

ISTQB.ISTQB-CTFL-KR.v2025-03-01.q124

□□□□:	ISTQB-CTFL-KR
□□□□:	ISTQB-Foundation Level Exam (ISTQB-CTFL Korean Version)
□□□:	ISTQB
□□ □□ □□□:	124
□□:	v2025-03-01
# □□ □:	957
# □□ □□□:	1240
https://www.krdump.com/ISTQB.ISTQB-CTFL-KR.v2025-03-01.q124.html	

NEW QUESTION: 1

□□□ □□ □□□ □, □□□ □□□□ □□□ □□□ □□□□ □□□ □□□□□?

- A. □□□□
- B. □□□
- C. □□
- D. □□

Answer: B ([LEAVE A REPLY](#))

In defect reporting, the attribute that indicates the degree of impact that the defect has on the system is the severity. Severity reflects the seriousness of the defect in terms of its impact on the operation of the system, ranging from minor issues that do not significantly affect the system's functionality to critical defects that can cause system failure. Therefore, option B is the correct answer.

NEW QUESTION: 2

□□ □ □□ □□□ □□ □□ □ □□ □□ □□□□□?

- A. 90%□ □□ □□□□□ □□□□ 90%□ □□ □□□□□ □□□□ □□□□□.
- B. 100% □□ □□□□□ □□□□ □□ □□ □□□ □□□ □□□□□ □□ □□□□ □□□□ □□□□□.
- C. 100% □□□ □□□□ □□□ 100% □□ □□□□ □□□ □□□□□.
- D. 80%□ □□ □□ □□□ □□ □□ □□ □□ □□□ □□□ 80%□ □□□□□□ □□□□□.

Answer: D ([LEAVE A REPLY](#))

Statement coverage measures the percentage of executable statements that have been exercised by a test suite.

Achieving 80% statement coverage means that 80% of the executable code lines have been tested. This metric helps in understanding how much of the code has been covered during testing. However, it does not guarantee branch coverage, variable initialization, or detection of all possible defects. The ISTQB CTFL Syllabus v4.0 explains statement coverage as a measure of

the extent to which the code has been tested, without implying other types of coverage or testing goals.

NEW QUESTION: 3

□□ □□ □ □□□□ □□□ □ □□ □□□ □□□□□ □□□□ □□ □□ □□□□□?

- A. □□□□ □□□ □□□□ □□□□ □□□□ □□ □□□ □□□□.
- B. □□□□□ □□□□ □□□ □□□ □□□□□.
- C. □□□□□ □□□□ □□□ □□□□ □□ □□□ □□□□ □ □□□ □ □□□□.
- D. □□□□ □□□□ □□ □□ □□ □□□ □□□□ □□□ □□□□ □ □□□□.

Answer: B ([LEAVE A REPLY](#))

* 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: 4

□□ □ □□□ □□ □□ □□□□?

A. □□□□□ □□□ □□ □□

B. □3□□ □□□ □□□ □□

C. □□ □□□, □□□ □□□ □ □□□□ □□□ □□□□.

D. □□□ □□ □□ □□ □□

Answer: A ([LEAVE A REPLY](#))

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

NEW QUESTION: 5

□□□ □□ □□ □□ □ □□ □□ □□□□?

I. □□□□□ □□□□□ □□□□ □□ □□□ □□□□□.

II □□□ □□□ □□ □□□ □□□□ □□□.

III □□ □□ □ □□□ □□□ □□□ □□□□□.

IV □□□□ □□ □□ □□□ □□ □□□□ □□□□.

A. II, III

B. □, □

C. 1, 4

D. 3, 4

Answer: C ([LEAVE A REPLY](#))

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: 6

□□ □□□□ □□ □□□□ □□□□ □□□ □ □□ □□(TAS)□ □□□ □□ □□ □□□□ □ □□□.

300,000€ □□□ □□ □□□□□ □□□□ □□ □□□□ □□□ □□□ □□ □□□ 100%□□ □. □ □□□□□□ "IF (TAS >= 300.000)" □□ "IF (TAS = 300,000)"□□□ □□□□ □ □□ □ □□ □□□□ □□□ □□□□. □ □□□ □□ 3□ □□□ □□□ □□□ □□ □ □□ □ □□ □□□□□(TAS□ □□ □□):

TC1 = 299,999 TC2 = 300,000 TC3 = 300,001

□□ □ □□ □□□ □□□□□?

- A. TC1□ □□□ □□ □□□□□.
- B. TC2□ □□□ □□ □□□□□.
- C. TC3□ □□□ □□□□□.
- D. □ □□ □□□ □□ □ □□ □□ □□□ □□□□□ □□□□□.

Answer: (SHOW ANSWER)

The requirement specifies that a bonus should be paid if the total amount of sales (TAS) made during the year is 300,000€ or more. The software incorrectly implements this requirement with "IF (TAS = 300,000)" instead of "IF (TAS >= 300,000)". Using boundary value analysis (BVA), which is a common technique in software testing, the three test cases provided (TC1 = 299,999, TC2 = 300,000, and TC3 = 300,001) cover the critical boundary values around the condition.

- * TC1 tests just below the boundary (299,999),
- * TC2 tests exactly at the boundary (300,000),
- * TC3 tests just above the boundary (300,001).

Since the software incorrectly checks for TAS equal to 300,000, only TC2 will fail because the condition is exactly met and highlights the incorrect implementation of the decision logic.

NEW QUESTION: 7

□□ □□□ □□ □□□□□□ □□ □□ □□□ □□□□□□□□. □□□□□ □□ □ □□ □□ □□ □□□□ □□□ □□□□□. □□ □□□□ □□ □ □□ □□□□ □□□ □□ □□□□□□?

- A. □□□□□□ □□□ □□ □□□□ □□ □□□
- B. □□ □□□ □□□ □□□ □□□

C. □□□□ □□□□ □□□ □□□□ □□ □□□□□ □□□

D. □□□ □□ □□□ □□ □□□ □□□□□ □□□ □□□□ □□ □□□

Answer: (SHOW ANSWER)

Confirmation testing, also known as re-testing, is performed to verify that specific defects have been successfully fixed.

Option A: "Testing due to the application of a new version of the interface" would typically involve regression testing, not confirmation testing.

Option B: "Testing that fixes resolved the defects in the search function" fits the description of confirmation testing as it focuses on ensuring that specific issues have been addressed.

Option C: "Testing if a system still works after update of an operating system" is an example of regression testing, as it checks the overall system behavior after an update.

Option D: "Testing to ensure the adding of new functionalities haven't broken existing functions" is another example of re (ISTQB not-for-profit association) (Udemy) it checks for unintended consequences of new changes.

Therefore, the correct answer is B6†source9†source.

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 8

□□□ □□□□□□ □□ □□□□ □□□□□ □□□ □□□□ □ □□ □□□ □□□ □□ □
□□ □□ □ □□□□□?

A. □□□

B. □□

C. □□ □□□

D. □□□ □□

Answer: (SHOW ANSWER)

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

NEW QUESTION: 9

□□□ □□□□□ □□□□□ □□□□□□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□
□□□ □□□ □□□ □□□ □□□□ □□□ □□□□.

- A.** □□ □□□□□
- B.** □□□□ □□□□'□□ □□□ □□□□ □□□□□.
- C.** '□□□ □□ □□'□□ □□□ □□□□ □□□□□.
- D.** '□□ □□'□□□ □□□ □□□□ □□□□□.

Answer: ([SHOW ANSWER](#))

The fact that defects are usually not evenly distributed among the various modules that make up a software application, but rather their distribution tend to reflect the Pareto principle, is expressed by the testing principle referred to as 'Defects cluster together'. This principle states that a small number of modules contain most of the defects detected, or that a small number of causes are responsible for most of the defects. This principle can be used to guide the test analysis and design activities, by prioritizing the testing of the most critical or risky modules, or by applying more rigorous test techniques to them. Therefore, option C is the correct answer.

References: ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 1.2.1, page 11; ISTQB Glossary v4.02, page 16.

NEW QUESTION: 10

[illegible]

- A.** □□□□□ □□ □□□□ □□□□□ □□□ □□□□ □□□ □□□ □ □□ □□ □□ □□
□□□, □□□□ □□□□ □□ □□ □□□□.
- B.** □□□□□ □□ □□□□ □□□□ □□ □□ □□ □□□□□□ □□□□ □□ □□ □□ □
□□ □□□□□ □□□□ □□ □□ □□□□□□ □□□ □ □□ □□ □□□□□.
- C.** □□□□□ □□ □□□□□ □□ □□□ □□ □□□□□□ □□□□ □□ □□□ □□ □□
□□□□ □□□□ □□□□ □□□□ □□ □□□ □□□ □□□ □□□□ □ □□□□.
- D.** □□□□□ □□ □□□□ □□□□ □□□□□□ □□□□□ □□□ □□□□□ □□□□ □
□□. □□□ □□□ □□ □□□ □□□□□□ □□□□ □□□ □□□ □□□ □□□ □□ □ □
□□□□ □ □ □□□□.

Answer: D (LEAVE A REPLY)

Checklist-based testing is a technique where testers use pre-determined checklists to ensure that important aspects of a work product are evaluated. Over time, these checklists should be reviewed and updated periodically to maintain their effectiveness in detecting defects. As systems evolve, outdated checklists may miss new types of defects, thus diminishing their usefulness. Therefore, statement D is true according to the ISTQB CTFL syllabus.

NEW QUESTION: 11

□□□□□ □□□□□□□ □□□ S100□ □□□□ □ □□□□ 50% □□□ □□ □□□□□
 □. □□□ □□ □□□ □□□ □□□□ 5%□ □□□□ □□□ □□□ □□□□□□. □□ □□
 □□ □□ □ □□□□□□ □□ □□□ 5%□ □□ 50%□ □□ □□□□ □□□ □□□□□□.
 □□ □□ □ □□□□□□ □□ □□□□ □□□□ □□ □□□□□□?

- A. □□ □□□ □□ □□□□, 50%□ □□□ □□□ □□□□, □□□ □□ □□□ □□□□□.
- B. □□□ □□ □□□ □□ □□ □□□ □□□□ □ □□□ □□ □□□□□.
- C. □□□ □□□ □□ □□□□□ □□□□□ □□, □□ □ □□ □□□ □□□□ □□ □□ □□ □□□□□.
- D. □□□ □□□ □□□ □□□ □□ □□□□ □□□□□□□.

Answer: B ([LEAVE A REPLY](#))

According to the ISTQB CTFL syllabus, a defect is a deviation from the expected result which in this scenario is the incorrect discount applied to the customers. The root cause, as per the ISTQB definition, is the originating cause of a defect, which in this case is the incorrect requirement stating 50% instead of 5%.

Therefore, the incorrect requirement is the root cause and the customer receiving the wrong discount is the effect of this root cause. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.4.3

"Defects, Root Causes, and Effects".

NEW QUESTION: 12

□□ □□□ □□□ □□□ □□□□□:

- A. □□, □□□, □□□, □□ □□, □□□, □□
- B. □□, □□□. □□□. □□ □□. □□□. □□
- C. □□, □□□□ □□□. □□□, □□ □□. □□□. □□
- D. □□. □□□, □□□. □□ □□. □□□. □□

Answer: A ([LEAVE A REPLY](#))

In a formal review, the roles involved typically include the author, management, facilitator (also known as moderator), review leader, reviewers, and scribe. Each role has specific responsibilities to ensure the effectiveness and efficiency of the review process:

- * The author creates and refines the work product being reviewed.
- * Management allocates resources and supports the review process.
- * The facilitator manages the review meeting, ensuring it proceeds smoothly.
- * The review leader plans the review and ensures it meets its objectives.
- * Reviewers examine the work product to identify defects.
- * The scribe records issues raised during the review meeting.

NEW QUESTION: 13

□□□□□ □□□□ □□ □□ □□ □ □□□ □□ □□□□□?

- A. □□ □□□□ □□ □□ □□ □□ □□ □□□□ □□□ □□□ □□□□ □□□ □□ □ □ □□□.
- B. □□□□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□ □□ □□□ □ □□□□ □□□□.
- C. □□ □□ □□ □□□□ □□ □□ □□ □□□ □□□□ □□ □□ □□ □□ □□□□ □□□ □□□□□□□ □□□ □ □□□□.

D. □□□□ □□□□ □□ □□ □□□□□ □□□ □□□ □□□□ □□□ □□ □□□□ □□ □□ □□□ □□ □□□□.

Answer: C ([LEAVE A REPLY](#))

Achieving full code coverage does not guarantee that the component or system is fully tested or free of defects. Code coverage metrics indicate the extent to which the source code has been tested, but they do not account for the quality of the tests or whether all possible scenarios have been considered. Other types of testing, including functional, performance, and security testing, are necessary to ensure comprehensive testing.

The ISTQB CTFL Syllabus v4.0 highlights that while high code coverage is beneficial, it does not equate to complete testing.

NEW QUESTION: 14

□□□□□ □□□□□ □□ □□□ □□ □□ □□□ □□□□ □□□ □□□.

"□□□□ □□□□□, □□□ □ □□□□ □□□ □□ □□ □□□□□ □ □□□□□ □□□□ □□□□."

□□□ □□□□ □□ □□□ □□ □□ □□□ □□□ □□□□.

[a] □□□□ □□□□ □□□□□ □□□□□ □□□ □, □□ □□□ □□□□ □□□□ □□ □□□□ □□□ □□□.

[b] □□□□□ □□□□□ □□□□□ □□□□ □□□□ □□ □□□□□ □□□□, □□□ □□ □□□□□ □□□ □ □□□□□ '□□□□ □□' □□□ □□ □□□□ □□□.

[c] □□□□ □□□□ □□□ □ □□□□□ □□□□□ □□ □□ □ □□□ □□□ □□□ □□ □□ □□ □□□ □□ □□□□ □□ 5□ □□□□□□ □□□ □□□.

[d] □□□□ □□□□ □□□ □□ □□□□□ □□□□ □□ □□ □□ □□□□ □□□□□ □□□.

[e] □□□□ □□□□ □□□ □□ □□□□□ □□□□□ □□□ □□□□□ □□□□ □□ □□ □□□□ □□□□□ □□□. □□□ □□□□ □□□ □, □□ ATDD □□□ □ □□ □□ □□□ □□□□ □□ □□□□ □□□□□?

A. □□□□ □□□□ □□□ □□ □□□□□ □□□□ □□ □□ □□□□ □□□□□.

B. □□□□ □□□□ □□□ □□ □□□□□ □□□□, □□□ □ □□□□(□□□ 5□ □□□□ □□□)□ □□□□, '□□□□ □□' □□□ □□ □, □□□□□ □□□□□ □□□□□ □□□ □□□ □□□□ □□□□ □□□□.

C. □□□□ □□□□ □□□ □□ □□□□□ □□□□, □□□ □□□□□ □□□ □ □□□□ □□□ □□□ □□□□□.

D. □□□□ □□□□ 10□ □□□ □□□ □□ □□□□ □□□□ Visa □□ □□□ □□□□□ □□□□ □□□ □□□ □□ □□ □□□ □□□□ '□□' □□□ □□ □ □□□□□ □□□ □□ □□□ □□□□ □□ □□□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

ATDD stands for Acceptance Test-Driven Development, which is a collaborative approach to software development and testing, in which the acceptance criteria of a user story are defined and automated as executable tests before the implementation of the software system. ATDD tests are usually written in a Given-When-Then format, which describes the preconditions, the

actions, and the expected outcomes of a test scenario. ATDD tests are intended to verify that the software system meets the expectations and the needs of the users and the stakeholders, as well as to provide feedback and guidance for the developers and the testers.

