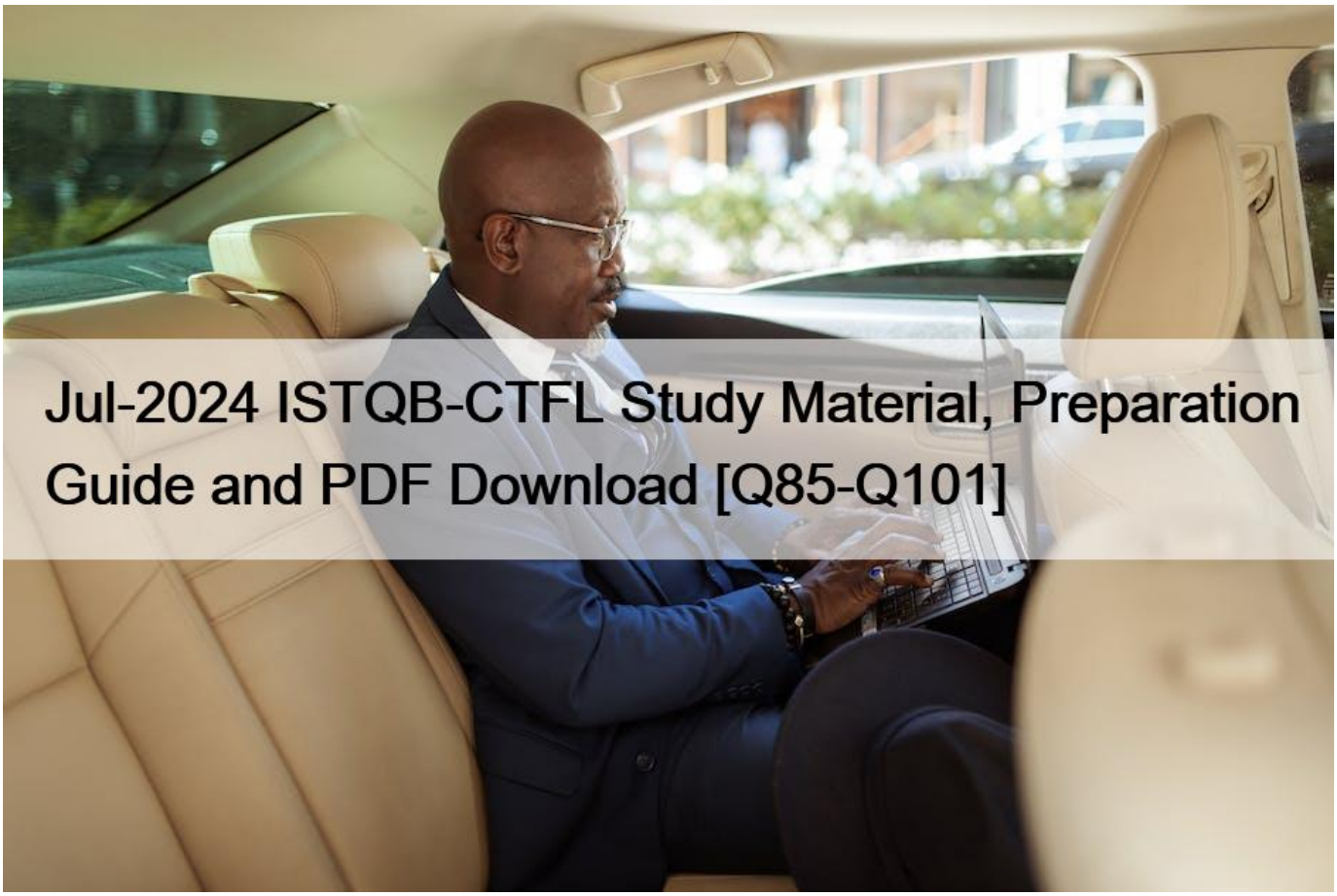
Jul-2024 ISTQB-CTFL Study Material, Preparation Guide and PDF Download [Q85-Q101



Jul-2024 ISTQB-CTFL Study Material, Preparation Guide and PDF Download Free ISTQB-CTFL Certification Sample Questions with Online Practice Test NEW QUESTION 85

4 equivalence classes are given for integer values:

0 < x <100

100<= x <= 200

200 < x < 500

x >= 500

Which of the following options represent correct set of data for valid equivalence class partitions?
* 50; 100; 200. 1000
* 0. 1.99, 100.200,201.499, 500;
* 0.50; 100; 150.200.350.500;

* 50; 100; 250; 1000

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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 35.

NEW QUESTION 86

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?

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

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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 41.

