

ISTQB.ISTQB-CTFL.v2026-04-09.q129

Exam Code:	ISTQB-CTFL
Exam Name:	ISTQB Certified Tester Foundation Level (CTFL v4.0)
Certification Provider:	ISTQB
Free Question Number:	129
Version:	v2026-04-09
# of views:	124
# of Questions views:	1290
https://www.freeqas.com/qa/ISTQB/ISTQB-CTFL/ISTQB.ISTQB-CTFL.v2026-04-09.q129.html	

NEW QUESTION: 1

Which of the following issues cannot be identified by static analysis tools?

- A. Very low MTBF (Mean Time Between failure)
- B. Potentially endless loops
- C. Referencing a variable with an undefined value
- D. Security vulnerabilities

Answer: (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. Reference = 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: 2

Which of the following statements regarding the testing quadrants model is correct?

- A. The testing quadrants model assigns test types to the corresponding V-Model test levels.
- B. The testing quadrants model provides a way of differentiating between types of tests and assists in explaining them to stakeholders.
- C. The testing quadrants model assigns tests from one of the four quadrants to each iteration in the Agile development.
- D. With the testing quadrants model, tests can be either business-oriented or product-oriented.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 3

A software calculates the annual car tax using three inputs:

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

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

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

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

Answer: B ([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

Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 34.

NEW QUESTION: 4

You are working in an agile team on the creation of software for use in robotic assistance systems that support industrial manufacturing processes. You are currently in the effort estimation phase using planning poker. In the current sprint, three user stories are to be implemented:

- A . The assistance system is to have a customizable layout. For this purpose, the user should be able to choose between 3 combinations of font size and background color at startup.

B . In the initialization phase, the assistance system should validate the inputs of the 10 customer specific and process specific parameters against a list of 5 consistency rules and display the result to the user.

C . The assistance system currently supports the manufacturing process with 2 robot gripper arms, this shall be extended to 5 gripper arms. In addition, the system shall perform a pre-calculation of the movements of all gripper arms in real time and automatically stop the system in case of a calculated collision of the gripper arms.

You are now asked to estimate the test effort for each of these user stories by using planning poker. The following value rules apply for the test effort estimation:

3 - low (≤ 10 test cases)

8 - medium (> 10 and ≤ 20 test cases)

13 - high (> 20 and ≤ 50 test cases)

21 - very high (> 50 and ≤ 100 test cases)

34 - too complex (number of test cases cannot be estimated from information) For each of the three user stories, which planning poker value makes the most sense when estimating the test effort?

A. A:13; B:13; C:13;

B. A:3; B:13; C:34;

C. A:3; B:21; C:8;

D. A:21; B:13; C:13;

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

NEW QUESTION: 5

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

A. Planning the review

B. Determines if the review objectives have been met

C. Decide on the execution of reviews

D. Allocate time for review

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

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: 6

What is 'Component Testing'?

- A. Integration Testing
- B. Functional testing
- C. Experience-based testing
- D. A test level

Answer: (SHOW ANSWER)

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.

NEW QUESTION: 7

Which of the following statements about re-testing and regression testing are TRUE?

- I Re-testing should be performed after a defect is fixed.
- II Regression testing should always be performed after a defect is fixed.
- III Re-testing and regression testing may be performed at any test level.
- IV Regression testing may include functional, non-functional and structural testing.
- V Re-testing should be included in the debugging activity.

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

Answer: A (LEAVE A REPLY)

The following statements about re-testing and regression testing are true:

- I) Re-testing should be performed after a defect is fixed. Re-testing is a type of testing that verifies that a defect has been successfully resolved by executing a test case that previously failed due to that defect. Re-testing should be performed after a defect is fixed and delivered to ensure that it does not cause any new failures or side effects.

III) Re-testing and regression testing may be performed at any test level. Re-testing and regression testing are not limited to a specific test level, but can be applied at any level depending on the context and objectives. For example, re-testing and regression testing can be performed at unit level, integration level, system level or acceptance level.

IV) Regression testing may include functional, non-functional and structural testing. Regression testing is a type of testing that verifies that previously tested software still performs correctly after changes. Regression testing may include various types of testing depending on the scope and purpose of the changes. For example, regression testing may include functional testing to check if the software meets its requirements, non-functional testing to check if the software meets its quality attributes, or structural testing to check if the software meets its design or code standards. The following statement about re-testing and regression testing is false:

II) Regression testing should always be performed after a defect is fixed. Regression testing is not always necessary after a defect is fixed, as some defects may have a low impact or low likelihood of affecting other parts of the software. Regression testing should be performed after a defect is fixed only if there is a risk of introducing new defects or causing existing defects due to the changes made to fix the defect. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 2, page 19; Chapter 4, page 45.

NEW QUESTION: 8

A company runs a pilot project for evaluation of a test automation tool. Which of the following is NOT a valid object of this pilot project?

- A. Get familiar with the functionality and options of the tool
- B. Check how the tool fits to the existing test processes
- C. Train all testers on using the tool
- D. Decide upon standards for tool implementation

Answer: (SHOW ANSWER)

A pilot project is a small-scale experiment or trial that is conducted to evaluate the feasibility, effectiveness, and suitability of a test automation tool before implementing it on a larger scale¹.

The objectives of a pilot project may vary depending on the context and scope of the test automation initiative, but some common ones are²:

To get familiar with the functionality and options of the tool

To check how the tool fits to the existing test processes and environment To assess the benefits and challenges of using the tool To decide upon standards and guidelines for tool implementation and usage To estimate the costs and resources required for tool deployment and maintenance

Therefore, option C is not a valid objective of a pilot project, as it is not necessary to train all testers on using the tool at this stage. Training all testers on using the tool would be more appropriate after the tool has been selected and approved for full-scale implementation, and after the standards and guidelines have been established. Training all testers on using the tool during the pilot project would be inefficient, costly, and premature, as the tool may not be suitable or effective for the intended purpose, or may be replaced by another tool later.

Reference:

- 1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82
- 2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 83
- 3: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 84
- 4: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 85

NEW QUESTION: 9

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

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

Answer: C ([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: 10

The ISTQB glossary defines Quality Assurance as: "Plan or quality management focused on providing confidence that quality requirements will be fulfilled. Which of the following is not one of the Quality Assurance activities?"

- A.** Requirements elicitation
- B.** Defect analysis
- C.** Functional Testing
- D.** Performance Testing

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: 11

You are a test manager responsible for implementing risk-based testing in your organization which deals with software in the healthcare domain. You are writing a handbook of various product risk mitigation options. Which ONE of the following options correctly represent the correct mitigation options?

- A. Selecting a tester with required knowledge related to compliance and standards
- B. Using a third party testing company to transfer the risk to that company
- C. Choosing a tool for automated unit testing to reduce the risks
- D. Increasing the number of testers to be able to take care of all the risks

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 12

Which ONE of the following options BEST describes the third test quadrant (Q3)?

- A. This test quadrant contains smoke tests and non-functional tests (except usability tests). These tests are often automated.
- B. This test quadrant is business-facing and criticizes the system. It contains exploratory testing, usability testing, and user acceptance testing.
- C. This test quadrant contains functional tests, examples, user story tests, user experience prototypes, API testing, and simulations. These tests check the acceptance criteria and can be manual or automated.
- D. This test quadrant is technology-facing and supports the team. These tests should be automated and included in the continuous integration (CI) process.

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

Comprehensive and Detailed In-Depth Explanation:

The Agile Testing Quadrants framework categorizes tests based on their purpose and audience:

Quadrant 1 (Q1): Technology-facing tests (unit and component tests).

Quadrant 2 (Q2): Business-facing tests supporting development (e.g., BDD tests).

Quadrant 3 (Q3): Business-facing tests that critique the system (B).

Includes usability testing, exploratory testing, and UAT to ensure software meets user expectations.

Quadrant 4 (Q4): Technology-facing tests that critique the system (e.g., performance, security testing).

Option B correctly defines Q3 since it focuses on evaluating the user experience, exploring the system, and validating business expectations.

NEW QUESTION: 13

A test manager defined the following test levels in her test plan; Component, System and Acceptance.

Which Software Development lifecycle is the Test Manager most likely following?

- A. Waterfall
- B. Prototyping
- C. V-Model
- D. Agile

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

The test manager is most likely following the V-model for software development. The V-model is a software development model that defines four testing levels that correspond to four development phases: component testing corresponds to component design, integration testing corresponds to architectural design, system testing corresponds to system requirements specification, and 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: 14

Which of the following answers describes a reason for adopting experience-based testing techniques?

- A. Experience-based test techniques provide more systematic coverage criteria than black-box and white-box test techniques
- B. Experience-based test techniques completely rely on the tester's past experience for designing test cases.
- C. Experience-based test techniques allow designing test cases that are usually easier to reproduce than those designed with black-box and white-box test techniques.
- D. Experience-based test techniques tend to find defects that may be difficult to find with black-box and white-box test techniques and are often useful to complement these more systematic techniques.

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

Experience-based testing techniques leverage the tester's intuition and prior experience to identify defects that systematic techniques might miss. These techniques are valuable because they can uncover issues based on real-world usage and scenarios that aren't always covered by more formalized black-box and white-box methods. The ISTQB CTFL Syllabus v4.0 highlights the

complementary nature of experience-based techniques in providing a broader defect detection strategy.

NEW QUESTION: 15

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

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

Answer: C (LEAVE A REPLY)

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

* ISTQB Certified Tester Foundation Level Syllabus v4.0:

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

* ISTQB News Release on CTFL v4.0:

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

NEW QUESTION: 16

A number of characteristics are given for impact of SDLC on the testing effort.