Based on the given information, the ATDD test that is most likely to be written first is the one that corresponds to option B, which is:

Given the logged-in user is on the Change Password page When the user enters the correct current password, enters a valid new password (different from the last 5 passwords), and presses the Change Password button Then the user receives an email confirming that the password has been successfully changed This ATDD test is most likely to be written first, because it covers the main functionality and the happy path of the user story, as well as the most important acceptance criterion [a]. It also verifies that the user can change the password with a valid new password that meets the criteria related to the length, the type of characters, and the history of the passwords, as specified in the acceptance criterion [c]. The other options are not likely to be written first, because they either cover less critical or less frequent scenarios, such as entering a wrong current password [d] or an invalid new password [e], or they are not related to the user story or the acceptance criteria at all, such as submitting a purchase order [d]. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles¹

* ISTQB Glossary of Testing Terms v4.0, Acceptance Test-Driven Development, User Story, Acceptance Criterion, Given-When-Then²

NEW QUESTION: 15

□□ □ □□ □□□ □□□ □□ □□ □□□□□?

A. □□□ □□□ □□□ □□□□ □□ □□ □□ □□□ □□□ □□□ □□ □□□□□.

B. □□□ □□□ □□ □□ □□□ □□□□ □□□□.

C. □□□□□ □□□ □□ □□□□ □□□□□ □□□□ □□□ □□□ □□□□.

D. □□□ □□□ □□□□□ □□ □□ □ □□ □□□ □□□ □□□□□ □□ □□□□□□.

Answer: C ([LEAVE A REPLY](#))

The absence-of-errors fallacy, as explained in the ISTQB syllabus, is the erroneous belief that having fewer defects found in testing equates to a system being more successful upon release. This misconception can lead stakeholders to undervalue thorough testing. Answer C illustrates this fallacy perfectly: assuming that because few defects were found, the system will be successful, neglects the many other factors that contribute to system success, including user satisfaction and fit-for-purpose. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 1.1.5 "Absence-of-errors fallacy".

NEW QUESTION: 16

□□□ □□□ □□ □□□ □□ □□□ □□□□ □□□□ □□ □□□ □□□□□. □□ □□□

□□ □□□□ □□□ □□□□.

□□ \$100.00 =2%

\$100.01 ~ \$500.00 = 4%

\$500.01 ~ \$1,000.00 = 5%

\$1,000.00 □□ = 7%

2□ □□□ □□□ □□□ □, □□ □□□ □□ □□ □ □□ □□ □□□□□ □□ □□ □□□
□□ □□ □□□ □□□□□?

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 (LEAVE A REPLY)

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

* Missing \$100.01 and \$500.01.

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

* Covers \$100.00, \$100.01, \$500.00, \$1000.00, and \$1000.01.

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

* Missing \$100.01 and \$500.01.

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

* Missing \$100.01 and \$500.01.

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

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ □□□□ ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 17

□□ □□□□□ QA □□□□ □□□ □□ □□□ □□□□ □□ □□□ □□□□ □□□.

□□ □ □□ □□ □□□ □□□□□?

- A. □□□□ □□ □□□ □□□ □□□□□.
- B. □□□□ □□ □□ □□ □ □□
- C. □□□□ □□□□□ □□ □□□ □□□□□.
- D. □□□□□ □□□ □□□□□□□□ □□□ □□□□□.

Answer: D (LEAVE A REPLY)

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

- * It is low cost, as it does not require purchasing any additional software or hardware.
 - * It is easy to implement within a week, as it does not require installing or configuring any complex software or hardware.
 - * It is accessible and transparent, as it can be viewed and updated by anyone who has access to the intranet.
 - * It is structured and organized, as it can store and display various information about incidents, such as identifier, summary, description, severity, priority, status, resolution, etc. The other options are not suitable for this case, as they have several disadvantages over the chosen option:
 - * Documenting incidents on a large board in the lab is not a good option, as it is not accessible or transparent to anyone who is not physically present in the lab. It is also not structured or organized, as it may not store or display all the necessary information about incidents.
 - * Purchasing and deploying an incident management tool is not a good option, as it is not low cost or easy to implement within a week. It may require spending a significant amount of money and time on acquiring, installing and configuring the software or hardware.
 - * Managing the incidents through emails and phone calls is not a good option, as it is not structured or organized. It may lead to confusion, inconsistency or loss of information about incidents.
- Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 3, page 32-33.

NEW QUESTION: 18

□□□ □□□ □□ □□□□□□.

□□: 01.01.01

□□: □□ □□□ □□□ □□ □□□□ □□□ '□□' □□ '□□'□ □□□□□.

□□□: □□

□□□ □□□: □□

□□ □ □□ □□ □□□□ □□□ □□ □□□ □□□□□?

I. □□□□□□□ □□ □□ □□

II. □□□□ □□

III. □□□□ □□

IV □□□ □□□ □□ □□□ □□ □/□□ □□

A. 4

B. 1. 4.

C. 1. 2.

D. 2, 3

Answer: B (LEAVE A REPLY)

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

I: Identification or configuration of the application: This detail is crucial as it specifies which version or configuration of the application is affected, helping in reproducing the issue. IV. The actions and/or conditions that came before pressing the button: Understanding the sequence of actions leading to the issue is vital for replicating and diagnosing the problem.

The name of the developer (II) and the recommendation of the developer (III) are not typically included in an incident report as they do not contribute to identifying or resolving the incident. The focus is on the incident's details, reproduction steps, and the system's state rather than on personnel or proposed solutions at this stage.

Therefore, option B, which includes both I and IV, is the correct answer.

NEW QUESTION: 19

□□ □□ □□□□ □□□□□□□ □□ □□□□ □□ □□□□ □□ □□□□ □□□□ □□□
□□□□(□□□ □□ □□□ □□□□ □□□ □)□ □□□□□ □□ □□ □□□□ □□□ □□
□□□□□.

A. □□□ □□□□ □ □□

B. □□□ □□

C. □□□ □□

D. □□□ □□

Answer: B (LEAVE A REPLY)

This answer is correct because test implementation is the activity where test work products, such as test cases, test data, test scripts, test harnesses, test stubs, or virtual services, are created and verified. Test implementation also involves setting up the test environment and preparing the test execution schedule. A virtual service emulating a real third-party service and the automated test scripts that interact with that service are examples of test work products that are typically

created during test implementation. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.3

NEW QUESTION: 20

□□ □□ □□□□ □□ □□□ □□ □□□□ □□□ □□□□□?

A. □□ □□□□ □□ SW □□□□□ □□□ □□□ □□□□ □□□□ □□□□ □□□□ □□□□.

B. □□ □□□ □□ □□□ □□□□ □□□ □□□□.

C. □□ □□□ □□□□ □□ □□□ □□ □□□□ □□(□□□ □□)□ □□ □□ □□□□ □□□□.

D. □□ □□□ □□□□ □ □□ □□□□ □□ □□□□ □ □□□□□.

Answer: D ([LEAVE A REPLY](#))

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: 21

□□ □ □□□□ □□□ □□□□ □□ □□ □□□ □□□□□?

A. □□□ □□ □□□ □□ □ □□□ □□ □□

B. □□ □ □□□ □□ □□□□ □□□□□.

C. □□□ □□□ □□ □□□ □□□□.

D. □□□ □□ □□ □□□ □□ □□□ □□ □□

Answer: C ([LEAVE A REPLY](#))

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 schedules6†source.

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 22

□□□□□ □□□ □□□ □□ □□ □□ □ □□ □□ □□□□□?

A. □□□□□ □□□ □□ □□ □□ □□ □ □□ □□ □□□ □□□□□ □□ □□ □□□□□ □□□ □□□ □□□□□□□ □□ □□ □ □□ □□□ □□□ □□ □□□□□.

B. □□□□ □□□□ □□□ □□ □□ □□□□□ □□□□ □□ □□ □□ □□□□□ □□□ □□□ □□□□□ □□□□. □□□□ □□ □□□□□ □□□□ □□□ □□□ □□□ □□□□ □□□ □□□□ □□□ □ □□ □□□□□.

C. □□ □□□ □□ □□□ □□ □□ □□□□□ □□□ □□□□□ □□ □□□ □□□ □□□ □□□ □□□□ □□ □□ □□ □□ □□ □□ □□□□ □□ □□ □□□ □□□□□ □□□□□.

D. □□ □□ □□□□□ □□□ □□□ □□□□□ □□□□ □□□ □□□□ □□□ □□ □□ □□□ □□ □□□□ □□□ □□□ □□□□ □ □□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

This answer is correct because code-related white-box test techniques are test design techniques that use the structure of the code to derive test cases. They provide an objective measure of coverage, such as statement coverage, branch coverage, or path coverage, which indicate how much of the code has been exercised by the test cases. Code-related white-box test techniques can be used to complement black-box test techniques, which are test design techniques that use the functional or non-functional requirements of the system or component to derive test cases. By combining both types of techniques, testers can increase their confidence in the code and find more defects. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.3.2.2

NEW QUESTION: 23

□□ □□□, □□ □□□ □□□ □ □□ □□ □□□ □□□□ □ □□□ □□□ □□□ □□□ □□□□□ □□□ □□□□□ □□□□ □□ □□□□□ □□□□ □□□□□.

A. □□□ □□

B. □□□ □□.

C. □□□ □□.

D. □□□ □□

Answer: ([SHOW ANSWER](#))

Test planning involves defining the overall approach to testing, including scheduling, resources, and milestones. It is during this phase that the detailed schedule for each testing activity is

determined based on estimates, resource availability, and constraints. The ISTQB CTFL Syllabus v4.0 outlines that test planning encompasses the creation of test plans and schedules to ensure that testing activities are properly managed and controlled.

NEW QUESTION: 24

Agile □□□□□ □□ □□□ □□□ □□□ 2□□ □□□ □□ □□ □□□ □□□□, □□□□ □ □□□□□□□ □□ □□□ □□□ □□□ □ □□□ □□ □□ □□□□ □□□□□□ □□ □□□□□.

□□□□ □□ □□ □□□ □□ □□ □□□□ □□ □□□ □□□ □ □□ □□ □□□□?

- A. □□ 1
- B. 2□□
- C. 3□□
- D. 4□□

Answer: C ([LEAVE A REPLY](#))

The unscripted tests produced by the tester's experience during the two-hour test session belong to the testing quadrant Q3. The testing quadrants are a classification of testing types based on two dimensions: the test objectives (whether the testing is focused on supporting the team or critiquing the product) and the test basis (whether the testing is based on the technology or the business). The testing quadrants are labeled as Q1, Q2, Q3, and Q4, and each quadrant represents a different testing perspective, such as unit testing, acceptance testing, usability testing, or performance testing. The testing quadrant Q3 corresponds to the testing types that have the objective of critiquing the product from the business perspective, such as exploratory testing, usability testing, user acceptance testing, alpha testing, beta testing, etc. The unscripted tests performed by the tester in the given scenario are examples of exploratory testing and usability testing, as they are based on the tester's experience, intuition, and learning of the web application, and they focus on some specific usability issues, such as the user interface, the user satisfaction, the user feedback, etc. The other options are incorrect, because:

- * The testing quadrant Q1 corresponds to the testing types that have the objective of supporting the team from the technology perspective, such as unit testing, component testing, integration testing, system testing, etc. These testing types are usually performed by developers or testers who have access to the source code, the design, the architecture, or the configuration of the software system, and they aim to verify the functionality, the quality, and the reliability of the software system at different levels of integration.
- * The testing quadrant Q2 corresponds to the testing types that have the objective of supporting the team from the business perspective, such as functional testing, acceptance testing, story testing, scenario testing, etc. These testing types are usually performed by testers or customers who have access to the requirements, the specifications, the user stories, or the business processes of the software system, and they aim to validate that the software system meets the expectations and the needs of the users and the stakeholders.
- * The testing quadrant Q4 corresponds to the testing types that have the objective of critiquing the product

* from the technology perspective, such as performance testing, security testing, reliability testing, compatibility testing, etc. These testing types are usually performed by testers or specialists who have access to the tools, the metrics, the standards, or the benchmarks of the software system, and they aim to evaluate the non-functional aspects of the software system, such as the efficiency, the security, the reliability, or the compatibility of the software system under different conditions or environments.

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles

* ISTQB Glossary of Testing Terms v4.0, Testing Quadrant, Exploratory Testing, Usability Testing, Unit Testing, Component Testing, Integration Testing, System Testing, Functional Testing, Acceptance Testing, Story Testing, Scenario Testing, Performance Testing, Security Testing, Reliability Testing, Compatibility Testing

NEW QUESTION: 25

□□□ □□ □□□ □□□ □□□ □□ □□□□ □□□ □□□□ □□ □□□□ □□ □□ □□ □□□?

A. □□□ □□ □□ □□□ SOL□ □□□□ □□□ □□ □□□ □□□□□□□□ □□□ SQL □□□ □□□ □ □□□□.

B. □□□ □□□□□□ □□□□ □□□□□□□ □□□ □□□□□ □□□□ □□ □ □□□□ □□.

C. '□□□ □□ □□' □□□ □□□ □□□ □□□□□ □□□ □□ □□□ □□□□□ □□□ □ □□ □□□ □□□□ □□□□□.

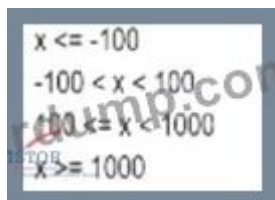
D. '□□ □□'□ □□ □□ □□□ □□ □□□ 10□ □□□□□□, □□ □□ □□□ 13□□□□ □.

Answer: B ([LEAVE A REPLY](#))

An incident report during testing should focus on factual observations of failures or defects in the system, including their impacts and how they deviate from expected results. Options A, C, and D describe specific issues that are directly related to the system's behavior or performance and are suitable for inclusion in an incident report. Option B, which describes the user interface as "complicated and confusing" and relates to the tester's personal difficulty in following the test script, is more subjective and relates to the tester's experience rather than an objective observation of a system failure. Therefore, option B is the least appropriate for an incident report.

NEW QUESTION: 26

□□□ 4□□ □□ □□□□□□.



□□ □□ □ □□ □□□ □□□□ □ x□ □□ □□□ □□□ □□ □□□□ □□ □□□□□?

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

Answer: D (LEAVE A REPLY)

* 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:

- * $(x \leq -100)$
- * $(-100 < x < 100)$
- * $(100 \leq x < 1000)$
- * $(x \geq 1000)$

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: 27

□□□ □□ □□ □□ □□□□□ □□ □□□ □□□ □□□.

□□□□ □□

□□ □□ □□□ □ □□ □□ □□□ □□ □□ □□□ □□ □□ □□□ □□□□□(□□ □□□ □□ □□□ □ □□□ □□□ □)?

- A. □□ □ -> □□□ -> □□□ -> □□ □ -> □□□ -> □□□
- B. □□ □ -> □□ -> □□ □□ -> □□ -> □□ -> □□ □
- C. □□□ -> □□ □ -> □□□ -> □□□ -> □□□
- D. □□□ -> □□□ -> □□□ -> □□□ -> □□□ -> □□ □ -> □□□

Answer: (SHOW ANSWER)

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 transitions6†source9†source.

References:

- * Certified Tester Foundation Level v4.0
- * 10 Sample Exams ISTQB Foundation Level (CTFL) v4.0

NEW QUESTION: 28

□□□ □□□□ □□□ □□ □□□ □ □□ □□□ □□□ □□□□.

- A. □□□ □□□ □□□□ □□ □ □□□□.
- B. □ □□□ □□□□ □□ □□□□ □□□ □ □□□□.
- C. □□ □□□ □□ □□□□ □□□□ □□ □ □□□□.
- D. □ □□□ □□□□ □□□□ □□ □□□□ □□□□ □□□□□.

Answer: A ([LEAVE A REPLY](#))

One possible risk of introducing test automation is that the selected tool may not be fit-for-purpose. This means that the tool might not meet the specific needs and requirements of the project, leading to inefficiencies and possibly failing to provide the expected benefits. It is crucial to evaluate and select the appropriate tool based on the project's context and objectives. The ISTQB CTFL syllabus highlights the importance of careful tool evaluation and selection to ensure it aligns with the testing goals and the development environment.

