy or completeness of a test suite.

To test the true outcome of the condition, we need a test case that simulates the user pressing Cancel after booking a bill. This test case will exit the loop and end the process flow.

To test the false outcome of the condition, we need a test case that simulates the user not pressing Cancel after booking a bill. This test case will repeat the loop and book another bill.

Therefore, we need at least two test cases to achieve 100% branch coverage of the process flow. One test case for each possible outcome of the condition.

Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION 120

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 125

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 126

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 129

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

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

Answer: B

Explanation:

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

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

Exploratory testing and statement coverage are not suitable for testing the login page of a web site, as they are more applicable for testing components that require learning, creativity and intuition or structural analysis. Exploratory testing is an approach to testing that emphasizes learning, test design and test execution at the same time. Exploratory testing relies on the tester's skills, creativity and intuition to explore the software under test and discover defects. Statement coverage is a type of structural testing that measures how many statements in a program have been executed by a test suite. Statement coverage can be used to assess the adequacy or completeness of a test suite.

Decision coverage and fault attack are not suitable for testing the login page of a web site, as they are more applicable for testing components that have complex logic or potential errors. Decision coverage is a type of structural testing that measures how many decision outcomes in a program have been executed by a test suite. Decision coverage can be used to assess the adequacy or completeness of a test suite. Fault attack is a type of functional testing that deliberately introduces faults into a system in order to provoke failures or errors. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 4, page 34-46; Chapter 5, page 47-48.

NEW QUESTION 132

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 136

In what way do Configuration Management effects testing?

- A. Without proper configuration management, test planning cannot proceed.
- B. Proper configuration management ensures that testers can uniquely identify the tested item
- C. Configuration management is important for developers, not for testers

D. There is very little influence of configuration management practices on the test project.

Answer: B

Explanation:

Configuration management is a process that establishes and maintains consistency among work products throughout their life cycle. Configuration management affects testing in various ways, such as:

? Proper configuration management ensures that testers can uniquely identify the tested item, which can help traceability, reproducibility and accountability.

? Proper configuration management ensures that testers have access to consistent versions of software components and testware, which can help reliability, compatibility and efficiency.

? Proper configuration management ensures that testers can track changes and defects in software components and testware, which can help verification, validation and reporting.

? Proper configuration management ensures that testers can control the configuration of the test environment, which can help stability, security and performance.

Configuration management is not a prerequisite for test planning, as test planning can proceed without configuration management, although it may be less effective or accurate. Configuration management is not important for developers only, but for testers as well, as it affects the quality and consistency of the testing process and products. Configuration management has a significant influence on the test project, as it affects various aspects of testing, such as traceability, reproducibility, reliability, compatibility, efficiency, verification, validation, reporting, stability, security and performance. Verified References: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer, Chapter 6, page 60-61.

NEW QUESTION 141

Which of the following does MOT describe a reason why testing is necessary?

A. The customer decided that 100% branch coverage shall be achieved

B. The acquisition of test automation tools was based on the assumption that it will be used in all projects

C. For avionics and pharmaceutical systems software testing is mandated by standards

D. The risks associated with delivering the system are far higher than the cost of testing

Answer: B

Explanation:

Testing is necessary for various reasons, such as:

? To detect defects and failures that may affect the quality, performance, reliability or security of a software product or system

? To verify that a software product or system meets its specified requirements, expectations and standards

? To validate that a software product or system fulfills its intended purpose and satisfies its stakeholders' needs

? To provide information and feedback about the status and risks of a software product or system

? To comply with regulations or contractual obligations that mandate testing for certain types of software products or systems The following statements describe some reasons why testing is necessary:

? A) The customer decided that 100% branch coverage shall be achieved. This is a reason why testing is necessary, as it reflects a contractual obligation or a quality standard that requires testing to measure and achieve a certain level of code coverage.

? C) For avionics and pharmaceutical systems software testing is mandated by standards. This is a reason why testing is necessary, as it reflects a regulation or a compliance requirement that mandates testing for certain types of software products or systems that have high safety or security risks.

? D) The risks associated with delivering the system are far higher than the cost of testing. This is a reason why testing is necessary, as it reflects a risk-based approach that considers testing as an investment to reduce the probability and impact of potential failures or defects. The following statement does not describe a reason why testing is necessary:

? B) The acquisition of test automation tools was based on the assumption that it will be used in all projects. This is not a reason why testing is necessary, as it reflects a business decision or a resource allocation that does not justify the need or purpose of testing. Test automation tools are not always suitable or beneficial for all projects, and testing can be performed with or without test automation tools. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 5-6.

NEW QUESTION 144

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

Up to \$100.00 = 2%

\$100.01 to \$500.00 = 4%

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

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

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

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

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

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

Answer: B

Explanation:

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

Given the problem statement:

? Up to \$100.00 = 2%

? \$100.01 to \$500.00 = 4%

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

? Above \$1,000.00 = 7%

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

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

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

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

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

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

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

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

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

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

References:

? Certified Tester Foundation Level v4.0

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

NEW QUESTION 148

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

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

Answer: D

Explanation:

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

References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 1, page 4.

NEW QUESTION 150

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 a test leader to estimate the test effort
- C. A tester installs a test ten in the test 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 153

What is test oracle?

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

Answer: C

Explanation:

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

NEW QUESTION 154

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questions and Answers in PDF Format

ISTQB-CTFL Practice Exam Features:

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

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