- i. Finishing of requirements review leading to test analysis
- ii. Both - static and dynamic testing performed at unit testing level
- iii. Frequent regression testing may need to be performed
- iv Extensive product documentation
- v. More use of exploratory testing

Which of the following statements is MOST correct?

- A.** i and iv are characteristics of sequential models; ii. iii and v are characteristics of iterative and incremental models
- B.** iii and iv are characteristics of sequential models and i. ii and v are characteristics of iterative and incremental models
- C.** ii and iii are characteristics of sequential models; iv and v are characteristics of iterative and incremental models
- D.** ii and iv are characteristics of sequential models; i. iii and v are characteristics of iterative and incremental models

Answer: (SHOW ANSWER)

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 17

Which of the following is the BEST reason for selecting a particular type of software development lifecycle model?

- A. The project manager's preference
- B. Tester skill level with the software development lifecycle model
- C. The project team's overall familiarity with the model
- D. The type of product being developed

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: 18

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

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

Answer: B (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).

Reference:

Certified Tester Foundation Level v4.0

ISTQB Foundation Level Syllabus 4.0 (2023)

NEW QUESTION: 19

Consider the following code

```
int premium=2500;
if (age<30)
{
    premium = premium +1500;
}
```

Which options suits for a correct combination of Boundary value and expected result. Assume first number as boundary followed by expected result.

A. 29, 1500

30. 2500

B. 29. 4000

30. 2500

C. 29, 2500

30. 1500

D. 30, 1500

31,2500

Answer: ([SHOW ANSWER](#))

In the given code snippet, the premium is increased by 1500 if the age is less than 30. Therefore, at the boundary value of age 29, the premium should be $2500 + 1500 = 4000$, and at age 30, the premium should remain at its initial value of 2500, as the condition is no longer met. Option B correctly reflects this with 29, 4000 and 30, 2500 as the boundary value and the expected results, respectively.

NEW QUESTION: 20

You are a tester working on a critical project. Based on the risk analysis you execution? Priority 1 is the highest and Priority 3 is the lowest priority.

Test Case	Priority	Dependencies
Test A	3	-
Test B	1	Test D
Test C	2	Test A
Test D	3	-

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

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

NEW QUESTION: 21

Which of the following is not an example of a typical generic skill required for testing?

- A. Be able to apply test-driven development
- B. Be able to use test management tools and defect tracking tools
- C. Be able to communicate defects and failures to developers as objectively as possible
- D. Possess the necessary social skills that support effective teamwork

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

Test-driven development is not an example of a typical generic skill required for testing, but rather an example of a specific technical skill or a development practice that may or may not be relevant for testing, depending on the context and the objectives of the testing activities. Test-driven development is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design, and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a skill that is generally expected or needed for testers, especially for testers who are not involved in unit testing or who do not have access to the source code. The other options are examples of typical generic skills required for testing, which are skills that are applicable and beneficial for testing in any context or situation, regardless of the specific testing techniques, tools, or methods used. The typical generic skills required for testing include:

- * Be able to use test management tools and defect tracking tools: These are tools that help testers to plan, organize, monitor, and control the testing activities and resources, as well as to record, track, analyze,
- * and resolve the defects detected during testing. These tools can improve the efficiency, the effectiveness, and the communication of the testing process, as well as to provide traceability, metrics, and reports for the testing outcomes.
- * Be able to communicate defects and failures to developers as objectively as possible: This is a skill that involves the ability to report and describe the defects and failures found during testing in a clear, concise, accurate, and unbiased manner, using relevant information, evidence, and

terminology, without making assumptions, judgments, or accusations. This skill can facilitate the collaboration, the understanding, and the resolution of the defects and failures between the testers and the developers, as well as to prevent conflicts, misunderstandings, or blame games.

* Possess the necessary social skills that support effective teamwork: These are skills that involve the ability to interact, cooperate, and coordinate with other 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. These skills can include communication, negotiation, leadership, motivation, feedback, conflict resolution, etc. These skills can enhance the quality, the productivity, and the satisfaction of the testing process, as well as to foster a positive and constructive testing culture. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles

* ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches

* ISTQB Glossary of Testing Terms v4.0, Test-driven Development, Test Management Tool, Defect Tracking Tool, Defect Report, Failure, Social Skill2

NEW QUESTION: 22

In a two-hour uninterrupted test session, performed as part of an iteration on an Agile project, a heuristic checklist was used to help the tester focus on some specific usability issues of a web application.

The unscripted tests produced by the tester's experience during such session belong to which one of the following testing quadrants?

A. Q1

B. Q2

C. Q3

D. Q4

Answer: ([SHOW ANSWER](#))

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: 23

Test automation allows you to:

- A.** demonstrate the absence of defects
- B.** produce tests that are less subject to human errors
- C.** avoid performing exploratory testing
- D.** increase test process efficiency by facilitating management of defects

Answer: (SHOW ANSWER)

Test automation allows you to produce tests that are less subject to human errors, as they can execute predefined test scripts or test cases with consistent inputs, outputs, and expected results. Test automation can also reduce the manual effort and time required to execute repetitive or tedious tests, such as regression tests, performance tests, or data-driven tests. Test automation

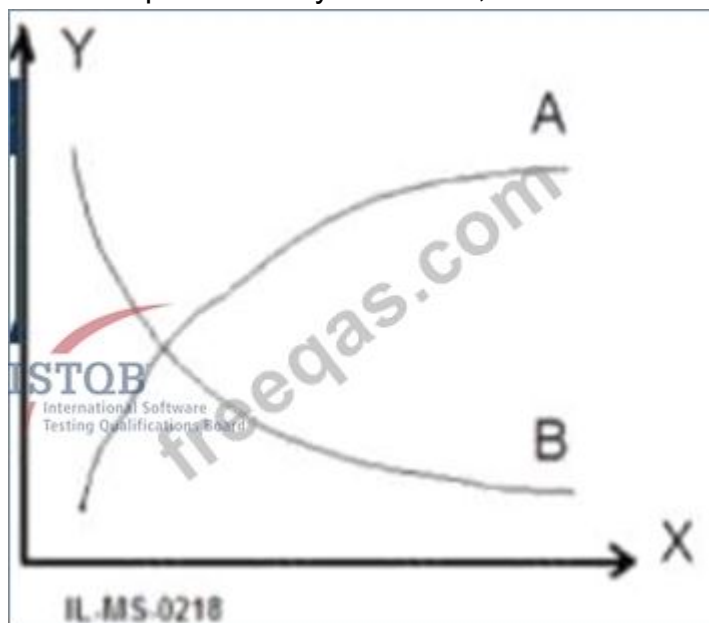
does not demonstrate the absence of defects, as it can only verify the expected behavior of the system under test, not the unexpected or unknown behavior. Test automation does not avoid performing exploratory testing, as exploratory testing is a valuable technique to discover new information, risks, or defects that are not covered by automated tests. Test automation does not increase test process efficiency by facilitating management of defects, as defect management is a separate activity that involves reporting, tracking, analyzing, and resolving defects, which may or may not be related to automated tests. References: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

* ISTQB® Certified Tester Foundation Level Syllabus v4.0, Chapter 3.3.1, Test Automation1

* ISTQB® Glossary of Testing Terms v4.0, Test Automation2

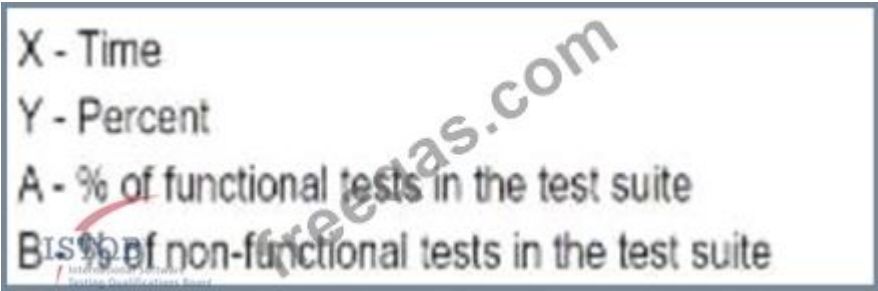
NEW QUESTION: 24

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



- A.
- X - Time
 - Y - Cost
 - A - Cost of test (per week)
 - B - Cost of finding a single bug (per week)

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

- C. 
- 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: 25

Which of the following is a valid collection of equivalence classes for the following problem: An integer field shall contain values from and including 1 to and including 15.

- A. Less than 0.1 through 14. 15 and more
- B. Less than 1.1 through 14. more than 15
- C. negative numbers. 1 through 15. above 15
- D. Less than 1.1 through 15. more than 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: 26

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



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

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

The figure depicts the life-cycle of a defect using state transition testing. State transition testing is a technique that models how a system transitions from one state to another depending on events or conditions. The figure shows six states (S0 to S5) and seven transitions (T0 to T6). The correct sequence of state transitions that follows the figure is S0->S1->S2->S3->S5->S1->S2->S3. This sequence represents the following scenario:

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

2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION: 27

A software calculates the annual car tax using three inputs:

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

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

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

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

Answer: B ([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: 28

Which of the following is the most correct statement about state testing techniques?

- A. Static techniques can be used before all code is ready for execution
- B. Static techniques find more defects than dynamic techniques.
- C. Static techniques can be used by inexperienced users.
- D. Static techniques are always cheaper than dynamic techniques.

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

State testing techniques are a type of dynamic testing techniques that are based on the behavior of the system under test for different input conditions and events. Dynamic testing techniques require the system to be executed with test cases, whereas static testing techniques do not. Static testing techniques can be applied before the code is ready for execution, such as reviews, inspections, walkthroughs, and static analysis. Static testing techniques can help find defects early in the development process, improve the quality of the code, and reduce the cost and effort of dynamic testing. References = ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Chapter 4, Section 4.2.1, Page 281; ISTQB Glossary of Testing Terms v4.0, Page 292

NEW QUESTION: 29

A Test Manager conducts risk assessment for a project. One of the identified risks is: "The sub-contractor may fail to meet his commitment". If this risk materializes, it will lead to delay in completion of testing required for the current cycle.