References: ISTQB CTFL Syllabus, Section 6.2, "Potential Benefits and Risks of Test Automation."

NEW QUESTION: 29

□□ □ □□□ □□□ □□ □□□ □□ □□□□ □□ □ □□□□ □□ □□□□□?

- A. □□□□ □□□□□ □□ □□ □□ □□□ □□ □□□ □□ □□□ □□□□□.
- B. □□□□ □□□□ □□ □ □□□ □□ □□□□ □□□ □□□□□.
- C. □□□□ □□ □□□□ □□ □□ □□ □□□ □□ □□□ □□ □□□ □□□□□.
- D. □□□□ □□ □□□□ □□ □ □□□ □□ □□□□ □□□ □□□□□.

Answer: C ([LEAVE A REPLY](#))

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: 30

□□□□ □□□□ 100% □□ □□□□ □□□□. □□□ □□ □ □□ □□□ □□□□ □□□□?

- A. □□ □□ □□□ □□□ □□□□□.
- B. □□ □□ □□□ □□□□ □□□□□□□□.
- C. □□ □□ □□□ □□□□ □□□□□□□□.
- D. "□□" □□□ □□□□□ □□□□□.

Answer: A ([LEAVE A REPLY](#))

If a program got 100% decision coverage in a test, then it is guaranteed that every executable statement is covered. Decision coverage (also known as branch coverage) is a type of structural coverage (also known as white-box coverage) that measures how many decision outcomes have been exercised by a test suite. A decision outcome is a possible result of a decision point (such as an if-then-else statement) in a program's code. Decision coverage requires that each decision point has both true and false outcomes executed at least once by a test suite. Decision coverage implies statement coverage, which is another type of structural coverage that measures how many executable statements have been executed by a test suite. Statement coverage requires that each executable statement is executed at least once by a test suite. Therefore, if a program got 100% decision coverage in a test, then it also got 100% statement coverage in a test, which means that every executable statement is covered. The other options are not guaranteed to be true if a program got

100% decision coverage in a test. Every output equivalence class has been tested and every input equivalence class has been tested are not guaranteed to be true if a program got 100% decision coverage in a test, because equivalence classes are based on functional requirements or specifications, not on code structure or logic.

Equivalence classes are used in specification-based testing (also known as black-box testing), which is a type of testing that does not consider the internal structure or implementation of the system under test. Decision coverage is used in structure-based testing (also known as white-box testing), which is a type of testing that considers the internal structure or implementation of the system under test. Therefore, achieving 100% decision coverage does not imply achieving 100% equivalence class coverage. The "dead" code has not been covered is not guaranteed to be true if a program got 100% decision coverage in a test, because dead code (also known as unreachable code) is code that can never be executed due to logical errors or design flaws.

Dead code can reduce readability and maintainability of the code, as well as increase complexity and size.

Decision coverage does not account for dead code, as it only considers the decision outcomes that are possible to execute. Therefore, achieving 100% decision coverage does not imply that the dead code has not been covered. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 36.

NEW QUESTION: 31

'□□□□ □□□'□ □□□□□?

- A. □□ □□□
- B. □□ □□□
- C. □□ □□ □□□
- D. □□□ □□

Answer: D ([LEAVE A REPLY](#))

Component testing is a test level. A test level is a group of test activities that are organized and managed together based on some common characteristics or objectives. A test level can be defined based on various factors, such as the scope and target of testing, the phase and model of development, the stakeholders and roles involved in testing, etc. Component testing (also known as unit testing or module testing) is a test level that focuses on verifying the functionality and quality of individual software components (such as modules, classes, functions, methods, etc.). Component testing can be performed by developers or testers using various techniques and tools depending on the type and complexity of the components. The other options are not test levels. Integration testing is another test level that focuses on verifying the functionality and quality of groups of software components that interact with each other or with external systems. 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. Experience-based testing is a category of test design techniques that rely on the tester's knowledge and intuition to derive and select test cases based on their experience with similar systems, technologies, domains, risks, etc. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 19.

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□. □□□□ □□□ □□□□ □□ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 32

□□ □□□ □□□ □□□ □□□.

1) □□ □□ □□□

2) □□ □□□

3) □□□ □□□

4) □□ □□□

□□ □ □□ □□□ □□□□□?

A. V-□□□ □□□□ □□ □ □□□ □□□□ □□□□□.

□□□ □□ □□□ □□□□ □□ □□ □□□□ □□ □□□□ □□□ □ □□□□.

B. □□ □□□ □□□ □□□□□ □□□ □□ V-□□□ □□ □□□□□.

□□ □□□□ □□ □□□□ □□□ □ □□□□.

C. □□ □□□□ □□ □□□□□ □□ □□□ □□□ □ □□□□.

□□ □□ □ □□□ □□□□ V□□□□ □□□□□.

D. □□□□□ □□ □□ □□ □□□□(V□□, □□□, □□□) □ □□ □□ □□□□ □□ □□
□ □□□ □□ □□□□□.

Answer: D ([LEAVE A REPLY](#))

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

NEW QUESTION: 33

□□ □ □□ □□□□ □□□ □ □□ □□ □□□ □□□□□?

A. □□□ □□

B. □□□□

C. □□□□ □□

D. □□ □□

Answer: ([SHOW ANSWER](#)**)**

Static analysis is the process of examining the work products of a software development or testing activity without executing them. Static analysis can be applied to various types of work products, such as requirements, design, code, test cases, etc. However, test plans are not suitable for static analysis, because they are high-level documents that describe the test objectives, scope, strategy, resources, schedule, and risks of a testing project. Test plans are not executable or formalized in a way that static analysis tools can analyze them.

Therefore, option A is the correct answer.

References: ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 2.2.1, page 20; ISTQB Glossary v4.02, page 45.

NEW QUESTION: 34

□□ □ □□□ □□□□ □□□ □□ □□□□□?

- A.** □□□□ □□□ □□□ □□□□□.
- B.** □□□□ □□ □□□ □□□□□ □□ □□□□□.
- C.** □□□□ □□□ □□□ □□□ □□□□□.
- D.** □□□□ □□□□ □□□ □ □□ □□□ □□ □□ □□□□ □□□.

Answer: A (LEAVE A REPLY)

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

NEW QUESTION: 35

□□ □□ □ □□ □□ □□□□□?

- A.** □□ □□□□ □□ □□□□ □□ □□□□ □□□ □ □□□, □□ □□ □□ □□□
□□ □□□ □ □□□□.
- B.** □□ □□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□, □□□□□ □□□□ □□
□□ □□ □ □□ □□□ □ □□□□□□ □□□ □□□□.
- C.** □□ □□□□ □□ □□□ □□□ □□□ □ □□□, □□□ □□□□ □□□ □ □□ □□□
□□□□ □□□ □ □□□□.
- D.** □□□□ □□□ □□□ □□ □□ □□□ □□□ □□ □□□□ □□□ □□□ □□□ □□□
□ □□□□.

Answer: D (LEAVE A REPLY)

Statement D is correct. According to the ISTQB CTFL syllabus, both black-box test techniques (which focus on testing without internal knowledge of the application) and experience-based test techniques (which rely on testers' experience and intuition) can be applied to both functional and non-functional testing. Functional testing is concerned with what the system does, whereas non-functional testing looks at how the system performs under certain conditions. These techniques are versatile and can be employed to address both these aspects.

NEW QUESTION: 36

☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐ ☐☐

☐☐☐?

- A.** ☐☐☐ ☐ ☐☐
- B.** ☐☐ ☐☐☐ ☐☐☐ ☐ ☐ ☐☐ ☐☐☐☐☐
- C.** ☐☐ ☐ ☐☐ ☐☐☐ ☐☐

D. □□ □□□ □□□ □ □ □□ □□

Answer: ([SHOW ANSWER](#))

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: 37

□□ □ □□□ □□□ □□□□ □□□ □□ □□ □□□□□?

A. □□ □□ □□□ □□ □□

B. □□□□ □□ □□

C. □□□□ □□□□ □□ □□□

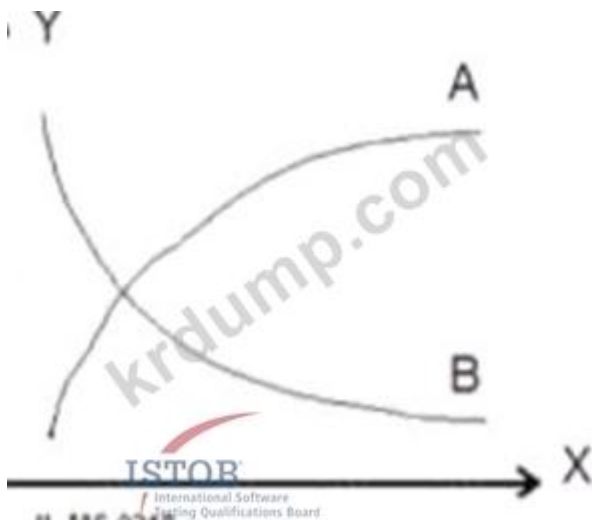
D. □□□□ □ □ □□ □□

Answer: ([SHOW ANSWER](#))

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

NEW QUESTION: 38

□□ □□□ □□□ □□□□□ □□□□ □□□ □□□□ □□□□□. □□ □□ A, B□ □ XY□ □□□□ □□ □□□□□.



X - Time

Y - Cost

A - Cost of test (per week)

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

X - Time
Y - Number of defects
A - Number of open defects
B - Number of closed defects

B.

X - Time
Y - Percent
A - % of functional tests in the test suite
B - % of non-functional tests in the test suite

C.

X - Time
Y - Count
A - Total number of executed tests
B - Number of open bugs

D.

Answer: D (LEAVE A REPLY)

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: 39

□□□□ □□□□ □□□ □□□ □□□□.

A. □□□ □□□ □□□□ □ □□ □□□ □□ □□ □□□□□.

B. □□□ □□□ □□ □□ □□□ □□□□□ □□□□□.

C. □□ □□□ □□ □ □□□ □□□ □□ □□□ □□□ □□□□□.

D. □□□ □□□ □□□ □□ □ □□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

One of the primary objectives of testing is to ensure that the software or system being tested meets all regulatory requirements. This is crucial in many industries where compliance with laws and standards is mandatory. According to the ISTQB CTFL Syllabus v4.0, testing aims to evaluate the quality of the software product and verify that it complies with specified requirements, including regulatory requirements.

NEW QUESTION: 40

□□ □ □□□ □□□□ □□□ □□ □ □□□ □□ □□□□□?

A. □ □□□□ □□

B. □□□□ □□ □□ □□

C. □□□ □□□ □□ □□□□□ □□□

D. □□□□ □□□ □□□ □□□ □□ □□□□ □□ □□□ □□□ □□

Answer: B ([LEAVE A REPLY](#))

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

Test automation allows testers to automate repetitive tasks such as regression testing, freeing up their time to focus on more complex and exploratory testing. This leads to improved efficiency and helps in avoiding human errors associated with repetitive tasks.

Option A: "More subjective assessment" contradicts the benefit of automation as it focuses on objectivity.

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

Option D: "Negligible effort to maintain the test assets" is misleading as maintaining automated tests often requires effort and attention to changes in the system under test.

Therefore, the correct answer is B (ISTQB not-for-profit association) (ISTQB).

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 41

□□□ □□ □□□□□ □□□□ □□□□□ □□□□. □ □□□□ □ □□ □□ □□□ □□□
□□□□□(□□ □□ □□ □□): □□□□, □□, □□ □□□. Preferred Guest Card □□□□ □
□□□□ □□□ □□□□.
□□□□ □□□□□ □□□ □□□□ □□ □□ □□□ □ □□□□.

Conditions

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



8	For upgrade to Gold Luxury	YES	NO	NO	NO
	For upgrade to Silver	N/A	YES	N/A	NO

□□ □ □□□ □□□ □□ □□□□ □□□ □□□□□?

□□ A : □□□□ □□□ □□□□□□ □□□

□□ B: □□□□ □□□ □□□ □□□ □□□ □□ □□□

A. □□ A: □□□□□□ □□□□ □□; □□ B: □□ □□□ □□□ □□□□□□ □□□

B. □□ A: □□□□□□ □□□□ □□□□. □□ B: □□□□□□ □□□□ □□□□.

C. □□ A: Gold Luxury □□□ □□□□□□ □□□; □□ B: □□□□□□ □□□□ □□

D. □□ A: □□ □□□ □□□□□□ □□□□□. □□ B: □□ □□□ □□□□□□ □□□□□.

Answer: C (LEAVE A REPLY)

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

NEW QUESTION: 42

□□ 12□ □□□ □□ □□□ □□ □□□ □□ □□□ □□ □□□ □□□ □, 13% □□□ □□
□□□ □□□ □□□ □□□ □□□□ □□ 2□ □□□ □□□ □□□□ □□ □□□□ □□□ □
□ □□□ □□ □□ □□□ □□□□□?

A. 10□ □□, 10□ □□, 50□ □□, 50□ □□.

B. 99,999□□, 100,000□□, 499,999□□, 500,000□□.

C. 10□ □□. 50□ □□.

D. 99,000□□, 500,001□□.

Answer: A (LEAVE A REPLY)

For boundary value analysis, the test cases should include the boundary values just inside and just outside the equivalence partition for the 13% interest rate range:

- * R100,000 (just inside the previous range)
- * R100,001 (start of the 13% range)
- * R500,000 (end of the 13% range)
- * R500,001 (just outside the range)

These values ensure that both the edges of the partition are tested.

NEW QUESTION: 43

□□ □ □□ □□□□ □□□□ □□ □□□□□?

- I. □□□ □□□ □□□□
- II. □□ □□□□ □□□□ □□□
- III. □□□□□ □□□ □□ □□□
- IV. □□ □□□□ □□ □□□□□ □□□□□.
- V □□□□ □ □□ □□□□□ □□□ □□□□□□.

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

Answer: C ([LEAVE A REPLY](#))

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

NEW QUESTION: 44

□□□ □□□ □□ □□ □□□□ □□□ □□□□□ □□□□ □□□. □ □□□□ □□ □□□ □□□□□ □□□□ □□□□ □□□□. □□ □□□ □□ □□□ □□ □□ □□□ □□ 20□ □□ □ □ □ □□□□ □□□□ □□□. □ □□□□ □□□ □□ □□ □□ □□□□ □□ □□ □□ □ □ □ □□ □□□□□?

- A. □ □□□□ □□□□ □□□□ □□ □□ □□□□ □□□□ □□□□.
- B. □□ □□ □□□□ □□□ □□□ □□□□□□ □□□□ □□□□□.
- C. □□ □□ □□□□ □□□ □□ □□□□ □□□□□.
- D. □□ □□□□ □□ □□ □□□□ □□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

Data migration testing is a critical part of maintenance testing during the decommissioning of a system. When a system is decommissioned, data often needs to be transferred to another system or archived securely. This process ensures that the data remains intact, accessible, and secure in

its new location. Therefore, statement B is true as it aligns with the ISTQB CTFL syllabus guidelines on handling system decommissioning and data preservation.

NEW QUESTION: 45

□□ □□ □□□ "□□□□"□ □□ □□ □□□ □□□□□.

1 - □□□ □□□□ □□□ 5□□ □□□□□.

2 - □□□ □□□□ □□ □□□□ □□□ □□□□ □□□.

□□ □□□ □□□ □□□ □□, □ □□□ □□□□□ □ □ □□ □□□ □□□□ □□□□□?