NEW QUESTION 87

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

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

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

NEW QUESTION 88

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

* Archiving automation code
* Test status reporting
* Test process improvement
* Build release and maintenance

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 89

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

I Estimation details

II Total number of open and closed defects

III Actual effort spent

IV Defect reports

V Number of executed, failed, blocked tests
* III.V
* II, III
* I. II. IV
* II, III.V

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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 141.

NEW QUESTION 90

A software calculates the annual car tax using three inputs:

– E; the emission level of the vehicle

– P: the power of the vehicle

-T the type of the vehicle

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

Which of the following answers contains a correct list of a boundary values for the P input?
* 14,351
* 14,15,350,351
* 15,350
* 5.175.500
A correct list of boundary values for the P input should include the minimum and maximum values of the valid range (15 and 350), as well as the values just below and above the boundaries (14 and 351). Boundary value analysis is a test design technique that involves testing the values at or near the boundaries of an input domain or output range, as these values are more likely to cause errors than values in the middle. Option B satisfies this condition, as it has all four boundary values (14, 15, 350, 351). Option A has only two boundary values (14 and 351), option C has only two boundary values (15 and 350), and option D has no boundary values at all. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 34.

NEW QUESTION 91

Which of the following statements about estimation of the test effort is WRONG?
* Once the test effort is estimated, resources can be identified and a schedule can be drawn up.
* Effort estimate can be inaccurate because the quality of the product under tests is not known.
* Effort estimate depends on the budget of the project.
* Experience based estimation is one of the estimation techniques.
* 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:

* A. Once the test effort is estimated, resources can be identified and a schedule can be drawn up, as they are interrelated aspects of the test planning process3. Resources are the people, tools, equipment, and facilities needed to perform the testing activities4. Schedule is the time frame and sequence of the testing activities, aligned with the project milestones and deadlines5.

* B. Effort estimate can be inaccurate because the quality of the product under tests is not known, as it affects the number and severity of the defects that may be found and the rework that may be needed to fix them6. Quality is the degree to which the software product satisfies the specified requirements and meets the needs and expectations of the users and clients7.

* D. Experience based estimation is one of the estimation techniques, which relies on the judgment and expertise of the testers and other project stakeholders to estimate the test effort based on similar projects or tasks done in the past. Experience based estimation can be useful when there is a lack of historical data, formal methods, or detailed information about the software product and the testing activities.

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 92

Which of the following statements is the BEST example of non-functional testing?
* Tests which capture the time it takes to save a file
* Tests which calculate overtime pay for those employees entitled to such
* Tests related to “what” the system should do
* Tests based on the internal structure of a component or system

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 93

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

1 – The valid ZIP code format is 5 numeric digits.

2 – The code has to exist in the post office’s official ZIP code list

Using equivalence classes partitioning, how many test cases are required to test this field?
* 8
* 3
* 6
* 4

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

* Valid ZIP code format and valid ZIP code value (e.g., 12345)

* Valid ZIP code format and invalid ZIP code value (e.g., 99999)

* Invalid ZIP code format and valid ZIP code value (e.g., 1234)

* Invalid ZIP code format and invalid ZIP code value (e.g., ABCDE) Therefore, four test cases are required to test this field, one for each equivalence class. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 4, page 37-38.

NEW QUESTION 94

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?
* Thursday is a valid input boundary value.
* A minimum of 6 valid test cases are derived from boundary value analysis based on input age.
* $3.01 is a valid output boundary value.
* 7 and 13 are boundary values for the equivalence partition including age 10.
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 95

A program is used to control a manufacturing line (turn machines on and off. start and stop conveyer belts, add raw materials to the flow. etc.). Not all actions are possible at all times. For example, there are certain manufacturing stages that cannot be stopped – unless there is an emergency. A tester attempts to evaluate if all such cases (where a specific action is not allowed) are covered by the tests.

Which coverage metric will provide the needed information for this analysis?
* Code coverage
* Data flow coverage
* Statement coverage
* Branch Coverage

Branch coverage is a type of structural coverage metric that measures the percentage of branches or decision outcomes that are executed by the test cases. A branch is a point in the code where the control flow can take two or more alternative paths based on a condition. For example, an if-else statement is a branch that can execute either the if-block or the else-block depending on the evaluation of the condition. Branch coverage ensures that each branch is taken at least once by the test cases, and thus reveals the behavior of the software under different scenarios. Branch coverage is also known as decision coverage or all-edges coverage.

Branch coverage is suitable for testing the cases where a specific action is not allowed, because it can verify that the test cases cover all the possible outcomes of the conditions that determine the action. For example, if the program has a condition that checks if the manufacturing stage can be stopped, then branch coverage can ensure that the test cases cover both the cases where the stage can be stopped and where it cannot be stopped. This way, branch coverage can help identify any missing or incorrect branches that may lead to undesired or unsafe actions.