Which of the following sentences correctly describes the risk?

- A. It is a product risk since any risk associated with development timeline is a product risk.
- B. It is no longer a risk for the Test Manager since an independent party (the sub-contractor) is now managing it

C. It is a object risk since successful completion of the object depends on successful and timely completion of the tests

D. It is a product risk since default on part of the sub-contractor may lead to delay in release of the product

Answer: D (LEAVE A REPLY)

* A product risk is a risk that affects the quality or timeliness of the software product being developed or tested¹. Product risks are related to the requirements, design, implementation, verification, and maintenance of the software product².

* The risk of the sub-contractor failing to meet his commitment is a product risk, as it could cause a delay in the completion of the testing required for the current cycle, which in turn could affect the release date of the product. The release date is an important aspect of the product quality, as it reflects the customer satisfaction and the market competitiveness of the product³.

* The other options are not correct because:

* A. It is not true that any risk associated with development timeline is a product risk. Some risks could be project risks, which are risks that affect the management or control of the software project, such as budget, resources, schedule, or communication¹. For example, a risk of losing a key project stakeholder is a project risk, not a product risk.

* B. It is not true that the risk is no longer a risk for the Test Manager since an independent party is managing it. The Test Manager is still responsible for ensuring that the testing activities are completed according to the test plan and the quality objectives⁴. The Test Manager should monitor and control the sub-contractor's performance and communicate with him regularly to identify and mitigate any potential issues or deviations⁵.

* C. It is not clear what is meant by "object" in this option, but it could be interpreted as the software system under test or the test object⁶. In any case, the risk is not an object risk, as it does not affect the successful completion of the object, but rather the successful completion of the testing of the object. An object risk could be a risk that affects the functionality, reliability, usability, efficiency, maintainability, or portability of the software system under test². For example, a risk of the software system having a high complexity or a low testability is an object risk, not a product risk.

References =

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

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

* 3 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 99

* 4 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 100

* 5 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 101

* 6 ISTQB® Certified Tester Foundation Level Syllabus v4.0, 2023, p. 102

NEW QUESTION: 30

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



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

Answer: D (LEAVE A REPLY)

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

NEW QUESTION: 31

What is test oracle?

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

Answer: (SHOW ANSWER)

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

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam!

PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:

<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As

Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 32

Consider a given test plan which, among others, contains the following three sections: "Test Scope", "Testing Communication", and "Stakeholders". The features of the test object to be tested and those excluded from the testing represent information that is:

- A.** not usually included in a test plan, and therefore in the given test plan it should not be specified neither within the three sections mentioned, nor within the others
- B.** 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
- C.** usually included in a test plan and, in the given test plan, it is more likely to be specified within "Testing Communication" rather than in the other two sections mentioned
- D.** usually included in a test plan and, in the given test plan, it is more likely to be specified within "Stakeholders" rather than in the other two sections mentioned

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: 33

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

What are the advantages and the disadvantages of this change?

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

Disadvantage: pulled to support tasks and having less time for testing

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

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

NEW QUESTION: 34

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

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

Answer: ([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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 7.

NEW QUESTION: 35

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

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

Answer: ([SHOW ANSWER](#))

Debugging the software to find the reason for defects is not a common objective of testing, but rather a task of development or maintenance. Debugging is a process of locating and fixing errors in the software code, while testing is a process of finding and reporting defects in the software behavior or quality. Testing does not aim to fix defects, but rather to provide information on their existence and impact. The other options are common objectives of testing. Finding defects in the software is one of the main objectives of testing, as it helps to improve the quality and reliability of the software. Preventing defects is another objective of testing, as it helps to avoid rework and reduce costs and risks. Providing information on the status of the system is another objective of testing, as it helps to support decision making and risk management. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 3.

NEW QUESTION: 36

Why should you choose a test technique?

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

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

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

Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 31.

NEW QUESTION: 37

Which of the following statements about the shift-left approach is true?

- A.** Shift-left in testing can be implemented only in Agile/DevOps frameworks, as it relies completely on automated testing activities performed within a continuous integration process
- B.** Performance testing performed during component testing, is a form of shift-left in testing that avoids planning and executing costly end-to-end testing at the system test level in a production-like environment
- C.** Shift-left in testing can be implemented in several ways to find functional defects early in the lifecycle, but it cannot be relied upon to find defects associated with non-functional characteristics
- D.** Continuous integration supports shift-left in testing as it can reduce the time between the introduction of a defect and its detection, thereby reducing the cost to fix it

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: 38

Select which of the following statements describe the key principles of software testing?

- i. Testing shows the presence of defects, not their absence.
- ii. Testing everything is possible.
- iii. Early testing is more expensive and is a waste of time.
- iv. Defects cluster together.
- v. Testing is context dependent.
- vi. Beware of the pesticide paradox.
- vii. Absence of errors is a fallacy.

Select the correct answer:

- A.** i, iv, v, vi and vii
- B.** i, ii, v, vi and vii
- C.** iii, iv, v, vi and vii
- D.** ii, iii, iv, v and vi

Answer: A (LEAVE A REPLY)

The key principles of software testing include: i. Testing shows the presence of defects, not their absence. iv. Defects cluster together. v. Testing is context dependent. vi. Beware of the pesticide paradox. vii. Absence of errors is a fallacy. These principles highlight the importance of recognizing the limitations and context of testing, as well as the potential for repeated tests to become less effective.

NEW QUESTION: 39

You are responsible for applying the correct technique for a review of the requirements document for a project to develop a new software application. You identify the reviewers and the required roles, including the meeting leader, who is the requirements document author, and a separate role for a scribe. Additionally, you decide to take a relatively informal approach to the requirements review. The goal of the review is to find defects in the requirements document, such as omissions, Inconsistencies, and duplications. Another goal of the review is to improve the software application's usability and accessibility by considering the various stakeholders' viewpoints.

Which of the following statements BEST describes this scenario?

- A. This scenario is using a pair review type and a perspective-based review technique
- B. This scenario is using a walkthrough review type and a checklist-based review technique
- C. This scenario is using a walkthrough review type and a perspective-based review technique
- D. This scenario is using a pair review type and a checklist-based review technique

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

This scenario is using a walkthrough review type and a perspective-based review technique. In a walkthrough, the author of the document leads the meeting and it typically includes a meeting leader and a scribe, as described. This type of review is informal, focuses on discussion, and often involves scenario-based reading of the document to understand different user perspectives(ISTQB Main Web).References:

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

NEW QUESTION: 40

A Static analysis tool analyzes a given program's CONTROL FLOW among other things. Which of the following options represents the most likely outcome of the control flow analysis:

- A. Identification of unreachable code
- B. Report on adherence to the coding standards
- C. Number of comment lines
- D. 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: 41

Which of the following types of tools is BEST suited for determining source code compliance with the guidelines provided by a coding standard?

- A. Containerisation tool
- B. Fault seeding tool.
- C. Static analysis tool.
- D. Test data preparation tool

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: 42

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

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

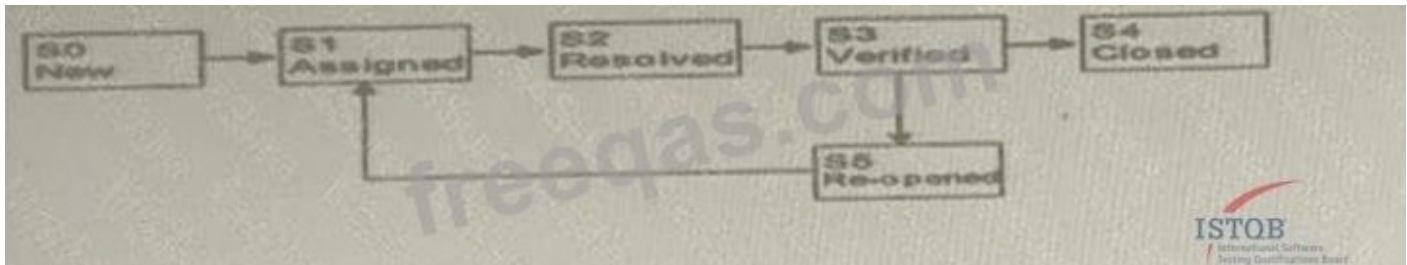
Answer: D ([LEAVE A REPLY](#))

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: 43

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



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

Answer: ([SHOW ANSWER](#))

The figure depicts the life-cycle of a defect using state transition testing. State transition testing is a technique that models how a system transitions from one state to another depending on events or conditions. The figure shows six states (S0 to S5) and seven transitions (T0 to T6). The correct sequence of state transitions that follows the figure is S0->S1->S2->S3->S5->S1->S2->S3. This sequence represents the following scenario:

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

2018 Syllabus - Springer], Chapter 4, page 40-41.

NEW QUESTION: 44

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

What are the advantages and the disadvantages of this change?

A. Advantage: More independence in deciding what and how to test,

Disadvantage: Isolation from the development team knowledge

B. Advantage: being closer to customer perspective,

Disadvantage less independence in perspectives

C. Advantage: pulled to support tasks and having less time for testing, Disadvantage less chances to move a tester to development

D. Advantage: increased chances to move a tester to development;

Disadvantage: pulled to support tasks and having less time for testing

Answer: B (LEAVE A REPLY)

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

NEW QUESTION: 45

Consider the following iteration planning tasks where a tester can provide value:

* Break down user stories into tasks (particularly testing tasks)

* Estimate test effort for all testing tasks

* Identify and refine functional and non-functional aspects of the test object Which ONE of the following tasks should be ADDED to the above list?