A. 8

B. 3

C. 6

D. 4

Answer: ([SHOW ANSWER](#))

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

- * Valid ZIP code format and valid ZIP code value (e.g., 12345)
- * Valid ZIP code format and invalid ZIP code value (e.g., 99999)
- * Invalid ZIP code format and valid ZIP code value (e.g., 1234)
- * Invalid ZIP code format and invalid ZIP code value (e.g., ABCDE) Therefore, four test cases are required to test this field, one for each equivalence class. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 4, page 37-38.

NEW QUESTION: 46

□□ □□ □ □□□□ □□ □□□ □□□ □□ □□ □□□□□?

A. □□□□ □□ □□ □□

B. □□ □ □□□□□ □□

C. □□ □□□□ □□□ □□□□□.

D. □□ □□□ □□

Answer: C ([LEAVE A REPLY](#))

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.

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

NEW QUESTION: 49

```
int x = 1;
int y = 2;
int y = 3;
X = y=z;
int z = 3;
X = z-y
Z = x +y
Y = z+y
```

A. int x = 1.

int y = 2.

int y = 3.

X = y=z;

B. int x = 1.

int y = 2.

int z = 3.

X = z-y

C. int x = 1.

int y = 2.

int z = 3.

Z = x +y

D. int y = 2

int z = 3.

Y = z+y

Answer: (SHOW ANSWER)

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: 50

□□ □□□ □ □□□□□□ □□ □□□□□ □ □□ □□□ □□ □□□ □□ □□□□ □□□
□□□□ □□ □□□□ □□□□□ □□□. □ □□ □□□ □□ □□□□ □□ □ □□ □□ □□
□ □□□□: 3%, 5%, 7%, 10%(□□□□ □ □□ □□ □□□ □□□). □□□ □□ □□□ □□
□□ □□ □□("□□" □□ "□□□□"□□ □□□)□ □□ □ □□ G1, G2, G3□□ □□□ □□
□□□ □□ □□□□□:

* [G1]: less than 300 euros

* [G2]: between 300 and 2000 euros

* [G3]: greater than 2000 euros

□ □□□□□ □□□ □□ □□□□□□ □□□□ □ □□□ □□□□ □□□ □□ □□ □□ □
□□ □□□□?

A. 12

B. 6

C. 4

D. 3

Answer: B ([LEAVE A REPLY](#))

The minimum number of test cases needed to cover the full decision table associated with this scenario is 6.

This is because the decision table has 4 conditions (type of customer and amount of sale) and 4 actions (bonus percentage). The conditions have 2 possible values each (Basic or Premium, and G1, G2 or G3), so the total number of combinations is $2 \times 2 \times 2 \times 2 = 16$. However, not all combinations are valid, as some of them are contradictory or impossible. For example, a sale cannot be both less than 300 euros and greater than 2000 euros at the same time. Therefore, we need to eliminate the invalid combinations and keep only the valid ones.

The valid combinations are:

Type of customer

Amount of sale

Bonus percentage

Basic

G1

3%

Basic

G2

5%

Basic

G3

7%

Premium

G1

5%

Premium

G2

7%

Premium

G3

10%

These 6 combinations cover all the possible values of the conditions and actions, and they are the minimum number of test cases needed to cover the full decision table.

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents,

NEW QUESTION: 51

□□ □□□□ □□ □ □□ □□□□□ □□ □□ □□□ □□□□□?

- A. □□ □□ □□, □□ □□, □□□ □□□ □□ □□□□□ □□ □□□ □□ □□□□ □□ □□□□.
- B. □□ □ □□□ □□□ □□□ □□ □□□□ □□□□□ □□ □□□ □□ □□ □□□□ □□ □□□□.
- C. □□ □□□□ "□□ □□" □□□ □□□ □□□ □□□□ □□□□□. □□ □□□ □□□□ □□□ □□□□ □□□□ □□□□.
- D. □□ □□□□ □□□□□□ □□□□ □□□ □□□ □□ □ □□□ □□ □□□ □□□□ □□□□ □□□□.

Answer: A ([LEAVE A REPLY](#))

Static testing is a form of testing that does not involve executing the software, but rather analyzing it for defects, errors, or violations of standards. Static testing can be applied to any software development product, including requirements specifications, design specifications, code, test cases, test plans, user manuals, etc.

Static testing can be done by using various techniques such as reviews, inspections, walkthroughs, checklists, static analysis tools, etc. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus

- Springer, page 7.

NEW QUESTION: 52

□□□ □□ □□ □□ □□□ □□□□□ □□□ □□ □(4□□ □□, 4~8□□, 9~12□□ □□ 12 □□ □□)□ □□ □□ □□ □□□ □□ □□ □□ □□(□□, □□, □□)□ □□□□ □□ □□ □□□□□.

□□ □□□ □□□ □□□□□□.

□□ □□ □□

T1 3 □□55

T2 14 □□ 95

T3 9 □□75

□□ □□□ INPUT □□ □□□□ □□ □□□ □□□□ □ □□□ □□ □□□ □□□□ □□ □□ □□□□□?

A. 1

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

Answer: C ([LEAVE A REPLY](#))

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

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

NEW QUESTION: 53

□□□ □□ □□□□ □□□ □□□□ □□□.

- A. □□ □□ □□ □□□ □□□ □□□ □□□□ □ □□□ □□ □□□ □□ □□□□ □□ □□ □□□ □□□□□□.
- B. □□□ □□ □□□ □□□ □□□□ □□ □□ □□□ □□□□ □ □□□ □□ □□□□ □□ □□ □□□ □□□□□.
- C. □□□□ □□□ □□□ □□□□ □□□□ □□□ □□ □□□ □□ □□□□ □□ □□□□ □□ □□□□ □□ □□□ □□□□ □□□□ □□ □□□.
- D. □□□ □□□□ □□ □ □□□ □□□, □□ □□□□□□ □□ □ □□ □□ □□ □□□□□ □ □□□ □□□□□.

Answer: A ([LEAVE A REPLY](#))

A test status report is a document that provides a snapshot of the testing activities and their progress during a particular period. It should include information about any impediments encountered during the test execution and the actions taken to resolve them, which helps stakeholders understand the challenges and how they were addressed .

Option B describes an activity related to test completion rather than ongoing status reporting.

Option C is incorrect because the structure and contents of the report may vary based on the audience's needs. Option D, while important, is not the primary purpose of a test status report, which focuses more on the current status and impediments.

NEW QUESTION: 54

□□ □ □□ □□ □□□ □□ □□ □ □□ □□ □□□□□?

- A. □□□□□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□ □□□ □□□ □□□□ □□□□ □□□□□□□□ □□□ □ □□□□.
- B. □□ □□ □□□ □□ □□□□ □□ □□□□ □□□□□ □□ □□□□ □□□ □□□ □□ □□□ □□ □□ □□□□□ □□□□ □□□□ □□□□ □□ □□□ □□□□ □ □□□ Shift-Left□□□.
- C. □□□□□ □□□□ □□□ □□ □□ □□□ □□□ □□□ □□ □□ □□ □□ □□□□ □ □□ □ □□□ □□□□ □□□ □□□ □□□ □□ □ □□□ □□ □□□□.
- D. □□□□ □□□ □□□ □□□ □ □□□□ □□□□ □□□ □ □□□□ □□□□□ □□ □ □□ □□□□ □□□ □□□□ □ □□ □□□ □□ □ □□□□.

Answer: D (LEAVE A REPLY)

This answer is correct because shift-left in testing is an approach that aims to perform testing activities as early as possible in the software development lifecycle, in order to find and fix defects faster and cheaper, and to improve the quality of the software product. Continuous integration is a practice that supports shift-left in testing, as it involves integrating and testing the software components frequently, usually several times a day, using automated tools and processes. Continuous integration can reduce the time between the introduction of a defect and its detection, thereby reducing the cost to fix it and the risk of accumulating defects that could affect the functionality or performance of the software product. References: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.3, Section 3.2.1.3

NEW QUESTION: 55

□□ □□ □□□ □□ □□□ □□□ □□□ □□□ □□□ □□ □□ □□ □□□□
□?

- A. □□ □□ □□
- B. □□ □□□ □□□□□ □□
- C. □□□ □□ □□
- D. □□□□ □□

Answer: B (LEAVE A REPLY)

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

NEW QUESTION: 56

- □ □□□ □□□□ □□ □□ □ □□ □□ □□□□□?
- A. □□□ □□□□ □□□ □□□□ □□ □□□□ □□□ □□□□ □□□□ □□□□.
 - B. BDD □ ATDD □□ □□□ □□□□ □□ □□□□ □□□ 02□ □□□□□.
 - C. □□□ □□□□ Q3 □□□□ □□□□, □□□□□ □□□□ □□ □□□□□ □□□□□.
 - D. TDD □□ □□□ □□□□ □□ □□□□ □□□ □□□□□ □□□□□ Q4 □□□□ □□□ □□.

Answer: B (LEAVE A REPLY)

The correct statement is B. According to the ISTQB CTFL syllabus, the testing quadrants help to categorize tests based on their purpose and whether they are technology-facing or business-facing, and whether they support the team or critique the product. Quadrant Q2 includes tests that are business-facing and support the team, such as automated acceptance tests produced during Behavior-Driven Development (BDD) and Acceptance Test-Driven Development (ATDD) . Quadrant Q3 includes business-facing tests that critique the product, such as exploratory testing, usability testing, and user acceptance testing. These tests are typically manual and focus on evaluating the product from a user perspective, rather than being part of a continuous integration process. Quadrant Q4 includes technology-facing tests that critique the product, such as performance tests, security tests, and other non-functional tests, which can be automated but are not related to unit tests produced during TDD .

NEW QUESTION: 57

□□ □ □□□□ □□□□ □□□ □□ □ □□□□ □□ □□□□□?

- A. □□□□ □□□ □□□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□□ □□ □□□□ □□□ □□□ □□ □□□□ □□□ □□□□ □□□ □□ □□□□.
- B. □□ □□ □□□ □□□□ □□ □□□□ □□□ □□ □□□□ □□□□ □□ □□□ □□□ □□□□.
- C. □□□□ □□□ □□ □□□ □□ □ □□□□ □□ □□□ □ □□□□.
- D. □□□□ □□□ □□ □□□□□ □□□ □□ □□ □□□ □□□ □□ □□□□ □□□ □□ □□□□ □□ □□□ □□ □□□□ □□□□ □□□□ □ □□□□.

Answer: C ([LEAVE A REPLY](#))

Independent testing offers several advantages, such as unbiased testing and detection of different defects.

However, a drawback is that an independent test team may be isolated from the development team and project team. This can lead to communication gaps, reduced collaboration, and a lack of understanding of the project context.

According to the ISTQB Certified Tester Foundation Level (CTFL) syllabus v4.0, an independent test team may not have the same level of understanding of the system as the development team, leading to potential issues in communication and integration (ISTQB not-for-profit association).

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 58

□ □□□□ □□□ □□□□ □□□□□ □□□. □ □□□□□ □□□ □□□ □□□□ □□□ □□□□. □ □□□ □□ □□□ □□□ □□ □□□ □□□□□?

- A. □□□□ □□□ □□□, □□ □□ □□□.
- B. □□ □□, □□□ □□.

C. □□□ □□□, □□ □□.

D. □□ □□, □□ □□.

Answer: B ([LEAVE A REPLY](#))

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: 59

Which of the following is NOT an outcome of control flow analysis?

- A. Identification of unreachable code
- B. Report on adherence to coding standards
- C. Complexity measurement
- D. Number of comment lines and number of source code lines

Answer: A ([LEAVE A REPLY](#))

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

Number of comment lines and number of source code lines are outcomes of complexity measurement.

Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 8.

NEW QUESTION: 60

Which of the following is NOT a principle of the Waterfall model?

- A. Early testing
- B. V-model
- C. Requirements gathering and design
- D. Finding defects later in the development cycle

Answer: (SHOW ANSWER)

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

* ISTQB Certified Tester Foundation Level Syllabus v4.0:

https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf

<https://www.istqb.org/news/posts/istqb-releases-certified-tester-foundation-level-v40-ctfl/>

□□ □ □□ □□□ □□□□ □□□□ □□□ □□ □□□□ □□ □□□□□?

- Answer: D (LEAVE A REPLY)**

ISTQB-CTFL-KR                     **ISTQB-CTFL-KR**                          

□□□ □□□□ □□□□ Agile □ □□ □□ □□□□ □□□□ □ □□□□□□□.

- □ □□ □□ □□ □□□□□?
- A.** □□□ □□□□ □□□□ □□□ □□□□ □□□ □□□ □□ □□ □□□□□ □
□□ □ □□□□.
- B.** □□□ □□□□ □□□□ □□□ □□□□ □□□□ □□ □□□ □□ □□ □□□□□ □□
□ □ □□□□.
- C.** □□□ □□□□ □□□ □□□□ □□□□ □□□ □□ □□□ □□□□ □□□□ □□□ □
□□□□ □□□ □ □□□□.

D. □□□ □□□□ □□□□ □□□ □□□□ □□ □□ □□□□□ □□□ □□□ □□□□ □
□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

The Testing Quadrants model helps Agile teams by categorizing different types of tests and their purposes.

This differentiation helps test managers explain the testing strategy to all stakeholders, ensuring everyone understands the scope and objectives of each test type. This model aids in planning, executing, and tracking testing activities across different quadrants, making it easier to align with stakeholders' expectations and project goals. Reference: ISTQB CTFL Syllabus V4.0, Section 5.1.7

NEW QUESTION: 63

[illegible]

- A.** □□ □□□□
- B.** □□ □□□ □□□□□□ □□□ □□□□□□.
- C.** □□ □□□ □□□□□
- D.** □□□ □□ □□ □□

Answer: B (LEAVE A REPLY)

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

- * Planning the review (correct responsibility)
- * Determines if the review objectives have been met (incorrect responsibility)
- * Decide on the execution of reviews (correct responsibility)
- * Allocate time for review (correct responsibility) The responsibility of determining if the review objectives have been met belongs to the moderator role, not to the manager role. Verified

References:

[A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 3, page 28-29.

NEW QUESTION: 64

□□ □□□ □□ □□□ □□□ □□□ □□□□□? □□□□ 1□□ 15□□□ □□ □□□□ □
□□.

- A.** 0.1 □□ ~ 14. 15 □□
B. 1.1 □□ ~ 14. 15 □□
C. □□. 1~15. 15 □□
D. 1.1 □□ ~ 15. 15 □□

Answer: ([SHOW ANSWER](#))

Equivalence partitioning is a black-box test design technique where inputs to the software or system are divided into groups that are expected to exhibit similar behavior. For an integer field

that should accept values from 1 to 15, the valid equivalence class is 1 through 15. The invalid equivalence classes are numbers less than 1 and numbers more than 15. Therefore, option D, "Less than 1, 1 through 15, more than 15," correctly identifies the valid equivalence class along with the two invalid classes, covering all possible input scenarios for the field. Options A, B, and C either do not accurately capture the valid range or incorrectly specify the range boundaries.

NEW QUESTION: 65

□□□ □ □□□ □□□□ □□ □□ □□ □□□□ □□ □□□ □□□□ □□ □□□□ □□ □□□.

Test Case	Priority	Logical Dependencies
TC1	Low	TC5
TC2	High	TC3
TC3	High	TC4
TC4	High	ISTQB
TC5	Low	TC2
TC6	Medium	-

□□ □□□ □□ □□ □ □□□ □□□□ □□□□ □□ □□□ □□ □□□ □□□ □□ □ □ □□ □□□□□?

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

Answer: D (LEAVE A REPLY)

This answer is correct because it follows the logical dependencies and allows executing the test cases in priority order. TC4, TC3, and TC2 are executed first because they have the highest priority. TC6 is executed next because it has a logical dependency on TC2. TC1 is executed next because it has a logical dependency on TC5. Finally, TC5 is executed last because it has the lowest priority. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 documents

NEW QUESTION: 66

□□ □ □□□□□ □□□ □□□ □□□ □□ □□ □ □□□□ □□ □□ □□□□□?