The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types of coverage metrics, such as statement coverage, branch coverage, data flow coverage, etc. Code coverage does not specify which type of coverage metric is used for the analysis. Data flow coverage is a type of structural coverage metric that measures the percentage of data flow paths that are executed by the test cases. A data flow path is a sequence of statements that define, use, or kill a variable. Data flow coverage is useful for testing the correctness and completeness of the data manipulation in the software, but not for testing the conditions that determine the actions. Statement coverage is a type of structural coverage metric that measures the percentage of statements or lines of code that are executed by the test cases. Statement coverage ensures that each statement is executed at least once by the test cases, but it does not reveal the behavior of the software under different scenarios. Statement coverage is a weaker criterion than branch coverage, because it does not account for the branches or decision outcomes in the code. Reference = ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, Chapter 4: Test Techniques, Section 4.3: Structural Testing Techniques, Pages 51-54.

NEW QUESTION 96

Which of the following statements about Experience Based Techniques (EBT) is correct?
* EBT use tests derived from the test engineers’ previous experience with similar technologies.
* EBT is based on the ability of the test engineer to implement various testing techniques.
* EBT is done as a second stage of testing, after non-experienced-based testing took place.

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

NEW QUESTION 97

Which of the following is an example of black-box dynamic testing?
* Functional Testing
* Code inspection
* Checking memory leaks for a program by executing it
* Coverage analysis
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 98

Which of the following statements about independent testing is WRONG?
* Independent testing is necessary because developers don’t know any testing.
* Independent testing is best suited for the system test level.
* A certain degree of independence makes the tester more effective at finding defects.
* Independent test teams may find other types of detects than developers who are familiar with the system’s structure.
Independent testing is testing performed by a person or group that is independent of the development team. Independent testing can have various degrees of independence, ranging from testers who are part of the same organization as developers to testers who are external contractors or consultants. Independent testing can have various benefits, such as reducing bias, increasing objectivity,

improving quality, or providing different perspectives. Independent testing is not necessary because developers don’t know any testing, as this is a wrong and disrespectful statement. Developers can perform various types of testing, such as unit testing, component testing, or integration testing. However, independent testing can complement developer testing by providing additional levels of verification and validation, such as system testing, acceptance testing, or non-functional testing. Verified Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 2, page 16-17.

NEW QUESTION 99

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?
* Decision table testing, state transition testing.
* Equivalence partitioning, Boundary value analysis.
* Exploratory testing, statement coverage.
* Decision coverage, fault attack.

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 Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 4, page 34-46; Chapter 5, page 47-48.

NEW QUESTION 100

Which ONE of the following statements does NOT describe how testing contributes to higher quality?
* Properly designed tests that pass reduce the level of risk in a system.
* The testing of software demonstrates the absence of defects.

* Software testing identifies defects, which can be used to improve development activities.
* Performing a review of the requirement specifications before implementing the system can enhance quality.
* The testing of software does not demonstrate the absence of defects, but rather the presence of defects or the conformance of the software to the specified requirements1. Testing can never prove that the software is defect-free, as it is impossible to test all possible scenarios, inputs, outputs, and behaviors of the software2. Testing can only provide a level of confidence in the quality of the software, based on the coverage, effectiveness, and efficiency of the testing activities3.

* The other options are correct because:

* A. Properly designed tests that pass reduce the level of risk in a system, as they verify that the system meets the expected quality attributes and satisfies the needs and expectations of the users and clients4. Risk is the potential for loss or harm due to the occurrence of an undesirable event5. Testing can help to identify, analyze, prioritize, and mitigate the risks associated with the software product and project6.

* C. Software testing identifies defects, which can be used to improve development activities, as they provide feedback on the quality of the software and the effectiveness of the development processes7. Defects are flaws or errors in the software that cause it to deviate from the expected or required results or behavior. Testing can help to detect, report, track, and resolve the defects, and prevent them from recurring in the future.

* D. Performing a review of the requirement specifications before implementing the system can enhance quality, as it can ensure that the requirements are clear, complete, consistent, testable, and aligned with the needs and expectations of the users and clients. Requirements are the specifications of what the software should do and how it should do it. Testing can help to validate that the requirements are met by the software, and verify that the software is implemented according to the requirements.

References =

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

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

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

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

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

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

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

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

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

* [10] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 17

* [11] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 18

* [12] ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 19

NEW QUESTION 101

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

* The thought process is incorrect. The whole company should adopt same standard for review of all artifacts.
* The thought process is correct. The whole company should decide or the review method based on their CMM level.
* The thought process is incorrect. Same artifact can be reviewed using different review methods
* The thought process is correct. It wastes time to review same artifact using efferent review methods

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

3.2.2, page 33-34; Section 3.2.3, page 35-36.