A. Determining the test strategy

B. Participating in the detailed risk analysis of user stories and determining their testability

C. Planning the testing for the release

D. Writing testable user stories and acceptance criteria

Answer: B (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation: During iteration planning, testers provide input on risk analysis, testability, and the feasibility of test automation. Participating in risk analysis (B) helps identify areas needing more extensive testing.

* (A) is incorrect because the test strategy is usually set at a higher level (project or release level).

* (C) is incorrect as release planning is broader than iteration planning.

* (D) is incorrect since writing user stories is mainly the responsibility of the product owner, though testers contribute to defining acceptance criteria.

Risk-based test prioritization ensures testing efforts are aligned with business needs and software risks.

NEW QUESTION: 46

When should component integration tests be carried out?

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

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

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

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 47

You are a test manager responsible for implementing risk-based testing in your organization which deals with software in the healthcare domain. You are writing a handbook of various product risk mitigation options.

Which ONE of the following options correctly represent the correct mitigation options?

- A. Selecting a tester with required knowledge related to compliance and standards
- B. Using a third party testing company to transfer the risk to that company
- C. Increasing the number of testers to be able to take care of all the risks
- D. Choosing a tool for automated unit testing to reduce the risks

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

NEW QUESTION: 48

Given the following review process main activities and specific review activities:

- a.Planning
 - b.Initiate review
 - c.Issue communication and analysis
 - d.Fixing and reporting
- 1.Creating defect reports
 - 2.Estimating effort and timeframe
 - 3.Recording updated status of defects
 - 4.Selecting the people to participate

5.Distributing the work product and other material

6.Evaluating the review findings

Which of the following BEST matches the review process main activities with the appropriate specific review activities?

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-a, 4-b, 5-b, 6-c, 2-d, 3-d

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

Answer: (SHOW ANSWER)

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: 49

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

A. Most defects are found in a small subset of a system's modules.

B. If new defects are to be found we should run the same test set more often.

C. Testing is better if it starts at the beginning of a project.

D. How testing is done, is based on the situation in a particular project.

Answer: (SHOW ANSWER)

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: 50

You work as a test manager for a supplier of PC games which can be purchased and downloaded via an online store. A tester from your team recorded the following failure caused by a defect:

Issue ID: DEF00223167

Title: Saving after shopping in online shop fails

Date: 2023-04-18

Tester: Sabine Meier

Description: "Saving the game immediately after purchasing it in the online store fails. As a result, the purchase is paid for, but the goods are not delivered." Attached data: "log.txt", "screen1.bmp", "screen2.bmp" During the review of the defect report, you notice that at least one important piece of information is missing for defining how to deal with the defect. Which one?

- A. Tracking the quality of the work product
- B. Name of the developer who most likely caused the defect
- C. Severity of the defect
- D. Ideas for test process improvement

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

NEW QUESTION: 51

Out of the following. what is not needed to specify in defect report?

- A. Test environment details
- B. How to reproduce the defect
- C. How to fix the defect
- D. Severity and priority

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

A defect report is a document that records the details of a defect found during testing. A defect report typically contains the following items:

Identifier: A unique identifier for the defect report

Summary: A concise summary of the defect

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

Severity: The degree of impact that the defect has on the system
Priority: The level of urgency for resolving the defect
Status: The current state of the defect, such as new, open, resolved, closed, etc.

Resolution: The action taken to resolve the defect, such as fix, workaround, reject, etc. Out of these items, the one that is not needed to specify in a defect report is how to fix the defect. How to fix the defect is a technical solution that is usually determined by the developer who is assigned to resolve the defect. How to fix the defect is not part of the defect report, but rather part of the code change or patch that is delivered to fix the defect. The other items are needed to specify in a defect report, as they provide essential information for identifying, tracking and resolving defects.

Verified Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 3, page 32-33.

NEW QUESTION: 52

Which review type, also known as a "buddy check", is commonly used in Agile development?

- A. Inspection.
- B. Walkthrough.
- C. Technical review.

D. Informal review.

Answer: D (LEAVE A REPLY)

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: 53

Which of the following statements about how different types of test tools support testers is true?

- A.** The support offered by a test data preparation tool is often leveraged by testers to run automated regression test suites
- B.** The support offered by a performance testing tool is often leveraged by testers to run load tests
- C.** The support offered by a bug prediction tool is often used by testers to track the bugs they found
- D.** The support offered by a continuous integration tool is often leveraged by testers to automatically generate test cases from a model

Answer: B (LEAVE A REPLY)

The support offered by a performance testing tool is often leveraged by testers to run load tests, which are tests that simulate a large number of concurrent users or transactions on the system under test, in order to measure its performance, reliability, and scalability. Performance testing tools can help testers to generate realistic workloads, monitor system behavior, collect and analyze performance metrics, and identify performance bottlenecks. The other statements are false, because:

A test data preparation tool is a tool that helps testers to create, manage, and manipulate test data, which are the inputs and outputs of test cases. Test data preparation tools are not directly related to running automated regression test suites, which are test suites that verify that the system still works as expected after changes or modifications. Regression test suites are usually executed by test execution tools, which are tools that can automatically run test cases and compare actual results with expected results.

A bug prediction tool is a tool that uses machine learning or statistical techniques to predict the likelihood of defects in a software system, based on various factors such as code complexity, code churn, code coverage, code smells, etc. Bug prediction tools are not used by testers to track the bugs they found, which are the actual defects that have been detected and reported during testing. Bugs are usually tracked by defect management tools, which are tools that help testers to record, monitor, analyze, and resolve defects.

A continuous integration tool is a tool that enables the integration of code changes from multiple developers into a shared repository, and the execution of automated builds and tests, in order to ensure the quality and consistency of the software system. Continuous integration tools are not used by testers to automatically generate test cases from a model, which are test cases that are derived from a representation of the system under test, such as a state diagram, a decision table, a use case, etc. Test cases can be automatically generated by test design tools, which are tools that support the implementation and maintenance of test cases, based on test design specifications or test models. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 3.4.1, Types of Test Tools
ISTQBGlossary of Testing Terms v4.0, Performance Testing Tool, Test Data Preparation Tool, Bug Prediction Tool, Continuous Integration Tool, Test Execution Tool, Defect Management Tool, Test Design Tool

NEW QUESTION: 54

Which of the following characterizations applies to a test tool used for the analysis of a developer's code prior to its execution?

- A.** Tool support for test design and implementation.
- B.** Tool support for static testing.
- C.** Tool support for test execution and logging.
- D.** Tool support for performance measurement and dynamic analysis.

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

A test tool used for the analysis of a developer's code prior to its execution falls under the category of static testing tools. Static testing involves examining the code and documentation without executing the code. These tools are used to perform static analysis, which helps in identifying potential defects and code quality issues early in the development process. The ISTQB CTFL syllabus specifies that static analysis tools are essential for finding defects that do not manifest themselves during the execution of the program.

References: ISTQB CTFL Syllabus, Section 3.1, "Static Testing."

NEW QUESTION: 55

Which of the following statements best describes how configuration management supports testing?

- A.** Configuration management helps reduce testing effort by identifying a manageable number of test environment configurations in which to test the software, out of all possible configurations of the environment in which the software will be released
- B.** Configuration management is an administrative discipline that includes change control, which is the process of controlling the changes to identified items referred to as Configuration Items'
- C.** Configuration management is an approach to interoperability testing where tests are executed in the cloud, as the cloud can provide cost-effective access to multiple configurations of the test environments

D. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation

Answer: ([SHOW ANSWER](#))

This answer is correct because configuration management is a process of establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. Configuration management helps ensure that all relevant project documentation and software items are uniquely identified in all their versions and therefore can be unambiguously referenced in test documentation. This supports testing by providing traceability, consistency, and control over the test artifacts and the software under test. References: : ISTQB Glossary of Testing Terms v4.0, : ISTQB Foundation Level Syllabus v4.0, Section 2.2.2.2

NEW QUESTION: 56

Match each objective to the correct test level

Objective:

- A) Verifying whether the functional and non-functional behaviors of the system are as designed and specified.
- B) Verifying whether the functional and non-functional behaviors of the interfaces are as designed.
- C) Verifying whether the functional and non-functional behaviors of the components are as designed and specified.
- D) Establishing confidence in the quality of the system as a whole.

Test Level:

- 1. Component testing.
- 2. Integration testing.
- 3. System testing.
- 4. Acceptance testing.

A. A3, B2, C4, D1

B. A2, B3, C1, D4

C. A3, B2, C1, D4

Answer: ([SHOW ANSWER](#))

The test levels and their objectives can be matched as follows:

- * Verifying whether the functional and non-functional behaviors of the system are as designed and specified (A3: System testing).
- * Verifying whether the functional and non-functional behaviors of the interfaces are as designed (B2: Integration testing).
- * Verifying whether the functional and non-functional behaviors of the components are as designed and specified (C1: Component testing).
- * Establishing confidence in the quality of the system as a whole (D4: Acceptance testing).

NEW QUESTION: 57

Which of the following statements is true?

- A. A defect does not always produce a failure, while a bug always produces a failure
- B. A defect may cause a failure which, when occurring, always causes an error
- C. Failures can be caused by defects, but also by environmental conditions
- D. Bugs are defects found during component testing, while failures are defects found at higher test levels

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

Failures can be caused by defects, but also by environmental conditions. A failure is an event in which the software system does not perform a required function or performs a function incorrectly, according to the expected behavior. A defect is a flaw in the software system or a deviation from the requirements or the specifications, that may cause a failure. However, not all failures are caused by defects, as some failures may be caused by environmental conditions, such as hardware malfunctions, network interruptions, power outages, incompatible configurations, etc. Environmental conditions are factors that affect the operation of the software system, but are not part of the software system itself. The other statements are false, because:

A defect does not always produce a failure, while a bug always produces a failure. This statement is false, because a defect may or may not produce a failure, depending on the inputs, the outputs, the states, or the scenarios of the software system, and a bug is just another term for a defect, so it has the same possibility of producing a failure as a defect. For example, a defect in a rarely used feature or a hidden branch of the code may never produce a failure, while a defect in a frequently used feature or a critical path of the code may produce a failure often. A bug is not a different concept from a defect, but rather a synonym or a colloquial term for a defect, so it has the same definition and implications as a defect.