- A. "□□"□ 1000□□□□ □□ □□□ □□□ □□□□ □ □ □□□□□.
- B. □□□□ □□□ □□□□ □□ □□□ 9□□□, □□□□□ 10□□□□□□□.
- C. □□□□□□□ □□□ □ □□ □□ □□□ □□ □□□ □□□□.
- D. □□□□□□□ □□□□□ □□□ □ □□□ □□□ □□□ □□□□ □□□□.

Answer: (SHOW ANSWER)

A software bug can cause harm to a company by directly affecting its operations, reputation, user satisfaction, and financials. Option D, "When calculating the final price in a shopping list, the price of the last item is not added," describes a defect that directly impacts the core functionality of a

financial transaction, potentially leading to financial loss and customer dissatisfaction. This can have severe implications for the company's credibility and revenue. Options A, B, and C describe bugs that, while potentially annoying, do not have the same direct impact on the company's core operations and financial integrity as option D.

NEW QUESTION: 67

□□□ □□ □□□ □□□□□ □□□ □□ □□□ □□ □□ □□□□□ □□□ □□□□□□. □□ □□□ □□□□ □□ □□□?

- A. □□ □□ □□□.
- B. □□□□□ □□ □□□.
- C. □□ □ □□□ □□□□□□.
- D. □□□ □□□.

Answer: D ([LEAVE A REPLY](#))

When working on a project with incomplete requirements and under pressure to deploy, exploratory testing is particularly suitable. This type of testing allows testers to use their expertise and intuition to explore the system's functionality and identify defects without needing detailed specifications. Exploratory testing is flexible and can quickly adapt to changes and gaps in the requirements.

NEW QUESTION: 68

□□ □□ □ □□□ □□ □□ □□□□□?

- A. □□□ □□□ □□ □□□ □□□□□□□ □□□, □□□□ □□□ □□□ □□□ □□□ □ □□ □□□ □□□ □□□□ □□□□□.
- B. □□□ □□ □□□ □□□□ □□□ □□□□□ □□□□ □□□ □□□ □□□□□.
- C. □□□ □□ □□□ □□□ □□ □□ □□□□ □□□ □□□□□ □□□□□□ □□□□□.
- D. □□□ □□□□ □□, □□□□□□ □□ □□□ □□ □□ □□□□ □□□□□.

Answer: C ([LEAVE A REPLY](#))

Test Management tools are designed to support the planning, execution, and monitoring of the testing process.

They provide features for managing test cases, test runs, tracking defects, and reporting on testing activities.

However, the statement in option C describes Test Management tools as monitoring and reporting on the system's behavior during testing activities, which is not accurate. Test Management tools focus on the testing process itself rather than on the behavior of the system under test.

* Test data preparation tools (A) indeed create and manage test data for use during test execution.

* Test execution tools (B) automate the execution of test cases and the comparison of actual outcomes against expected results.

* Test comparators (D) are tools that compare actual outcomes with expected outcomes, highlighting discrepancies.

Therefore, option C is the correct answer as it inaccurately describes the function of Test Management tools.

NEW QUESTION: 69

□□ □ □□□ □□ □□□ □□□□ □□□ □□□□ □□ □□ □□ □□□□□?

A. □□□ □□

B. □□□□ □□□ □□□ □□

C. □□□□ □□□ □□ □□□□ □□□

D. □□□□ □□ □□□ □

Answer: ([SHOW ANSWER](#))

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

NEW QUESTION: 70

□□ □□ □□□ □□□□ □□□ □□ □□ □□ □ □□ □□ □□□□□?

A. □□ □□□ □□ □□ □□ □□□ □□□ □□ □□□□ □□□□□ □□ □ □□□□ □□□ □□.

B. ISO/IEC/IEEE 29119-3 □□□□ □□□□ □□ □□ □□ □□□ □ □□ □□□ □□ □□□ □□□□ □□□□.

C. □□ □□□□ □□□ □ □□□ □□□□□ □□ □□ □□□□ □□□□□.

D. □□ □□□ □□ □□ □ □□□ □□ □□□□ □□ □□□□□□ □□□ □□□ □□□ □□ □□□ □□□□ □□□□.

Answer: C ([LEAVE A REPLY](#))

During the review planning stage, key decisions are made, including the selection of standards and procedures to be followed during the review. This planning phase ensures that the review process is structured and adheres to agreed-upon standards, which can come from industry standards such as ISO/IEC/IEEE 29119-3.

The ISTQB CTFL Syllabus v4.0 emphasizes the importance of review planning in establishing the framework and guidelines for the review process.

NEW QUESTION: 71

□□ □ □□ □□ □□ □ □ □ □ □ □□□□?

- A. Shift-Left □□□ □□□ □□□□□ □□ □□□□□.
- B. □□□□□□ □□ □□ □□□ DevOps □□□ □□□□□.
- C. □□□□ □□□□ □□ □□□ □□ □□□□ □□□ □ □□□□.
- D. □□□□ □□ □□□ □□ □□ □□□ □□□ □ □□□□.

Answer: ([**SHOW ANSWER**](#)**)**

In a formal review process, the recorder's role is typically responsible for documenting the findings of the review team, including action items, decisions, and recommendations. This ensures that there is an accurate record of what was discussed and agreed upon, facilitating follow-up and continuous improvement. Therefore, statement C is correct as per the ISTQB CTFL syllabus.

NEW QUESTION: 72

□□ □ □□ □□□ □□□□□ □□ □□ □□ □□□ □□□□ □□ □□ □□□ □□□□□?

- A. □□□□ □□□□ □□□
- B. □□□□□ □□ □□□□□□ □□□ □□□ □□□ □□ □□
- C. □□□□ □□ □□□ □□ □□□□ □□□
- D. □□□□ □□□ □□

Answer: D ([**LEAVE A REPLY**](#)**)**

The choice of a software development lifecycle (SDLC) model is primarily influenced by the type of product being developed. Different products and project requirements may demand different SDLC models to address specific challenges and needs efficiently. For instance, a complex, safety-critical product might best be served by a Waterfall model due to its structured nature and phase dependencies, while a more iterative and incremental model might be suited for projects requiring frequent feedback and changes. References: ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 2.1 "Software Development Lifecycles".

NEW QUESTION: 73

□□ □ □□□□ □□□□ □□ □□ □ □□ □□ □□□□□?

- A. □□□□ □□□□ □□ □□ □□□ □□□ □□□□ □□□□ □□□□□.
- B. □□□□ □□□□ □□□ □□□ □□□ □□ □□□□□.
- C. □□ □□□ □□□□ □□□□ □□□ □□ □ □ □□□□□□ □□□□.
- D. □□□□ □□□ □□ □□□ □□□ □□□ □□□□□ □□ □□□ □□□ □□□ □ □□□ □.

Answer: ([**SHOW ANSWER**](#)**)**

Independent testing is testing performed by a person or group that is independent of the development team.

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

NEW QUESTION: 74

□□ □ PROJECT □□□ □□□ □□□ □□□ □□ □□□□□?

A. □□ □ □□ □□, □□□ □□ □□ □□ □□, □□ □□.

B. □□ □ □□ □□, □□□□□□ □□□ □□□ □□□□ □□, □□□ □□ □□ □□□ □□ □□.

C. □□□ □□□□ □□ □□, □□ □□, □□□□□ □□ □□ □□.

D. □□□□□ □□ □□□ □□ □□. □□□□□□ □□□ □□□ □□□□ □□.

Answer: A (LEAVE A REPLY)

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

- * Skill and staff shortages: This factor refers to the lack of adequate or qualified human resources to perform the project tasks. This may result in delays, errors, rework, or low productivity.
- * Problems in defining the right requirements: This factor refers to the difficulties or ambiguities in eliciting, analyzing, specifying, validating, or managing the requirements of the project. This may result in misalignment, inconsistencies, gaps, or changes in the requirements, affecting the project scope and quality.
- * Contractual issues: This factor refers to the challenges or disputes that may arise from the contractual agreements between the project parties, such as clients, suppliers, vendors, or subcontractors. This may result in legal, financial, or ethical risks, affecting the project delivery and satisfaction.

The other options are not correct because they list factors that contribute to PRODUCT risks, not project risks.

Product risks are the uncertainties or threats that may affect the quality or functionality of the software product or system. Some of the factors that contribute to product risks are:

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

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

* in errors, faults, or malfunctions of the software product or system.

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

NEW QUESTION: 75

□□ □ □□ □□ □□□ □ □□ □□□ □□□□□?

A. MTBF(□□ □ □□ □□)□ □□ □□

B. □□□□□ □□□ □□

C. □□□□ □□ □□ □□ □□ □□

D. □□ □□□

Answer: ([SHOW ANSWER](#))

Static analysis tools are software tools that examine the source code of a program without executing it. They can detect various types of issues, such as syntax errors, coding standards violations, security vulnerabilities, and potential bugs¹². However, static analysis tools cannot identify issues that depend on the runtime behavior or performance of the program, such as very low MTBF (Mean Time Between failure)³. MTBF is a measure of the reliability of a system or component. It is calculated by dividing the total operating time by the number of failures. MTBF reflects how often a system or component fails during its expected lifetime. Static analysis tools cannot measure MTBF because they do not run the program or observe its failures. MTBF can only be estimated by dynamic testing, which involves executing the program under various conditions and collecting data on its failures⁴. Therefore, very low MTBF is an issue that cannot be identified by static analysis tools.

The other options, such as potentially endless loops, referencing a variable with an undefined value, and security vulnerabilities, are issues that can be identified by static analysis tools. Static analysis tools can detect potentially endless loops by analyzing the control flow and data flow of the program and checking for conditions that may never become false⁵. Static analysis tools can detect referencing a variable with an undefined value by checking the scope and initialization of variables and reporting any use of uninitialized variables⁶. Static analysis tools can detect security vulnerabilities by checking for common patterns of insecure code, such as buffer overflows, SQL injections, cross-site scripting, and weak encryption. References = What Is Static Analysis? Static Code Analysis Tools - Perforce Software, How Static Code Analysis Works | Perforce, Static Code Analysis: Techniques, Top 5 Benefits & 3 Challenges, What is MTBF? Mean Time Between Failures Explained | Perforce, Static analysis tools - Software Testing MCQs - CareerRide, ISTQB_Chapter3 | Quizizz, [Static Code Analysis for Security Vulnerabilities | Perforce].

NEW QUESTION: 76

□□ □ □□□ □□□□ □□ □ □□□ □□ □□□□□?

- A. Exploratory testing is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented.
- B. Exploratory testing is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented.
- C. Exploratory testing is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented.
- D. Exploratory testing is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented. It is a form of testing that is not planned and is not documented.

Answer: D (LEAVE A REPLY)

Exploratory testing involves simultaneous test design and execution and is guided by a test charter, which outlines what needs to be tested, how it should be tested, and what to look for. This technique is typically conducted within predefined time periods, known as time-boxing, which allows testers to explore a system, understand its functionalities, and identify potential issues without detailed documentation or prior test case planning. The key aspects of exploratory testing include flexibility, adaptability, and the ability to respond to system behavior during testing. References:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 4.4.2.

ISTQB-CTFL-KR is a comprehensive resource for ISTQB Certified Tester Foundation Level (CTFL) exam preparation. It includes a large number of questions and answers, as well as a detailed explanation of the ISTQB-CTFL-KR syllabus. DumpTop is a leading provider of ISTQB-CTFL-KR questions and answers, and is known for its high quality and accuracy. DumpTop ISTQB-CTFL-KR questions and answers are available in PDF format, and can be downloaded for free. For more information, please visit <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 77

ISTQB defines four types of testing: static testing, dynamic testing, V-model testing, and SDLC testing.

- Static testing is a type of testing that is performed on the code before it is executed. It includes code reviews, code walkthroughs, and code inspections.
 - Dynamic testing is a type of testing that is performed on the code after it has been executed. It includes unit testing, integration testing, and system testing.
 - V-model testing is a type of testing that is performed on the code and the requirements. It includes code reviews, code walkthroughs, and code inspections.
 - SDLC testing is a type of testing that is performed on the code and the requirements. It includes code reviews, code walkthroughs, and code inspections.
- A. Static testing is a type of testing that is performed on the code before it is executed. It includes code reviews, code walkthroughs, and code inspections.
- B. Dynamic testing is a type of testing that is performed on the code after it has been executed. It includes unit testing, integration testing, and system testing.
- C. V-model testing is a type of testing that is performed on the code and the requirements. It includes code reviews, code walkthroughs, and code inspections.
- D. SDLC testing is a type of testing that is performed on the code and the requirements. It includes code reviews, code walkthroughs, and code inspections.

Answer: D (LEAVE A REPLY)

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: 78

□□ □□ □□□ □□ □ □□□ □□□□ □□□ □□□ □□□ □□□□□□: "□□□ □□", "□□ □□□□□□" □ "□□ □□□". □□□□ □□□ □□□ □□□ □□□□□ □□□ □□□ □□□ □□ □□□ □□□□□:

- A. □□□□□ □□□ □□□ □□□□ □□□□ □□□ □□□ □□□□□ □□□ □ □□□□ □□ □□ □□ □□□□□□ □ □□□.
- B. □□□□□ □□□ □□□ □□□□, □□□ □□□ □□□□□ □□□ □□ □ □□□□ "□□ □ □□" □□ □□□ □□□□ □ □□□□.
- C. □□□□□ □□□ □□□ □□□□, □□□ □□□ □□□□ □□ □ □□□□ "□□□ □□□ □□□"□ □□□ □□□□ □ □□□□.
- D. □□□□□ □□□ □□□ □□□□ □□□ □□□ □□□□□ □ □□□□□ □□□ □□□□ □□□□.
- □ □□□ □□□ □□□ "□□ □□□"

Answer: B ([LEAVE A REPLY](#))

The features of the test object to be tested and those excluded from the testing represent information that is usually included in a test plan and, in the given test plan, it is more likely to be specified within "Test Scope" rather than in the other two sections mentioned. The test scope defines the boundaries and limitations of the testing activities, such as the test items, the features to be tested, the features not to be tested, the test objectives, the test environment, the test resources, the test assumptions, the test risks, etc. The test scope helps to establish a common

understanding of what is included and excluded from the testing, and to avoid ambiguity, confusion, or misunderstanding among the stakeholders. The other two sections, "Testing Communication" and "Stakeholders", are also important parts of a test plan, but they do not directly address the features of the test object. The testing communication describes the methods, frequency, and responsibilities for the communication and reporting of the testing progress, status, issues, and results. The stakeholders identify the roles and responsibilities of the people involved in or affected by the testing activities, such as the test manager, the test team, the project manager, the developers, the customers, the users, etc. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning1

* ISTQB Glossary of Testing Terms v4.0, Test Plan, Test Scope2

NEW QUESTION: 79

□□ □ □□ □□□ □□□□□?

1. □□□ □□ □□□□ □□□□ □□ □□□ □□ □□□ □□□ □ □□□ □□□.
2. □□ □□□□ □□ □□ □ □□ □□□ □ □□ □□ □□ □ □□□□.
3. □□ □□□□ □□ □□ □□ □□□ □□□ □□□ □ □□□□.
4. □□ □ □□□ □□□□ □□ □□□ □□□ □□□□ □□ □□□□ □□□ □□□. □□□ □ □□□□:

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

Answer: C ([LEAVE A REPLY](#))

The statement "Early and frequent feedback helps to avoid requirements misunderstanding" is true. Early feedback from stakeholders, through reviews and other static testing techniques, helps clarify requirements and ensures that any misunderstandings are addressed promptly. This practice aligns with Agile principles and contributes to developing software that meets user needs more accurately.

References: ISTQB CTFL Syllabus, Section 2.1.1, "The Influence of Development Models on Testing" and Section 3.2.1, "The Advantages of Early Feedback."

NEW QUESTION: 80

□□ □□ □ □□ □□□□□ □□ □□□□□□□ □□□□ □□ □□□ □□□ □□□□ □□ □□□□□?

- A. □ □□□ □□□□ □□□□□ □□ □□□□□□ □□□ □□ □□□□ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□□ □□□ □□□.
- B. □□ □□□ □□□ □□ □□□ □□ □ □□□ □□□ □□□□ □□ □□□ □□ □□□ □□ □□ □□□ □ □□ □□ □□ □□□□□ □□□.
- C. □□□ □□□ □□□ □□□□ □□ □□□□□ □□□□ □□□ □□ □□□□□.

D. □□ □□□□ □□□□□□ □□ □□ □□□□□ □□ □ □□□□ □□ □□□□□ □□□ □□□□□ □□□.

Answer: ([SHOW ANSWER](#))

Good testing practice dictates that each test level (e.g., unit testing, integration testing, system testing) should have distinct test objectives aligned with the phase of the software development lifecycle it addresses. This ensures that testing is effective and relevant at each stage. According to the ISTQB CTFL Syllabus v4.0, establishing clear test objectives that are consistent with the development phase helps in achieving specific goals and improving the overall quality of the software product.

NEW QUESTION: 81

□□ □ □□ □□□ □□□□ □□ □□□ □□□□□?

- A. □□□□ □□□ □□□ □□□ □□ □□ □□□□ □□□□□.
- B. □□□ □□□ □□□□□ □□□ □□□ □□□ □ □□ □□□□ □□□.
- C. □□□□ □□□□ □□□□ □□□□ □□ □□□□.
- D. □□□□ □□□ □□□□□□ □□ □□□□□ □□□ □□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

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: 82

□□ □ □□ □□□□□□□ □□ □□□ □□□□ □□ □□□□□?

- A. □□ □□□□ □□□ □□□ □□□□□ □□□□ □□□ □ □□□
- B. □□ □□ □□ □□ □□ □□ □□□ □□□□□ □□□□ □ □□□□ □□ □□□
- C. □□□ □□□ □□□□ □□□□ □□□□ □□ □□□□ □□
- D. □□□ □□□□□ □□ □□□□□(□□□ □□□ □□) □□□□ □□□ □ □□

Answer: A ([LEAVE A REPLY](#))

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: 83

□□□ □□ □□□□□ □□ □□□ □□□□ □□ □□□□□.

- a. □□
- b. □□ □□
- c. □□ □□□□□□ □ □□
- d. □□ □ □□
- 1. □□ □□□ □□
- 2. □□□ □□ □□
- 3. □□□ □□□□□ □□ □□
- 4. □□□ □□□□ □□
- 5. □□ □□□ □ □□ □□ □□
- 6. □□ □□ □□

□□ □ □□ □□□□□ □□ □□□ □□ □□ □□□ □□ □ □□□□□ □□ □□□□□?

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

Answer: A ([LEAVE A REPLY](#))

Matching the main review process activities with the specific review activities, we see that planning includes estimating effort and timeframe (2) and selecting people to participate (4). Initiating a review involves distributing work products and other material (5). Issue communication

and analysis includes evaluating the review findings (6). Fixing and reporting would entail creating defect reports (1) and recording the updated status of defects (3).References:ISTQB Certified Tester Foundation Level Syllabus v4.0, Section 3.2 "Review Process".

NEW QUESTION: 84

□□ □□□□ □□□□ □ □□ □□ □□□□ □□□□ □□ □□ □□□ □□ □□□□□?

A. □□ □□□ □□□□ □□

B. IoT □□□□□□□□ □□□ □□□□ □□ □□ □□

C. □□ □□□ □□□□□□□□ □□ □□□ □□□□ □□□ □□□□□□□□.

D. □□□ □□ □□□ □□□□ □□□□ □□ □□□□□ □□

Answer: D ([LEAVE A REPLY](#))

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

Option A: "Completion of architecture of the bank system" is not a typical scenario for maintenance testing, as it describes a design phase rather than an operational change.

Option B: "Release of the early draft of the low level project design of an IoT application" is again not suitable for maintenance testing, as it refers to the design phase.

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

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

Therefore, the correct answer is D6†source.

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 85

□□ □ □□□ □□ □□□□ □□□ □ □□ □□ □□□□□?

I □□ □□ □□ □□□ □□□ □□ □□.

II. □□□□ □□□ □□□□ □□ □ □□.

III. □□, □□ □□ □□ □□□ □□ □□□ □□.

IV □□ □□□ □□ □□ □□ □□ □ □□ □□.

A. I, III, IV

B. I, II, III

C. III, IV

D. II, III, IV

Answer: ([SHOW ANSWER](#)**)**

An exit criterion is a condition that defines when a test activity has been completed or when a test phase can be concluded. An exit criterion can be based on various factors, such as:

* I) Estimates of defect density or reliability measures. These are quantitative measures that indicate how many defects are present in the software product or how likely it is to fail under certain conditions.

These can be used as exit criteria to ensure that the software product meets a certain level of quality or performance before moving to the next phase or releasing it to the customer.

* III) Accuracy measures, such as code coverage, functionality coverage or risk coverage. These are quantitative measures that indicate how much of the software product has been tested in terms of its code, functionality or risk. These can be used as exit criteria to ensure that the test suite is adequate or complete before moving to the next phase or releasing it to the customer.

* IV) Residual risks, such as lack of code coverage in certain areas, unresolved defects or unknown factors. These are qualitative measures that indicate the remaining risks or uncertainties associated with the software product after testing. These can be used as exit criteria to ensure that the residual risks are acceptable or manageable before moving to the next phase or releasing it to the customer. The following factor is not a valid exit criterion:

* II) The completion and publication of an exhaustive Test Report. This is not a valid exit criterion, as it does not reflect the quality or completeness of the testing process or product. A test report is a document that summarizes the results and outcomes of a test activity or phase. A test report can be used as an input for deciding whether to exit a test activity or phase, but it is not a condition that defines when to exit.

Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 2, page 13; Chapter 6, page 58-59.

NEW QUESTION: 86

□□ □ □□ □□ □□ □□ □□ □□ □□□□□?

A. □□ □□ □□□□□□ □□ □□□ □□□ □□□□□ □□□□□.

B. □□ □□□□ □□□ □□ □□ □□□ □□ □□ □□ □□□□□.

C. □□□□□ ell □□ □ □□ state□ □□□□ □□□□ □□□ □□ □□□□□□.

D. □□ □□ □□ □□□ □□ □□ □□□□□ □□□□□.

Answer: D (LEAVE A REPLY)

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: 87

"□□□ □□□ □□ □□□ □□ □□□□ □□□□ □□, □□ □□□□ □□□ □□□ □□□□ □□□."□□ □□□ □□□ □□□□□ □□ □□ □ □□ □□ □□□ □□□□?

- A. □□□ □□ □□.
- B. □□□ □□ □□.
- C. □□□ □□ □□.
- D. □□ □□ □

Answer: D ([LEAVE A REPLY](#))

The principle of early testing is applicable to all recognized software development models, including sequential, iterative, and incremental models. Starting test activities early in the lifecycle helps in identifying and addressing defects as soon as possible, which can save time and costs by preventing defects from propagating to later stages of development. This proactive approach enhances the overall quality and efficiency of the software development process. Reference: ISTQB CTFL Syllabus V4.0, Section 1.3

NEW QUESTION: 88

□□ □□ □□ □□□ □□ □□□□ □□□□ □□□ □□□□□ □□ □□□ □□□□ □ □□ □□□ □□□□□?

- A. □□□□□ □□
- B. □□ □□ □□.
- C. □□ □□ □□.
- D. □□□ □□□ □□ □□

Answer: C ([LEAVE A REPLY](#))

A static analysis tool is best suited for determining source code compliance with coding standards. These tools analyze the code without executing it and can check for adherence to coding standards, syntax errors, and other static properties of the code. The ISTQB CTFL syllabus emphasizes the role of static analysis tools in verifying that code meets predefined standards and guidelines.

References: ISTQB CTFL Syllabus, Section on static testing and tools.

NEW QUESTION: 89

□□□ □□□ □□□ □□ □□□□□ □□□□ □□□ □□□□.

□□□□ □□

□□ □□□ □□ □□ □ □□□□□ □□□, □□□ □□□□ □□ □ □□□□ □□ □□□□ □?

- 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: (SHOW ANSWER)

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: 90

□□ □ □□□ □□□ □□□ □□□□ □ □□□□ □□□ □ □□ □□□ □□□□□?

- A. □□ □□□□ □□ □□□□□ □□□ □□ □□□ □□□ □□□□□□□.
- B. □□ □□□□ □□ □□□ □□ □□□ □□□□□□□.
- C. □□ □□□□ □□ □□□□ □□□□ □□□□ □□□
- D. □□□ □□□□□ □□□□ □ □□□ □□□ □□□□□□□.

Answer: D ([LEAVE A REPLY](#))

One of the common risks associated with test automation tools is the underestimation of the effort required to maintain test scripts. Test scripts can become outdated or broken due to changes in the application, requiring significant effort to update and maintain them. This risk is highlighted in the ISTQB CTFL syllabus under the discussion of the benefits and risks of test automation.

References: ISTQB CTFL Syllabus, Section on test tools and automation.

NEW QUESTION: 91

□□ □ □□□ □□ □□□ □□ □□□ □□ □ □□□□ □□ □□□□□?

1. □□□ □□□ □□ □□, □□ □□ □ □□□ □□□□ □□□ □ □□ □□□ □□□ □□□ □□□.
 2. □□□ □□□□ □□□□□ □□□□□ □□□ □□□ □ □□ □□□ □□□ □□□□ □□ □.
 3. □□□ □□□□ □□ □□□□ □□□ □□□ □ □□□□.
 4. □□□ □□□ □□□ □□ □□□□ □□□ □□□ □□□ □□ □□□□□.
 5. □□□□ □□□, □□□ □ □□ □□ □□□□ □□□ □□□□ □□□.Black - □□□□ □□ □□□ White * □□□□□ □□□ □□ Experience - □□ □□ □□□ □□
- A. □□ -1,3.4 □□□ - 2 □□□ - 5
 - B. □□□ - 2, 3 □□ -1 □□□ - 4, 5
 - C. □□□ -1,2 □□ - 3, 4 □□□ - 5
 - D. □□□ - 2, 3 □□ -1,5 □□□ - 4

Answer: ([SHOW ANSWER](#))

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

References:

- * ISTQB Certified Tester Foundation Level Syllabus v4.0: ISTQB CTFL Syllabus v4.0 PDF

NEW QUESTION: 93

□□ □ □□□□ □□□ □□□ □□ □□□□□?

- A. □□□□ □□□□□□.
- B. □□□ □□□□ □□□□ □□□□□.
- C. □□□ □□□ □□□□□.
- D. □□□□ □□□□ □□□ □□□ □□□□□.

Answer: D ([LEAVE A REPLY](#))

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

NEW QUESTION: 94

□□ □□□ □□ □□ □□ □ □□□ □□ □□□□□?

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

Answer: D ([LEAVE A REPLY](#))

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: 95

□□ □ □□ □□□ □□ □□ □□□□□?

- A. □□□□□ □□□□ □□ □□

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: 97

□□ □ □□ □□□ □□□□□?

- A. □□ □□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□ □□□ □□□□ □□□ □ □□□ □□□ □□□□.
- B. □□□ □□□□□ □□□ □□ □□□ □□□□ □□ □□ □□□ □□ □□□□ □□□□□.
- C. □□ □□□□ □□□□ □□□□ □□□□□ □□□ □□□□ □□□□ □□□□□ □ □□ □□□ □□□ □□□□ □□□ □□□.
- D. □□□□□ □□□□ □□□□ □ □□□ □ □□ □□□ □□□ ISO/IEC 25010 □□□ □□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

Non-functional testing includes testing of both technical and non-technical quality characteristics. Non-functional testing is the process of testing the quality attributes of a system, such as performance, usability, security, reliability, etc. Non-functional testing can be applied at any test level and can use both black-box and white-box test techniques. Non-functional testing can cover both technical aspects, such as response time, throughput, resource consumption, etc., and non-technical aspects, such as user satisfaction, accessibility, compliance, etc. Therefore, option B is the correct answer.

References: ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 1.3.1, page 13; ISTQB Glossary v4.02, page 40.

NEW QUESTION: 98

□□□ ATM(□□□ □□□□□) □□□□□ □□□ □□ □□ □□□ □□□□ □□□□ □□□ □. □□ 1□ 4□ □□ □ □□ □□□ □□□ □□ □□□□□□, □□□ □□ □□□□.

□□ 1 □□

□□□ □□ □□ □□□ □□□ □□□□□□.

□□ 2 □□

□□ 1□ 4□ □□ □□ □□□ □□□ □□□ □□□ □, □□ □□□ □□ □ □□ □ □□□ □□ □□ □□□□ □□□□ □□□ □ □□□□?

- A. DT1, DT4
- B. DT3, DT4
- C. DT2, DT3

D. DT1.DT2

Answer: C (LEAVE A REPLY)

Decision table testing is used to analyze combinations of inputs to determine the appropriate outputs, often based on specific rules or conditions.

For the problem statement:

- * Rule 1: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = True, Correct PIN = True)
- * Outcome: Transaction = Approved
- * Rule 4: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = True, Correct PIN = False)
- * Outcome: Transaction = Declined

The additional test cases are:

- * DT1: (Withdrawal = Allowed, Balance = Insufficient, Fast Cash = True, Correct PIN = True)
- * Outcome: Transaction = Declined
- * DT2: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = False, Correct PIN = True)
- * Outcome: Transaction = Approved
- * DT3: (Withdrawal = Allowed, Balance = Insufficient, Fast Cash = True, Correct PIN = False)
- * Outcome: Transaction = Declined
- * DT4: (Withdrawal = Allowed, Balance = Sufficient, Fast Cash = False, Correct PIN = False)
- * Outcome: Transaction = Declined

From the given test cases, DT2 covers the scenario where Fast Cash is False, which is not covered in the initial cases. DT3 covers the case where Balance is Insufficient and PIN is incorrect.

Combining Rules 1 and 4 with DT2 and DT3 covers all the scenarios.

References:

- * Certified Tester Foundation Level v4.0
- * 10 Sample Exams ISTQB Foundation Level (CTFL) v4.0

NEW QUESTION: 99

□□□ □□□□□ 4~7□ □□□□ □□, □□ 1□ □□, □□□ 1□, □□□ □□□ 1□□ □□ 1□ □□ □□□□ □□□.

□□ □□□ □□ □□ □ □□ □□ □□□□ □□□ □□□ 2□ □□ □ □□□ □□□ □□□ □ □□□□? (□□: □□□ □□□ □□□□□□ □□)

- A. 1xA;aB11;Pq1ZZab;7iDD0a1x
- B. aB11;99rSp:5NnN10;7iDD0a1x
- C. 1xB: aB11: 99rSp: 5NnN10; 4NnN10T; 44ghWn19
- D. 1RhT;rSp53;3N3e10;8sBdby

Answer: D (LEAVE A REPLY)

The correct outcome of a two-value boundary value analysis applied to the password length is the set of test cases represented by option D. Boundary value analysis is a test design technique that focuses on the values at the boundaries of an equivalence partition, such as the minimum and maximum values, or the values just above and below the boundaries. A two-value boundary value analysis uses two values for each boundary, one representing the valid value and one

representing the invalid value. For example, if the valid range of values is from 4 to 7, then the two values for the lower boundary are 3 and 4, and the two values for the upper boundary are 7 and 8. The test cases in option D use these values for the password length, while also satisfying the other requirements of the password, such as containing at least one numeric character, one capital letter, and one lowercase letter. The test cases in option D are:

- * 1RhT: a 4-character password that is valid
 - * rSp53: a 5-character password that is valid
 - * 3N3e10: a 6-character password that is valid
 - * 8sBdbY: an 8-character password that is invalid
- The test cases in the other options are incorrect, because they either use values that are not at the boundaries of the password length, or they do not meet the other requirements of the password. For example, the test cases in option A are:

- * 1xA: a 3-character password that is invalid, but it does not contain a capital letter
- * aB11: a 4-character password that is valid
- * Pq1ZZab: a 7-character password that is valid
- * 7iDD0a1x: an 8-character password that is invalid

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.1, Black-box Test Design Techniques¹
- * ISTQB Glossary of Testing Terms v4.0, Boundary Value Analysis, Equivalence Partition²

NEW QUESTION: 100

□□ □ □□□ □□ □□□ □□ □□□□□?