A defect may cause a failure which, when occurring, always causes an error. This statement is false, because an error is not a consequence of a failure, but rather a cause of a defect. An error is a human action or a mistake that produces a defect in the software system, such as a typo, a logic flaw, a requirement misunderstanding, etc. An error is not observable in the software system, but rather in the human mind or the human work products, such as the code, the design, the documentation, etc. A failure is not a cause of an error, but rather a result of a defect, which is a result of an error. For example, an error in the code may cause a defect in the software system, which may cause a failure in the software behavior.

Bugs are defects found during component testing, while failures are defects found at higher test levels. This statement is false, because bugs and failures are not different types of defects, but rather different terms for defects and their manifestations. As mentioned before, bugs are just another word for defects, and failures are the events in which the software system does not perform as expected due to defects. Bugs and failures can be found at any test level, not only at component testing or higher test levels. Test levels are the stages of testing that correspond to the levels of integration of the software system, such as component testing, integration testing, system testing, and acceptance testing. Defects and failures can occur and be detected at any test level, depending on the test objectives, the test basis, the test techniques, and the test

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

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality¹

ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles¹

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

Development Lifecycles¹ ISTQB Glossary of Testing Terms v4.0, Failure, Defect, Bug,

Environmental Condition, Error, Test Level²

NEW QUESTION: 58

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

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

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

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

A. Use case based testing

B. Decision table

C. Boundary value analysis

D. Equivalence class partitioning

Answer: B (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.

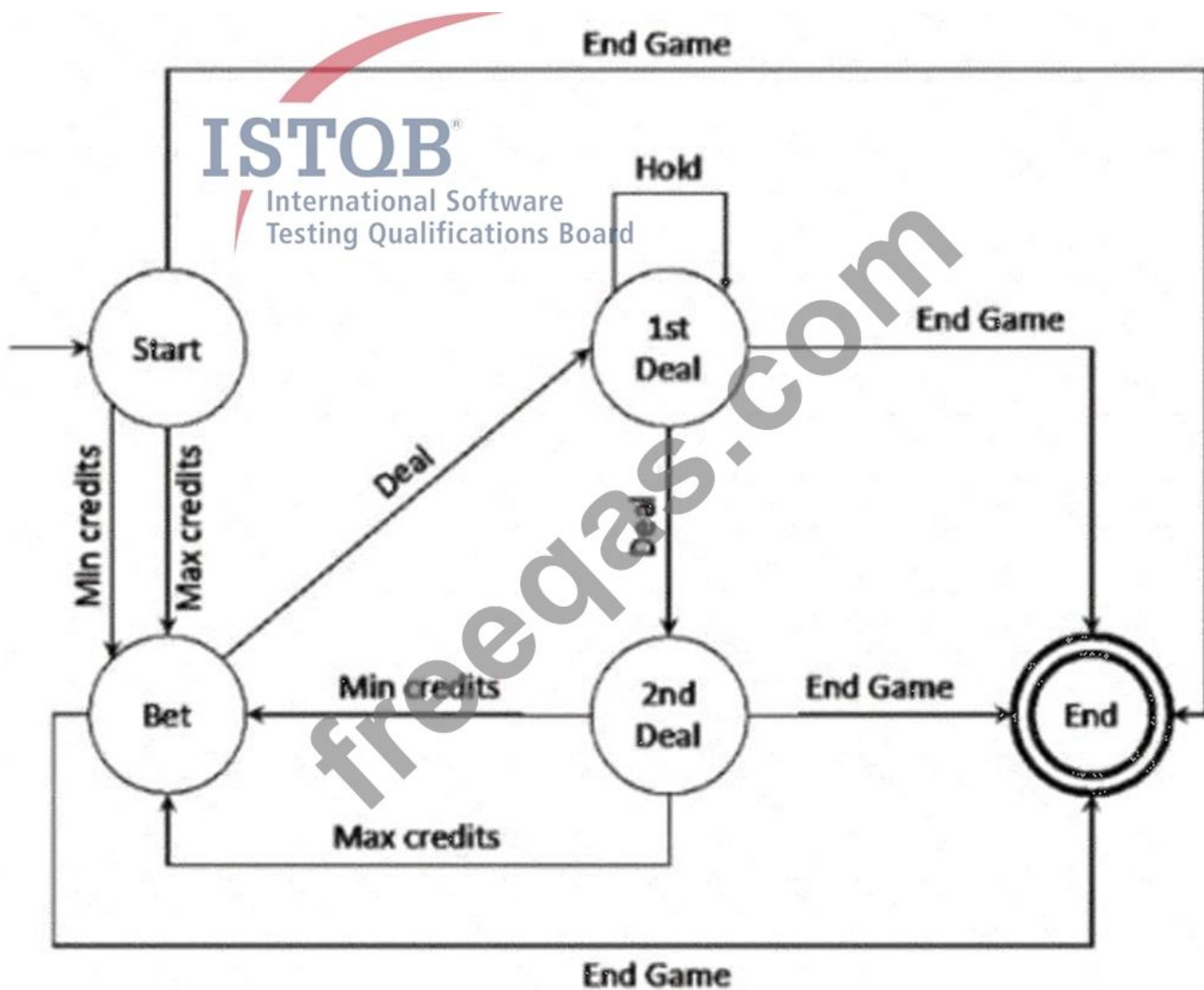
Boundary value analysis can be used to test components that have input values that can be divided into partitions of equivalent data. For example, for testing the ATM, we can identify boundary values for the input amount, such as the minimum and maximum amount allowed by the system or the user's account. 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: 59

Consider the following simplified version of a state transition diagram that specifies the behavior of a video poker game:



What Is 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?

- 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: 60

Which of the following best describes the relationship between a test progress report and a test summary report?

- A. The test report prepared during a test activity may be referred to as a test summary report, while a test report prepared at the end of a test activity may be referred to as a test progress report.
- B. There is no difference between a test progress report and a test summary report.
- C. Both the test progress report and the test summary report should always be generated via an automated tool.
- D. The test report prepared during a test activity may be referred to as a test progress report, while a test report prepared at the end of a test activity may be referred to as a test summary report.

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

NEW QUESTION: 61

Which of the following options is correct with respect to the differentiation of black-box techniques, white-box techniques, and experience-based testing?

- A. Black-box test techniques are based on requirements, white-box tests on source code. Therefore, black-box test techniques are always performed at higher test levels (e.g. system and acceptance level) and white-box test techniques at lower test levels.
- B. Experience-based testing can complement white-box and black-box test techniques well or can be used if the basic principles for these test techniques are lacking.
- C. Experience-based testing is an extension of black-box test techniques, since experience-based test techniques also do not require knowledge about the source code.
- D. The coverage items can be specified in advance, e.g., coverage of statements, states or checklists, for all three of the categories mentioned above.

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

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 62

The following 4 equivalence classes are given:

$x \leq -100$
 $-100 < x < 100$
 $100 \leq x < 1000$
 $x \geq 1000$

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

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

Answer: (SHOW ANSWER)

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.

Reference:

- 1: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 38
- 2: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 39
- 3: ISTQB Foundation Level Syllabus 2018, Version 4.0, p. 40

NEW QUESTION: 63

Which of the following is not an example of a typical content of a test completion report for a test project?

- A. The additional effort spent on test execution compared to what was planned
- B. The unexpected test environment downtime that resulted in slower test execution
- C. The residual risk level if a risk-based test approach was adopted
- D. The test procedures of all test cases that have been executed

Answer: D (LEAVE A REPLY)

This answer is correct because the test procedures of all test cases that have been executed are not a typical content of a test completion report for a test project. A test completion report is a

document that summarizes the test activities and results at the end of a test project. It usually includes information such as the test objectives, scope, approach, resources, schedule, results, deviations, issues, risks, lessons learned, and recommendations for improvement. The test procedures of all test cases that have been executed are part of the test documentation, but they are not relevant for the test completion report, as they do not provide a high-level overview of the test project outcomes and performance. Reference: ISTQB Foundation Level Syllabus v4.0, Section 2.5.3.2

NEW QUESTION: 64

You are testing the latest version of an air-traffic control system prior to production deployment using exploratory testing. After following an unusual sequence of input steps, the system crashes. After the crash, you document a defect report with the following information:

*Title: System crashes unexpectedly during input.

*Brief summary: System crashes when an unusual sequence of inputs is used.

*Version: V1.001

*Test: Exploratory testing prior to production deployment

*Priority: Urgent

*Risk: High

*References: Screenshot of crashed application

What critical information is missing from this report?

- A. Conclusions, recommendations, and approvals.
- B. Change history.
- C. Description of the defect to enable reproduction.
- D. Status of defect

Answer: ([SHOW ANSWER](#))

The critical information missing from the defect report is a detailed description of the defect to enable reproduction. A clear and concise description of the steps taken to reproduce the defect is essential for developers to understand the context and to be able to replicate the issue in their environment. Without this information, it can be challenging to diagnose and fix the defect. The ISTQB CTFL syllabus emphasizes the importance of providing all necessary details in a defect report to facilitate effective communication and resolution.

References: ISTQB CTFL Syllabus, Section 5.5, "Defect Management."

NEW QUESTION: 65

Which of the following issues cannot be identified by static analysis tools?

- A. Very low MTBF (Mean Time Between failure)
- B. Potentially endless loops
- C. Referencing a variable with an undefined value
- D. Security vulnerabilities

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

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: 66

Which of the following statements is true?

- A.** Experience-based test techniques rely on the experience of testers to identify the root causes of defects found by black-box test techniques
- B.** Some of the most common test basis used by white-box test techniques include user stories, use cases and business processes
- C.** Experience-based test techniques are often useful to detect hidden defects that have not been targeted by black-box test techniques
- D.** The primary goal of experience-based test techniques is to design test cases that can be easily automated using a GUI-based test automation tool

Answer: ([SHOW ANSWER](#))

Experience-based test techniques are test design techniques that rely on the experience, knowledge, intuition, and creativity of the testers to identify and execute test cases that are likely to find defects in the software system. Experience-based test techniques are often useful to detect hidden defects that have not been targeted by black-box test techniques, which are test design techniques that use the external behavior and specifications of the software system as the

test basis, without considering its internal structure or implementation. Experience-based test techniques can complement black-box test techniques by covering aspects that are not explicitly specified, such as usability, security, reliability, performance, etc. The other statements are false, because:

- * Experience-based test techniques do not rely on the experience of testers to identify the root causes of defects found by black-box test techniques, but rather to identify the potential sources of defects based on their own insights, heuristics, or exploratory testing. The root causes of defects are usually identified by debugging or root cause analysis, which are activities that involve examining the code or the development process to find and fix the errors that led to the defects.
 - * Some of the most common test basis used by white-box test techniques include the source code, the design documents, the architecture diagrams, and the control flow graphs of the software system. White-box test techniques are test design techniques that use the internal structure and implementation of the software system as the test basis, and aim to achieve a certain level of test coverage based on the code elements, such as statements, branches, paths, etc. User stories, use cases, and business processes are examples of test basis used by black-box test techniques, as they describe the functional and non-functional requirements of the software system from the perspective of the users or the stakeholders.
 - * The primary goal of experience-based test techniques is not to design test cases that can be easily automated using a GUI-based test automation tool, but rather to design test cases that can reveal defects that are not easily detected by other test techniques, such as boundary value analysis, equivalence partitioning, state transition testing, etc. Test automation is the use of software tools to execute test cases and compare actual results with expected results, without human intervention. Test automation can be applied to different types of test techniques, depending on the test objectives, the test levels, the test tools, and the test resources. However, test automation is not always feasible or beneficial, especially for test cases that require human judgment, creativity, or exploration, such as those designed by experience-based test techniques.
- 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 Techniques
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.2, White-box Test Design Techniques
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.2.3, Experience-based Test Design Techniques
- * ISTQB Glossary of Testing Terms v4.0, Experience-based Test Technique, Black-box Test Technique, White-box Test Technique, Test Basis, Test Coverage, Test Automation

NEW QUESTION: 67

A software development company invests heavily in tools to support the entire software development lifecycle, including testing. They have a tool that allows automated creation and

installation of builds and subsequent execution of various types of automated tests. Which ONE of the following is a CORRECT statement about this type of test tool?

- A. It is an example of automated unit testing tool
- B. It is an example of collaboration tool
- C. It is an example of test implementation tool
- D. It is an example of DevOps related tools

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

NEW QUESTION: 68

For each test case to be executed, the following table specifies its dependencies and the required configuration of the test environment for running such test case:

Test Case	Dependencies	Configuration
TC1		CONF2
TC2	TC4	CONF2
TC3	TC4	CONF1
TC4		CONF1
TC5	TC1	CONF2

Assume that CONF1 is the initial configuration of the test environment. Based on this assumption, which of the following is a test execution schedule that is compatible with the specified dependencies and allows minimising the number of switches between the different configurations of the test environment?

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

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

To determine the optimal test execution schedule that minimizes the number of configuration switches and respects the dependencies, we start with the initial configuration, CONF1.

* TC4: It has no dependencies and runs on CONF1 (initial configuration).

* TC3: Depends on TC4 and runs on CONF1. Since TC4 is already executed, we can proceed with TC3.

* TC2: Depends on TC4 and runs on CONF2. We switch to CONF2 after TC3.

* TC1: No dependencies and runs on CONF2. Since we are already in CONF2, we can execute TC1 next.

* TC5: Depends on TC1 and runs on CONF2. Since TC1 is already executed, we can proceed with TC5 without additional configuration switches.

By following this sequence (TC4, TC3, TC2, TC1, TC5), we respect the dependencies and minimize the number of configuration switches.

NEW QUESTION: 69

The four test levels used in ISTQB syllabus are:

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

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

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

Answer: D (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: 70

Which of the following would be the LEAST likely to be used as the basis for a test exit criteria?

- A.** Test schedules
- B.** Cost of testing performed so far
- C.** Confidence of testers in tested code

D. Number of unfixed defects

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

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: 71

Which of the following is correct with regards to debugging?

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

Answer: A ([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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 1, page 6.

NEW QUESTION: 72

Which of the following is a key characteristic of informal reviews?

- A.** Kick-off meeting
- B.** Low cost
- C.** Individual preparation
- D.** Metrics analysis

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

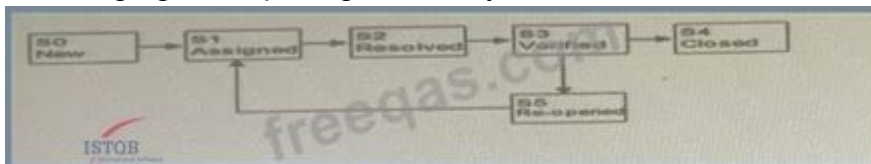
A key characteristic of informal reviews is low cost. Informal reviews are a type of review that does not follow a formal process or have any formal documentation. Informal reviews are usually

performed by individuals or small groups of peers or colleagues who have some knowledge or interest in the product under review.

Informal reviews can be done at any time and for any purpose, such as checking for errors, clarifying doubts, sharing ideas, etc. Informal reviews have low cost, as they do not require much time, effort, or resources to conduct. The other options are not key characteristics of informal reviews. Kick-off meeting is a characteristic of formal reviews, such as inspections or walkthroughs. Kick-off meeting is a meeting that is held before the review process starts, where the roles and responsibilities of the participants are defined, the objectives and scope of the review are agreed, and the logistics and schedule of the review are planned. Individual preparation is a characteristic of formal reviews, such as inspections or walkthroughs. Individual preparation is an activity that is performed by the reviewers before the review meeting, where they examine the product under review and identify any issues or questions that need to be discussed or resolved during the review meeting. Metrics analysis is a characteristic of formal reviews, such as inspections or walkthroughs. Metrics analysis is an activity that is performed after the review process is completed, where the data and results of the review are collected and analyzed to measure the effectiveness and efficiency of the review, as well as to identify any improvement actions or lessons learned for future reviews. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 9.

NEW QUESTION: 73

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



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

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

The figure depicts the life-cycle of a defect using state transition testing. State transition testing is a technique that models how a system transitions from one state to another depending on events or conditions. The figure shows six states (S0 to S5) and seven transitions (T0 to T6). The correct sequence of state transitions that follows the figure is S0->S1->S2->S3->S5->S1->S2->S3. This sequence represents the following scenario:

- * S0: The defect is not yet detected (initial state).
- * T0: The defect is detected by testing (event).
- * S1: The defect is reported and registered (state).
- * T1: The defect is assigned to a developer for fixing (event).
- * S2: The defect is being fixed by the developer (state).

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

NEW QUESTION: 74

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

Which coverage metric will provide the needed information for this analysis?

- A. Code coverage
- B. Data flow coverage
- C. Statement coverage
- D. Branch Coverage

Answer: ([SHOW ANSWER](#))

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

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

The other options are not correct because they are not suitable for testing the cases where a specific action is not allowed. Code coverage is a general term that encompasses various types

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

NEW QUESTION: 75

Which of the following is a task the Author is responsible for, as part of a typical formal review?

- A. Determining the people who will be involved in the review
- B. Recording the anomalies found during the review meeting
- C. Identifying potential anomalies in the work product under review
- D. Fixing the anomalies found in the work product under review

Answer: C (LEAVE A REPLY)

This answer is correct because identifying potential anomalies in the work product under review is one of the tasks the Author is responsible for, as part of a typical formal review. The Author is the person who creates the work product to be reviewed, such as a requirement specification, a design document, or a test case. The Author's tasks include preparing the work product for the review, identifying potential anomalies in the work product, and fixing the anomalies found in the work product after the review. References: ISTQB Glossary of Testing Terms v4.0, ISTQB Foundation Level Syllabus v4.0, Section 2.4.2.1

NEW QUESTION: 76

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

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

Answer: A (LEAVE A REPLY)

An incident is any event that occurs during testing that requires investigation. An incident report is a document that records the details of an incident. The next step the tester should take prior to recording the deviation is to try to recreate the incident before reporting. This can help confirm

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

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 77

A company runs a pilot project for evaluation of a test automation tool. Which of the following is NOT a valid object of this pilot project?

- A. Get familiar with the functionality and options of the tool
- B. Check how the tool fits to the existing test processes
- C. Train all testers on using the tool
- D. Decide upon standards for tool implementation

Answer: C (LEAVE A REPLY)

- * A pilot project is a small-scale experiment or trial that is conducted to evaluate the feasibility, effectiveness, and suitability of a test automation tool before implementing it on a larger scale¹.
- * The objectives of a pilot project may vary depending on the context and scope of the test automation initiative, but some common ones are²:
 - * To get familiar with the functionality and options of the tool
 - * To check how the tool fits to the existing test processes and environment
 - * To assess the benefits and challenges of using the tool
 - * To decide upon standards and guidelines for tool implementation and usage
 - * To estimate the costs and resources required for tool deployment and maintenance

* Therefore, option C is not a valid objective of a pilot project, as it is not necessary to train all testers on using the tool at this stage. Training all testers on using the tool would be more appropriate after the tool has been selected and approved for full-scale implementation, and after the standards and guidelines have been established. Training all testers on using the tool during the pilot project would be inefficient, costly, and premature, as the tool may not be suitable or effective for the intended purpose, or may be replaced by another tool later.