- A. □□ □□□
- B. □□ □□
- C. □□□□□ □□□□ □□□ □□□ □□□□□.
- D. □□□□ □□

Answer: ([SHOW ANSWER](#))

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: 101

□□□ □ □□ □□□ □□□ □□□ □□□ □□ □□□ □□□. □□ □□□□□, □□ □□□ □ □□ □□ □□ □□ □□ □□□ □□□ □□ □□□ □□□ □□□ □□ □□□ □□□ □□ □□□ □□ □□ □□ □□ □□□ □□□□□□. □□□ □□□□ □□□□ □□ □□□ □□□□□□, □□ □□□□□□ □□ □□□□ □□ □□□□□□. □□ □□□□ □□□□□, □□ □□□□□ □□□ □□ □□□ □□□□□□. □□ □□□ □□□ □□ □□□□ □□□ □□ □□ □□□□ □□ □□ □□□ □□ □□□□ □□ □□□□ □□ □□□ □□□□□□□?

- A. □□□ □□□ □□□ □□□ □□□□ □□ □□□ □□□ □□□□ □□ □□□□ □□□ □□□.
- B. □ □□□ □□□□ □□ □□ □□□□ □□□ □□□□. □□□ □□ □□□□□ □□ □□ □ □□□ □□□□□□ □□ □□□□ □□ □□□ □□□.
- C. □□□ □□□ □□□ □□□□ □□□ □□□□ □□□ □□ □□□□ □□□□ □□ □□□ □□□ □□□□.
- D. □□ □□□□ □□□□ □ □□ □□□ □□□□□ □□□ □□□ □□□ □ □□ □□□□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□□ □□□.

Answer: A ([LEAVE A REPLY](#))

As a functional tester, you should open a defect report providing detailed information on which devices and by running which tests you observed the issue. A defect report is a document that records the occurrence, nature, and status of a defect detected during testing, and provides information for further investigation and resolution.

A defect report should include relevant information such as the defect summary, the defect description, the defect severity, the defect priority, the defect status, the defect origin, the defect category, the defect reproduction steps, the defect screenshots, the defect attachments, etc.

Opening a defect report is a good practice for any tester who finds a defect in the software system, regardless of the type or level of testing performed. The other options are not recommended, because:

* The issue is related to performance efficiency, not functionality, but that does not mean that as a functional tester, you should not open any defect report as all the functional tests passed.

Performance efficiency is a quality characteristic that measures how well the software system performs its functions under stated conditions, such as the response time, the resource utilization, the throughput, etc.

Performance efficiency is an important aspect of the user experience, especially for web applications that run on different devices and networks. Even if the functional tests passed, meaning that the software system met the functional requirements, the performance issue observed on some devices could still affect the user satisfaction, the usability, the reliability, and the security of the software system.

Therefore, as a functional tester, you have the responsibility to report the performance issue as a defect, and provide as much information as possible to help the developers or the performance testers to investigate and resolve it.

NEW QUESTION: 102

Agile □□□ □□□□ □□ □□ □□□ □□□□ □□□□□. □□□ □□□ □□□ □□□ □ □□□ □□□ □□ □□□ □□□ □□□ □□□ □ □□□□.

□□ □□ □□ □ □ □□□ □□□ □ □□□□?

i. □□□□ □□□ □□ □□ □□□ □□□□□.

ii. □□□□□ □□□ □□ □□□□ □□ □□□ □□□ □□□□□ □□□□□.

iii. □□□□ □□ □□□ □□ □□ □□□□ □□□□ □□ □□□ □□□ □□□ □ □□□□.

iv. □□□□ □□ □□□□ □□□□ □□□ □□□□ □□□ □□ □□□□.

A. ii □ iii

B. i □ iii

C. i □ ii

D. ii □ iv

Answer: D ([LEAVE A REPLY](#))

In Agile teams, close collaboration among testers and other team members can lead to organizational risks such as: ii. Close interaction with developers causes a loss of the appropriate tester mindset. iv. Testers, once they have acquired technical development or business skills, leave the testing team.

These risks highlight the potential issues of diminished testing perspective and team turnover when testers integrate closely with developers and other roles.

NEW QUESTION: 103

□□□□□□ □ □□ □□□ □□□□ □□ □□□ □□□ □□□□□.

- E; □□□ □□ □□

- P : □□□ □

-T □□□ □□

P□ □□ □□ 15□ 350 □□□ □□ □□ □□ □ □ □□□□.

□□ □□ □ P □□□ □□ □□ □□ □□□ □□□ □□□□□?

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

Answer: B ([LEAVE A REPLY](#))

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

NEW QUESTION: 104

□□ □□□□ □□ □□□□ □□ □□ □□ □ □□ □□ □□□□□?

- A. □□□□ □□ □□□□ □□□ □ □□ □□□ □□□□ □□ □□ □□□□□ □□ □□□□ □ □□□□□.
- B. □□ □□□□ □□ □□□ □□ □□□□ □□□ □ □□ □□, □□ □□□□ □□ □□□□ □□ □□□□ □□□ □ □□□□.
- C. □□ □□□□ □□ □□□□ □□ □□□ □□□□□□, □□ □□□□ □□ □□□ □□□□ □ □□□□□ □ □□ □□□□□.
- D. □□ □□□□ □□□□□□ □□ □□ □□ □□□ □ □□□□ □□ □□□□ □□ □□ □□ □□ □□□□ □□□ □ □□□□.

Answer: A ([LEAVE A REPLY](#))

Static testing, such as code reviews and static analysis, is particularly effective at identifying issues related to code structure and modularization. These techniques allow for the inspection of the code without executing it, making it easier to spot problems related to how the code is organized. Dynamic testing, on the other hand, focuses on the execution of code and is better suited for identifying runtime issues but does not easily reveal structural problems. The ISTQB CTFL Syllabus v4.0 highlights the strengths of static testing in uncovering such structural issue

NEW QUESTION: 105

□□ □ □□ □□ □□□□ □□ □□□ □□ □□ □□ □□ □□□□ □□□ □ □□□□?

- i. □□□ □□ □ □□□ □□ □□ □□.
- ii. □□□ □□□ □□ □□.
- iii. □□□□□. □□□ □□ □□□ □□□.
- iv. □□□ □□ □□□ □□□ □□□.
- v. □□ □□ □.
- □□ □□□□□:
- A. i□ ii□.

- B. i ☐ iii ☐ ☐.
- C. I, ii, iv ☐ ☐.
- D. v ☐

Answer: D ([LEAVE A REPLY](#))

Metrics can be collected during and at the end of testing activities to assess various aspects including progress against the planned schedule and budget, the current quality of the test object, the adequacy of the test approach, and the effectiveness of the test activities with respect to the objectives. Collecting these metrics helps in understanding the overall performance and quality of the testing process.

NEW QUESTION: 106

□□ □□□ □□□ □□□□□□ □□□□ □□□□□ □□□□ □□□□ □□□□ □□□□
 □□ □□□□□ □□□□ □□□□.

□□□□□ □□ □□□ □□□ □□□□ □□□□ □ □□ □□□ □□ □□□ □□□□□?

- A. □□□□
- B. □□□ □□
- C. □□ □□
- D. □□ □□

Answer: C ([LEAVE A REPLY](#))

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.

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □ □ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF Special Discount: KrDump**)

NEW QUESTION: 107

ISTOB □□□□ □□ □□□ □□□ □□ □□□□□. "□□ □□ □□□ □□□ □□□□ □□□ □□□□ □ □□□ □ □□□ □□ □□ □□. □□ □ □□ □□ □□ □ □□□ □□ □□ □□□ □□□?

- A. □□□□ □□
- B. □□ □□
- C. □□ □□□
- D. □□ □□□

Answer: ([SHOW ANSWER](#))

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

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

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

* Functional Testing (C) and Performance Testing (D) are types of dynamic testing, which are actually Quality Control activities rather than Quality Assurance. They are concerned with the identification of defects in the product, not with the processes to prevent defects.

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

NEW QUESTION: 108

□□□ □□ □□□ □□□□ □□□□. "□□□□ □□□□□ □□ □□□□ □□ □□□ □□□ □ □□□ 1□ □□□ □□ □□□ □□□□□ □□ □□□□." □□ □ □□ □□ □□ □□□ □□ □□□ □□□ □□□ □ □□□□?

- i. □□□ □□ □□□ □□□□ □□□ □□ □□□ □□□ □, □□□ □□ □□□ □□□□□. □ □□ □□ □□□ 1□ □□□ □□ □□□□ □□□□□.
- ii. □□□ □□ □□□ □□□□ □□ □□□ □□□ □ □□ □□□ □□□□□. □□□ □□ □□ □ 1□ □□□ □□ □□□□ □□□□□.
- iii. □□□ □□ □□□□ □□□□ Enter□ □□□ □□□ □□ □□□ □□□ □, □□□ □□ □ □□ □□□□□. □□□ □□ □□□ 1□ □□□ □□ □□□□ □□□□□.
- 4. □□ □□□ □□□□, □□ □□□ □□□□, □□□ □□, □□ □□ □ □□ □□□ □□□□.
- v. □ □□ □□□ □□□□, □□□ □□ □□□ □□□□, □□□ □□ □□□ □□□□ □□□ □ □□□ □□ □□□□ □□□□□.

□□□ □□ □□□□□:

- A. i, ii □ v
- B. 4
- C. i □ iii

D. ☐

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 109

Which of the following is a type of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc.?

- A. Static testing is a type of testing that involves the analysis of software work products, such as requirements, design, code, or test cases, without executing them.
- B. Dynamic testing is a type of testing that involves the execution of software work products, such as code or test cases, using inputs and verifying outputs.
- C. Both static testing and dynamic testing can be applied to different test levels and test types, and can use different test techniques and tools, to evaluate the non-functional characteristics of the software product.
- D. Both static testing and dynamic testing are both types of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc.

Answer: D ([LEAVE A REPLY](#))

This answer is correct because static testing and dynamic testing are both types of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc. Static testing is a type of testing that involves the analysis of software work products, such as requirements, design, code, or test cases, without executing them. Dynamic testing is a type of testing that involves the execution of software work products, such as code or test cases, using inputs and verifying outputs. Both static testing and dynamic testing can be applied to different test levels and test types, and can use different test techniques and tools, to evaluate the non-functional characteristics of the software product. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.2.1.1, Section 2.2.1.2

NEW QUESTION: 110

Which of the following is a type of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc.?

- Static testing is a type of testing that involves the analysis of software work products, such as requirements, design, code, or test cases, without executing them.
 - Dynamic testing is a type of testing that involves the execution of software work products, such as code or test cases, using inputs and verifying outputs.
 - Both static testing and dynamic testing can be applied to different test levels and test types, and can use different test techniques and tools, to evaluate the non-functional characteristics of the software product.
 - Both static testing and dynamic testing are both types of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc.
- A. Static testing is a type of testing that involves the analysis of software work products, such as requirements, design, code, or test cases, without executing them.
- B. Dynamic testing is a type of testing that involves the execution of software work products, such as code or test cases, using inputs and verifying outputs.
- C. Both static testing and dynamic testing can be applied to different test levels and test types, and can use different test techniques and tools, to evaluate the non-functional characteristics of the software product.
- D. Both static testing and dynamic testing are both types of testing that can be used to highlight issues associated with non-functional characteristics, such as usability, performance, security, reliability, etc.

Answer: B ([LEAVE A REPLY](#))

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

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

Boundary value analysis is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Boundary value analysis is a technique that tests boundary values between partitions of equivalent data. Boundary values are values at the edge of an equivalence partition or at the smallest incremental distance on either side of an edge.

Boundary value analysis can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify boundary values for the input amount, such as the minimum and maximum amount allowed by the system or the user's account. However, boundary value analysis may not cover all possible combinations of input conditions, as some conditions may not have boundary values or may not be related to input values.

Equivalence class partitioning is not a technique that can cover all possible combinations of input conditions for withdrawing money from an ATM. Equivalence class partitioning is a technique that divides the input data and output results of a software component into partitions of equivalent data. Each partition should contain data that is treated in the same way by the component.

Equivalence class partitioning can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify equivalence partitions for the input amount, such as valid amount (within the range allowed by the system and the user's account) and invalid amount (outside the range allowed by the system or the user's account). However, equivalence class partitioning may not cover all possible combinations of input conditions, as some conditions may not be related to input values or may

have more than two partitions. Verified References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 4, page 34-46.

NEW QUESTION: 111

□□ □ □□□ □□ □□□□ □□□□□ □□ □□□ □□□□□?

- I. □□ □□ □□
- II. □□ □□□ □□ □□□ □ □
- III □□ □□□ □□
- IV. □□ □□
- V. □□□ □□□, □□□ □□□, □□□ □□□□ □

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

Answer: D ([LEAVE A REPLY](#))

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: 112

□□□ □□□ □□□□□:

- * □□□ □□ □□□ □□□□ "□□"□□□ □□□.
- * □□□ □□□ □□ □□□ □□□ □ □□□□.
- * □□□□ □□□ □□□□ □□□□ □□□ □□□□ □□ □□□.
- * □□□□ □□□ □□□□ □□□ □□□ □ □□□□.
- * □□□ □□ □□□ □□ □□□□□.

A. 2, 4

B. □, □

C. 4, 5

D. 2, 3, 4

Answer: (SHOW ANSWER)

* The question is about marking the correct sentences among the given statements related to defects, failures, and mistakes. According to the ISTQB glossary, the definitions of these terms are¹:

* Defect: A flaw in a component or system that can cause the component or system to fail to perform its required function, e.g. an incorrect statement or data definition. A defect, if encountered during execution, may cause a failure of the component or system.

* Failure: An event in which a component or system does not perform a required function within specified limits.

* Mistake: A human action that produces an incorrect result.

* Therefore, out of the five given statements, only two are correct, namely:

* A human mistake may produce a defect: This is true, as a mistake is a source or cause of a defect, e.g. a programmer may make a mistake in writing a code statement, which results in a defect in the software component.

* When a defect exists in a system it may result in a failure: This is true, as a defect is a potential or actual cause of a failure, e.g. a defect in the software component may cause the system to fail to perform a required function when the defect is encountered during execution.

* The other three statements are incorrect, namely:

* Defects are a result of environmental conditions and are also referred to as "Failures": This is false, as defects are not a result of environmental conditions, but of mistakes or other factors, and defects are not the same as failures, but rather the causes of failures.

* A system will totally fail to operate correctly when a failure exists in it: This is false, as a system may not necessarily fail completely or stop operating when a failure occurs, but may continue to operate with reduced functionality or performance, or with incorrect results.

* Defects occur only as a result of technology changes: This is false, as defects can occur due to various reasons, not only technology changes, such as human mistakes, design flaws, requirement changes, hardware failures, etc.

References:

* 1: ISTQB Glossary of Testing Terms 4.0, 2023, available at (ISTQB) and (ASTQB).

NEW QUESTION: 113

□□ □□ □□, □ "□□ □□"□ Agile □□□□ □□□□□ □□□□□?