1: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 82

2: ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 83

3 ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 84

4 ISTQB Certified Tester Foundation Level Syllabus 2018, Version 4.0, p. 85

NEW QUESTION: 78

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:

A. is a false myth

B. is expressed by the testing principle referred to as 'Tests wear out'

C. is expressed by the testing principle referred to as 'Defects cluster together'

D. is expressed by the testing principle referred to as 'Bug prediction'

Answer: C (LEAVE A REPLY)

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.

NEW QUESTION: 79

Which of the following statements about statement coverage is TRUE?

A. Achieving 90% statement coverage ensures that 90% branch coverage is achieved.

B. Achieving 100% statement coverage ensures that no variable within the code has been used without being initialised.

C. Achieving 100% statement coverage ensures that 100% branch coverage is achieved

D. Achieving 80% statement coverage ensures that 80% of all executable statements within the code have been exercised.

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: 80

Consider the following statements about risk-based testing.

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

- A. I III IV and V are true. II is false.
- B. II, III IV and V are correct. I is false.
- C. C. I, II and IV are true. III and V are false.
- D. II, III and V are true. 1 ants Iv are false.

Answer: ([SHOW ANSWER](#))

The following statements about risk-based testing are correct:

* II) Tests should be prioritized to find tie critical detects as early as possible. Risk-based testing involves prioritizing tests based on risk level, which reflects both the likelihood and impact of defects or failures.

Tests with higher risk level should be executed earlier than tests with lower risk level, in order to find and fix critical defects as soon as possible.

* III) Non-testing activities may also help to reduce risk. Risk-based testing does not only involve testing activities, but also other activities that can help mitigate risks, such as reviews, inspections, audits, simulations or prototyping.

NEW QUESTION: 81

Which ONE of the following statements does NOT describe how testing contributes to higher quality?

- A. Properly designed tests that pass reduce the level of risk in a system.
- B. The testing of software demonstrates the absence of defects.
- C. Software testing identifies defects, which can be used to improve development activities.
- D. Performing a review of the requirement specifications before implementing the system can enhance quality.

Answer: B ([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: 82

Scenario 2 "Big Drop":

A company "The Big Drop" provides bulk discounts and frequent customer discounts as follows:

Bulk Discounts:

Purchase Volume	Discount (%)
1 - 100 units	0%
101 - 1000 units	5%
>1000 units	10%

Frequent Customer Discounts:

Frequent Customer?	Discount (%)
YES	5%
NO	0%

How many possible decision rules can be extracted from this table?

- A. 5 decision rules
- B. 6 decision rules
- C. 32 decision rules
- D. 8 decision rules

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

Comprehensive and Detailed In-Depth Explanation: Decision rules define possible combinations of conditions and outcomes.

For this scenario:

* Bulk discount has 3 categories (0%, 5%, 10%)

* Frequent customer discount has 2 categories (Yes = +5%, No = 0%)

The total number of decision rules is: 3 bulk discount options × 2 frequent customer options = 6 rules.

Thus, the correct answer is 6 decision rules (B).

NEW QUESTION: 83

Which of the following statements is true?

- A. In Agile software development, work product documentation tends to be lightweight and manual tests tend to be often unscripted as they are often produced using experience-based test techniques
- B. Sequential development models impose the use of systematic test techniques and do not allow the use of experience-based test techniques
- C. In Agile software development, the first iterations are exclusively dedicated to testing activities, as testing will be used to drive development, which will be performed in the subsequent iterations
- D. Both in Agile software development and in sequential development models, such as the V-model, test levels tend to overlap since they do not usually have defined entry and exit criteria

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

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. Reference: ISTQB Foundation Level Syllabus v4.0, Section 3.1.1.2, Section 3.2.1.2

NEW QUESTION: 84

Which of the following statements refers to good testing practice to be applied regardless of the chosen software development model?

- A.** Tests should be written in executable format before the code is written and should act as executable specifications that drive coding
- B.** Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level
- C.** Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly
- D.** Involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle

Answer: D (LEAVE A REPLY)

The statement that refers to good testing practice to be applied regardless of the chosen software development model is option D, which says that involvement of testers in work product reviews should occur as early as possible to take advantage of the early testing principle. Work product reviews are static testing techniques, in which the work products of the software development process, such as the requirements, the design, the code, the test cases, etc., are examined by one or more reviewers, with or without the author, to identify defects, violations, or improvements. Involvement of testers in work product reviews can provide various benefits for the testing process, such as improving the test quality, the test efficiency, and the test communication. The early testing principle states that testing activities should start as early as possible in the software development lifecycle, and should be performed iteratively and continuously throughout the lifecycle. Applying the early testing principle can help to prevent, detect, and remove defects at an early stage, when they are easier, cheaper, and faster to fix, as well as to reduce the risk, the cost, and the time of the testing process. The other options are not good testing practices to be applied regardless of the chosen software development model, but rather specific testing practices that may or may not be applicable or beneficial for testing, depending on the context and the objectives of the testing activities, such as:

* Tests should be written in executable format before the code is written and should act as executable specifications that drive coding: This is a specific testing practice that is associated with test-driven development, which is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design,

and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a good testing practice to be applied regardless of the chosen software development model, as it may not be feasible, suitable, or effective for testing in some contexts or situations, such as when the requirements are unclear, unstable, or complex, when the test automation tools or skills are not available or adequate, when the testing objectives or levels are not aligned with the unit testing, etc.

* Test levels should be defined such that the exit criteria of one level are part of the entry criteria for the next level: This is a specific testing practice that is associated with sequential software development models, such as the waterfall model, the V-model, or the W-model, in which the software development and testing activities are performed in a linear and sequential order, with well-defined phases, deliverables, and dependencies. Test levels are the stages of testing that correspond to the levels of integration of the software system, such as component testing, integration testing, system testing, and acceptance testing. Test levels should have clear and measurable entry criteria and exit criteria, which are the conditions that must be met before starting or finishing a test level. In sequential software development models, the exit criteria of one test level are usually part of the entry criteria for the next test level, to ensure that the software system is ready and stable for the next level of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be relevant, flexible, or efficient for testing in some contexts or situations, such as when the software development and testing activities are performed in an iterative and incremental order, with frequent changes, feedback, and adaptations, as in agile software development models, such as Scrum, Kanban, or XP, when the test levels are not clearly defined or distinguished, or when the test levels are performed in parallel or concurrently, etc.

* Test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly: This is a specific testing practice that is associated with uniform software development models, such as the spiral model, the incremental model, or the prototyping model, in which the software development and testing activities are performed in a cyclical and repetitive manner, with similar phases, deliverables, and processes. Test objectives are the goals or the purposes of testing, which can vary depending on the test level, the test type, the test technique, the test environment, the test stakeholder, etc. Test objectives can be defined in terms of the test basis, the test coverage, the test quality, the test risk, the test cost, the test time, etc. Test objectives should be specific, measurable, achievable, relevant, and time-bound, and they should be aligned with the project objectives and the quality characteristics. In uniform software development models, the test objectives may be the same for all test levels, as the testing process is repeated for each cycle or iteration, with similar focus, scope, and perspective of testing. However, this is not a good testing practice to be applied regardless of the chosen software development model, as it may not be appropriate, realistic, or effective for testing in some contexts or situations, such as when the software development and testing activities are performed in a hierarchical and modular manner, with different phases, deliverables, and dependencies, as in sequential software development models, such as the waterfall model, the V-model, or the W-model, when the test objectives vary according to the test levels, such as

component testing, integration testing, system testing, and acceptance testing, or when the test objectives change according to the feedback, the learning, or the adaptation of the testing process, as in agile software development models, such as Scrum, Kanban, or XP, etc.

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

- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 1.3.1, Testing in Software Development Lifecycles¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.2, Test Monitoring and Control¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.4, Test Implementation¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.5, Test Execution¹
- * ISTQB Certified Tester Foundation Level Syllabus v4.0, Chapter 2.1.6, Test Closure¹
- * ISTQB Glossary of Testing Terms v4.0, Work Product Review, Static Testing, Early Testing, Test-driven Development, Test Level, Entry Criterion, Exit Criterion, Test Objective, Test Basis, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time²

NEW QUESTION: 85

In which of the following cases you would NOT execute maintenance testing?

- A. Retirement of the software or system
- B. Modifications to a released software or system
- C. Migration of the system data to a replacement system
- D. Update to the Maintainability requirements during the development phase

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

Maintenance testing is testing performed on a software product after delivery to correct defects or improve performance or other attributes. Maintenance testing can be triggered by various situations, such as modifications to a released software or system, migration of the system data to a replacement system, or retirement of the software or system. Maintenance testing is not executed when there is an update to the maintainability requirements during the development phase, as this is not a maintenance situation but rather a change request that should be handled by the development process. Verified References: [A Study Guide to the ISTQB® Foundation Level 2018 Syllabus - Springer], Chapter 2, page 18-19.

NEW QUESTION: 86

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

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

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

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: 87

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

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

Answer: A ([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: 88

Which of the following work products cannot be examined by static analysis?

- A. Test plans
- B. Source code
- C. Compiled code
- D. Formal models

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

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: 89

What role in a formal or technical review should mediate between different opinions to ensure an effective review?

- A. Moderator (or facilitator)
- B. Manager
- C. Scribe (or recorder)
- D. Reviewer

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

NEW QUESTION: 90

You are testing a system that is used in motor vehicles to warn the driver of an obstacle when reversing.