- A. □□.
- B. □□□□.
- C. □□ □□.
- D. □□□ □□.

Answer: ([SHOW ANSWER](#))

In Agile development, an informal review, often referred to as a "buddy check," is a common review type.

Informal reviews are unstructured and involve a pair of colleagues reviewing each other's work to identify defects early and provide immediate feedback. This type of review is less formal than inspections or walkthroughs and is particularly suitable for Agile environments where rapid feedback and flexibility are essential.

References: ISTQB CTFL Syllabus, Section 3.2.4, "Types of Reviews" and Section 2.1.4, "Agile Testing Practices."

NEW QUESTION: 114

□□□ □□□□□ □□ □□ □□ □ □□ □□ □□□□□?

- A. □□□ □□□□□ □ □□□ □□ □□□□ □□ □□□ □□□ □□□□□□.
- B. □□□ □□□□□ □□□ □□□□ □□ □□ □□ □□□ □□□□ □□□ □□□□ □□□ □□ □□□□□.
- C. □□□ □□□□□ □□□ □□□□ □□ □□ □□ □□□ □□□□ □□ □□□ □□□□□ □□□ □□□□.
- D. □□□ □□□□□ □□□ □□□□ □□ □□ □□ □□□ □□□□ □ □□□□ □□□ □□ □□.

Answer: B ([LEAVE A REPLY](#))

The test pyramid concept suggests that there should be more low-level tests (unit tests) and fewer high-level tests (end-to-end tests).

* As we move higher up the pyramid (e.g., from unit tests to integration tests to end-to-end tests), each test covers more production code.

* Higher-level tests (like end-to-end) validate larger parts of the application, including multiple units and their interactions.

This aligns with the principle that higher-level tests provide broader coverage but are fewer in number and more expensive to run and maintain.

NEW QUESTION: 115

□□ □ □□□□□ □□ □□ □□(SDLC)□ □□□□ □□□□ □□ □□□ □□□□ □□ □□ □□□□□?

- A. □□□□ SDLC□ □□ □□□ □□□□ □□□ □□□ □□□□ □□□.
- B. □□□□ SDLC □□□□ □□□□ □□ □□□ □□ □□□ □□□ □□□ □□□ □□□□ □□□.
- C. □□□□ □□□ □□□□ □□□□ □□□.

D. □□□□ □□□ □□□□□□, □□□□ □□□□ □□□□ □□□□□□.

Answer: (SHOW ANSWER)

Involving testers only during the testing phase is incorrect as per the ISTQB CTFL syllabus. Effective involvement of testers is crucial throughout the entire software development lifecycle (SDLC). This includes early stages such as requirement analysis and design, which allows testers to understand the design decisions and detect defects early. Early involvement helps in better understanding the project and ensures that quality is built into the product from the beginning. Furthermore, the ISTQB syllabus emphasizes the importance of testers contributing to all activities in the SDLC, including design discussions, to enhance defect detection and prevention. References: ISTQB CTFL Syllabus, Section 2.1.1, "The Influence of Development Models on Testing" and Section 1.1.1, "Test Objectives."

NEW QUESTION: 116

□□□□ □□□□ □□ □□□□ □□□□□. □ □□□ □□□□ □□□ □□□ □□□□ □□ □.

A. □□□ □□ □□□□

B. □□□□ □□

C. □□ □□□□□□

D. □□□ □□□

Answer: A (LEAVE A REPLY)

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

NEW QUESTION: 117

□□□ □□ □ □□ □□□ □□□□□.

1. □□□ □□□ □□ □□□□□ □□□□□□□□.

2. □□□ □□□ □□□ □□□□ □□□□ □□□□□□□□.

3. □□□ □□□ □□ □□ □□□ □□ □□ □□□□.

4. □□ □□□ □□□ □□□□□□ □□ □□□□□□.

5. □□□ □□□□□ □□□ □□ □□□ □□□□ □□□□.

6. □□□ □□□ □□□□□ □□□□ □□□□ □ □□ □□□□ □□ □□ □□□ □□□□□□.

□□ □ □□ □□ □□ □ □□ □□□□ □□ □ □□□□□□?

A. □□□□ - 2, 4, 5 □□□□ -1, 3, 4

B. □□□□ - 2, 4 □□□□ - 2, 4, 5, 6

C. □□ □□ -1,3,6 □□ □□ - 2, 5. 6

D. □□ □□ -1,3,56 □□ □□ -1,3

Answer: A ([LEAVE A REPLY](#))

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

* Entry Criteria:

* These are conditions that should be met before testing begins.

* Examples:

* 2. The test automation tool has been installed and properly configured

* 4. The performance test environment has been set-up and is available

* 5. The user stories have proper acceptance criteria defined

* Exit Criteria:

* These are conditions that should be met before testing concludes.

* Examples:

* 1. A defined level of code coverage has been achieved

* 3. The number of unresolved defects is within the predefined limit

* 6. The testing budget (ISTQB not-for-profit association) spent and the project sponsor bears the risk of not testing any further According to the ISTQB Certified Tester Foundation Level (CTFL) v4.0 syllabus, these criteria help in effectively managing the testing process6†source.

References:

* Certified Tester Foundation Level v4.0

* ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 118

□□ □□ □ □□ □□□ □□□ □□□□□?

A. □□ 300□□ □□ □□□ □□□□□ □□ □□□□ □□□□ □□ □ □□□ □.

B. □□□ □□□ □□□ □□ □□ □□

C. □□□ □□□ □□□ □□□□ □□ □□□ □□ □□□□□□ □□ □□□ □□□ □□□ □ □□□□.

D. □□□□ □□□ □□□ □□ □□□ □□ □□□ □□□□□.

Answer: ([SHOW ANSWER](#)**)**

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

NEW QUESTION: 119

□□ □□ □□ □□□ □□ □□□ □□□□ □□ □□□ □□□□ □□ □□□ □□□□ □□ □□ □□□ □□□ □□□□□. □□ □□□ □□ □□ □□ □□□ □□ □□□□ □□□ □□ □□ □□ □□ □□ □□□ □□□□□□.

□□ □□ □□□ □□□□□□.

1. □□□ □□ □□□ □□ □□ □□□ □□□□□: □□ □□, □□, □□, □, □□ □, □□ □□ □.

2. □□□□ □□ □□ □□, □□ □□ □□□ □□ □□□□ □□□□□.

□□□ □□ □ □□ □□□□ □□:

□□□□

□□□□

□□□□□

□□□□□□

□□ □□ □ □□ □□ □□ □□ □ □□□□ □□ □□□□□?

A. □□ □□□□ □□□ □□□ □□□ □□□□ □□□ □□ □□ □□ □□ □□ □□□□ □□ □□□□□.

B. □□ □□□ □□□ □□□ □□ □□ □□□ □□□ □□□□□ □□, □□ □□ □□□ □□□ □ □□ □□ □ □□□ □□□□□.

C. □□ □□□□ □□ □□□ □□ □□ □ □□ □□ □□□□ □□□□ □□□ □□□□ □□□ □ □□□□□.

D. □□ □□□□ □□□□ □□□□ □□□ □□□ □□□ □□ □□□□ □□□□□.

Answer: (SHOW ANSWER)

In a planning poker session, if there is a significant difference in the estimations, it indicates that there may be misunderstandings or different perspectives on the complexity of the user story. According to the agile principles, the team should discuss these differences to reach a common understanding. The goal is to ensure that all team members have a shared understanding of the user story's scope and complexity before finalizing the estimate.

NEW QUESTION: 120

□□ □□□ '□□□□□ □□□ □□ □□' □□(IEEE 829)□ □□□□□.

□□ □□□ □□□□?

A. □ □□□□ □□□ □□□ □□□, □□□ □ □□ □□□□ □□□ □□□ □□□.

B. □□□□ □□□ □□□ □□□ □□ □□□□ □□□ □□ □□□ □□□□ □□ □□ □ □□ □ □□□□.

C. □□□ □□□ □□□ □□□ □□□ □ □□□ □□□ □□□□ □□□□.

D. □ □□□ □□ □□□ □□ □□□□□ □□□ □□□□. □□□ □□ □□□□□ □□ □□□ □□ □□ □□□ □□□□.

Answer: (SHOW ANSWER)

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: 121

□□□□ □□□□ □□□ □□□ □□□□ □□□□.

- A. □□□□ □□□□□ □□□ □□□□ □ □□□□□.
- B. □□□□□□ □□□ □□□ □□□ □□□□ □□□□□□□□.
- C. □□ □□□ □□□□□□ □□□ □□□□.
- D. □□□□□□ □□□□ □□□□□□□.

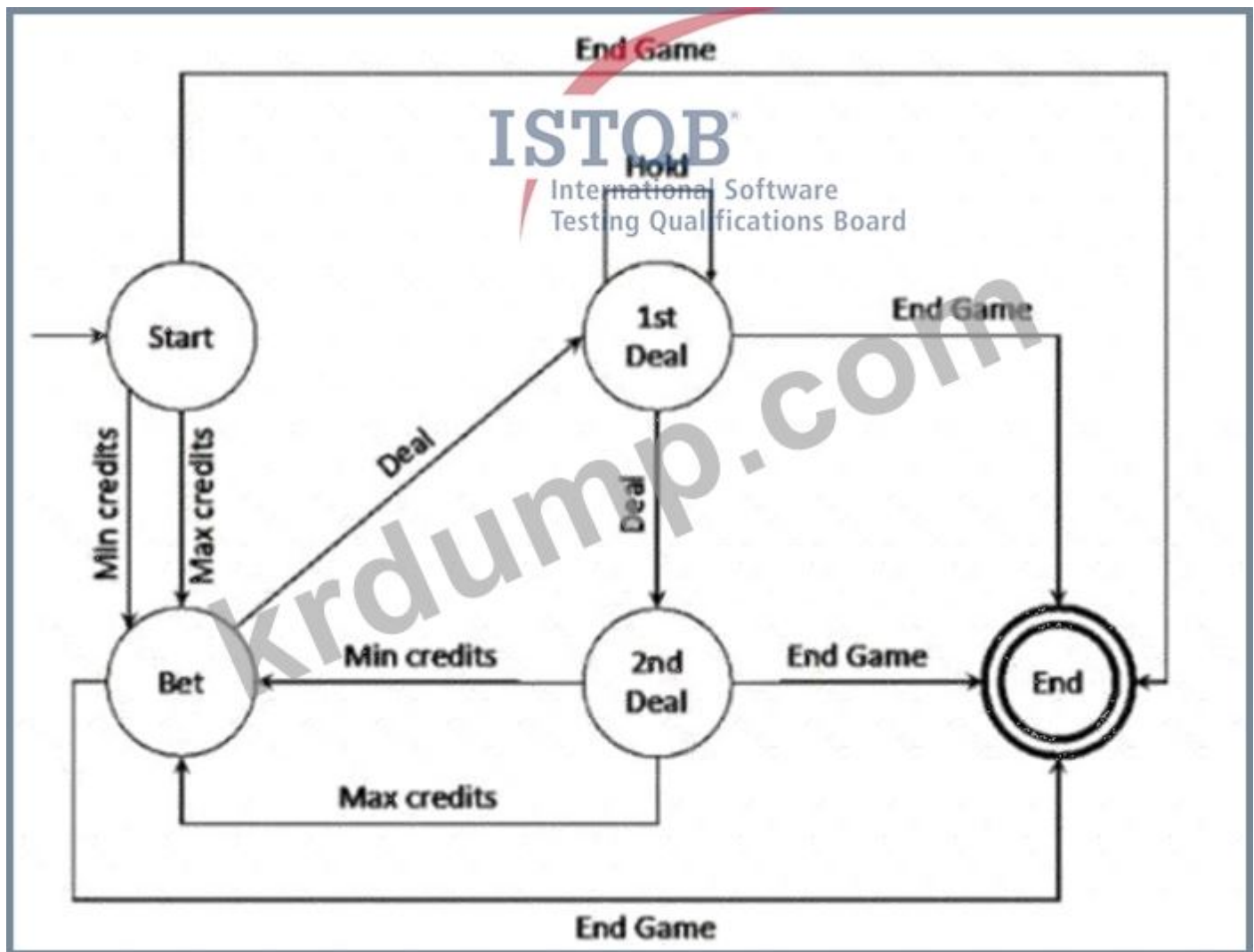
Answer: B ([LEAVE A REPLY](#))

This answer is correct because a typical objective of testing is to ensure that a software has been tested using a combination of test techniques, such as black-box, white-box, or experience-based techniques, that are appropriate for the test objectives, test levels, and test types. Testing using a combination of test techniques can increase the effectiveness and efficiency of testing, as different techniques can target different aspects of the software quality, such as functionality, usability, performance, security, reliability, etc. Testing using a combination of test techniques can also reduce the risk of missing defects that could be detected by one technique but not by another. References: ISTQB Foundation Level Syllabus v4.0, Section 2.3.1.1, Section 2.3.2

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □ □ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 122

□□□ □□ □□□ □□□ □□□□ □□ □□ □□□□□□ □□ □□□□ □□□ □□□□□.



"□□" □□□□ □□□□ "□" □□□□ □□□ □□ 3□ □□/2□ □□□ □□ □□□ □□□□
 □□□□ □ □□□ □□ □□□ □□□ □□ □□□□□?

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

Answer: D ([LEAVE A REPLY](#))

The minimum number of test cases needed to cover every unique sequence of up to 3 states/2 transitions starting in the "Start" state and ending in the "End" state is 4. This is because there are 4 unique sequences of up to 3 states/2 transitions starting in the "Start" state and ending in the "End" state:

- * Start -> Bet -> End
- * Start -> Deal -> End
- * Start -> 1st Deal -> End
- * Start -> 2nd Deal -> End

References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents.

NEW QUESTION: 123

□□ □ □□ □□□ □□□□□?

A. Agile □□□□□ □□□□ □□ □□ □□□ □□□ □□ □□ □□□ □□□ □□□ □□□□ □□ □□ □□□□ □□ □□□ □□□□.

B. □□□ □□ □□□ □□□□ □□□ □□□ □□□ □□□□ □□ □□ □□□ □□□ □□□ □□□□ □□□□.

C. Agile □□□□□ □□□□ □ □□ □□□ □□□ □□□□□ □□□□□. □□□□ □□ □□□ □□□ □□□ □□□□ □ □□□□ □□□□□.

D. Agile □□□□□ □□□ V □□□ □□ □□□ □□ □□□□ □□□ □□□ □□□□□ □□ □□ □ □□ □□□ □□ □□□ □□□ □□□ □□□□.

Answer: (SHOW ANSWER)

This answer is correct because in Agile software development, work product documentation, such as user stories, acceptance criteria, or test cases, tends to be lightweight and concise, as the focus is on working software and frequent communication rather than comprehensive documentation. Manual tests tend to be often unscripted, as they are often produced using experience-based test techniques, such as error guessing or exploratory testing, which rely on the tester's skills, knowledge, and creativity to find defects and provide feedback. References: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.2, Section 3.2.1.2

NEW QUESTION: 124

□□ □ □□□ □□□□ □□ □□ □□ □□□□□?

A. □□□ □□□□ □ □□□ □□□ □□□□ □□□

B. □□ □□□ □□ □□□ □□ □□ □□□ □□□□ □□□

C. □□□□ □□□ □□ □□□□ □□□ □□□

D. □□ □□ □□ □□□□ □□ □□□ □□□□ □□ □□□

Answer: A (LEAVE A REPLY)

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".

ISTQB-CTFL-KR □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ ISTQB-CTFL-KR □□! DumpTop □ □□ **ISTQB-CTFL-KR** □□ □□□ □□□□□□, DumpTop ISTQB-CTFL-KR □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □ □ DumpTop ISTQB-CTFL-KR □□□ □□□□□. <https://www.dumptop.com/ISTQB/ISTQB-CTFL-KR-dump.html> (409 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)