Output is provided by a series of LED lights (green, yellow, and red), each illuminated based on clearly defined conditions.

The following summary describes the functionality:

*Object within 10 metres, green LED lit.

*Object within 5 metres, yellow LED lit.

*Object within 1 metre, red LED lit.

*Setting sensitivity mode to "ON" will result in only the red LED being lit when the object is within 1 metre.

The following decision table describes the rules associated with the functioning of this proximity warning system:

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6
Distance < 10 m	Y	N	N	Y	N	N
Distance < 5 m	N	N	N	N	Y	N
Distance < 1 m	N	N	Y	N	N	Y
Sensitivity ON	N	N	N	Y	Y	Y
Actions						
Green LED	Y	N	N	N	N	N
Yellow LED	N	Y	N	N	N	N
Red LED	N	N	Y	N	N	Y

Which intended functionality is tested by Rule 5 in the decision table?

A. Object is within 5 metres of the vehicle and the sensitivity mode is switched "off", resulting in the yellow LED being lit.

B. Object is within 5 metres of the vehicle and the sensitivity mode is switched "on", resulting in the yellow LED being lit.

C. Object is within 5 metres of the vehicle and the sensitivity mode is switched "off", resulting in no LED being lit.

D. Object is within 5 metres of the vehicle and the sensitivity mode is switched "on", resulting in no LED being lit.

Answer: D (LEAVE A REPLY)

Rule 5 in the decision table indicates that when the object is within 5 metres of the vehicle and the sensitivity mode is switched "on", no LED is lit. This matches the conditions and actions described in the decision table provided, ensuring that only the red LED is lit when the sensitivity mode is on and the object is within 1 metre, otherwise no LED is lit .

NEW QUESTION: 91

Which of the following statements is correct?

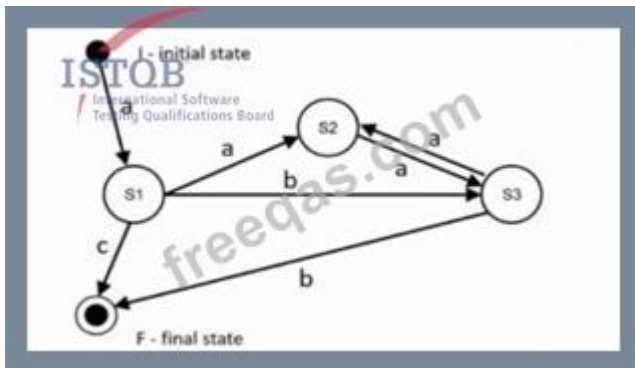
- A.** Testing based on the structure of the software under test (SUT) is white-box testing whereas testing the input-output behaviour without prewritten test cases is experience based testing.
- B.** Testing the software under test (SUT) using knowledge of the tester is experience based testing whereas without pre-written test cases is using black-box testing techniques.
- C.** Testing the software under test (SUT) is possible only after implementation in white-box testing whereas testing the software even before implementation is possible in exploratory testing.
- D.** Testing the software under test (SUT) based on design is black-box testing whereas testing the input- output behaviour is experience based testing.

Answer: ([SHOW ANSWER](#))

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 92

The following state diagram is given as basis for state transition testing and contains only valid transitions:



Explanation of the state diagram: States are depicted as nodes. The initial state is I, the final state is F. A state transition is depicted as a directed arrow with the initiating event as a label, e. g. from I to S1 with event a, and is also written as a triple (I,a,S1). Note: A test case is a sequence of events that initiates the corresponding sequence of state transitions.

The state diagram contains the following 7 state transitions:

- (I, a, S1)
- (S1, a, S2), (S1, b, S3), (S1, c, F)
- (S2, a, S3)
- (S3, a, S2), (S3, b, F)

What is the minimum number of test cases to achieve 100% coverage of all valid state transitions in the diagram?

- A. 2
- B. 4
- C. 3
- D. 8

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

NEW QUESTION: 93

A typical objective of testing is to ensure that:

- A. testing is used to drive the development of a software
- B. a software has been tested using a combination of test techniques
- C. there are no defects in a software that is about to be released
- D. a software has been properly covered

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

NEW QUESTION: 94

Which statement about use case testing is true?

- A. The test cases are designed to find defects in the data flow.
- B. The test cases are designed to be used by real users, not by professional testers
- C. The test cases are always designed by customers or end users.
- D. The test cases are designed to find defects in the process flow.

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

Use case testing is a technique that helps identify test cases that exercise the whole system on a transaction by transaction basis from start to finish. Use cases are descriptions of how users interact with the system to achieve a specific goal. Use case testing is not focused on data flow, but rather on process flow. Use case testing can be performed by professional testers, customers or end users, depending on the context. Use case testing does not require the test cases to be designed by customers or end users, but rather by anyone who has access to the use case specifications. Verified References: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 4, page 36.

NEW QUESTION: 95

Which of the following statements about exploratory testing is true?

- A. Exploratory testing is an experience-based test technique in which testers explore the requirements specification to detect non testable requirements
- B. When exploratory testing is conducted following a session-based approach, the issues detected by the testers can be documented in session sheets
- C. Exploratory testing is an experience-based test technique used by testers during informal code reviews to find defects by exploring the source code
- D. In exploratory testing, testers usually produce scripted tests and establish bidirectional traceability between these tests and the items of the test basis

Answer: B (LEAVE A REPLY)

Exploratory testing is an experience-based test technique in which testers dynamically design and execute tests based on their knowledge, intuition, and learning of the software system, without following predefined test scripts or test cases. Exploratory testing can be conducted following a session-based approach, which is a structured way of managing and measuring exploratory testing. In a session-based approach, the testers perform uninterrupted test sessions, usually lasting between 60 and 120 minutes, with a specific charter or goal, and document the issues detected, the test coverage achieved, and the time spent in session sheets. Session sheets are records of the test activities, results, and observations during a test session, which can be used for reporting, debriefing, and learning purposes. The other statements are false, because:

Exploratory testing is not a test technique in which testers explore the requirements specification to detect non testable requirements, but rather a test technique in which testers explore the software system to detect functional and non-functional defects, as well as to learn new information, risks, or opportunities. Non testable requirements are requirements that are ambiguous, incomplete, inconsistent, or not verifiable, which can affect the quality and effectiveness of the testing process. Non testable requirements can be detected by applying static testing techniques, such as reviews or inspections, to the requirements specification, before the software system is developed or tested.

Exploratory testing is not a test technique used by testers during informal code reviews to find defects by exploring the source code, but rather a test technique used by testers during dynamic testing to find defects by exploring the behavior and performance of the software system, without examining the source code. Informal code reviews are static testing techniques, in which the source code is analyzed by one or more reviewers, without following a formal process or using a checklist, to identify defects, violations, or improvements. Informal code reviews are usually performed by developers or peers, not by testers.

In exploratory testing, testers usually do not produce scripted tests and establish bidirectional traceability between these tests and the items of the test basis, but rather produce unscripted tests and adapt them based on the feedback and the findings of the testing process. Scripted tests are tests that are designed and documented in advance, with predefined inputs, outputs, and expected results, and are executed according to a test plan or a test procedure. Bidirectional traceability is the ability to trace both forward and backward the relationships between the items

of the test basis, such as the requirements, the design, the risks, etc., and the test artifacts, such as the test cases, the test results, the defects, etc. Scripted tests and bidirectional traceability are usually associated with more formal and structured testing approaches, such as specification-based or structure-based test techniques, not with exploratory testing. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents: ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 2.2.3, Experience-based Test Design Techniques¹ ISTQBGlossary of Testing Terms v4.0, Exploratory Testing, Session-based Testing, Session Sheet, Non Testable Requirement, Static Testing, Informal Review, Dynamic Testing, Scripted Testing, Bidirectional Traceability²

NEW QUESTION: 96

Which of the following statements about statement coverage is TRUE?

- A. Achieving 90% statement coverage ensures that 90% branch coverage is achieved.
- B. Achieving 100% statement coverage ensures that no variable within the code has been used without being initialised.
- C. Achieving 100% statement coverage ensures that 100% branch coverage is achieved
- D. Achieving 80% statement coverage ensures that 80% of all executable statements within the code have been exercised.

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: 97

"Statement Testing" is part of;

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

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

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

NEW QUESTION: 98

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

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

Answer: ([SHOW ANSWER](#))

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

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

NEW QUESTION: 99

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



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

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

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

NEW QUESTION: 100

Which of the following statements best describe Behavior-Driven Development (BDD)?

- A. A collaborative approach that allows every stakeholder to contribute to how the software component must behave.
- B. Expresses the behavior of an application with test cases written in Given When Then format.
- C. Is used to develop code guided by automated test cases.

D. A psychological technique in which the team's behavior in agile teams is evaluated.

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

Behavior-Driven Development (BDD) is a collaborative approach that enhances communication among project stakeholders, including developers, testers, and business analysts. It involves defining how software should behave through examples written in a common language understandable by all stakeholders, often using the Given-When-Then format.

NEW QUESTION: 101

Confirmation testing is performed after:

- A.** a defect is fixed and after other tests do not find any side-effect introduced in the software as a result of such fix
- B.** a failed test, and aims to run that test again to confirm that the same behavior still occurs and thus appears to be reproducible
- C.** the execution of an automated regression test suite to confirm the absence of false positives in the test results
- D.** a defect is fixed, and if such testing is successful then the regression tests that are relevant for such fix can be executed

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

Confirmation testing is performed after a defect is fixed, and if such testing is successful then the regression tests that are relevant for such fix can be executed. Confirmation testing, also known as re-testing, is the process of verifying that a defect has been resolved by running the test case that originally detected the defect.

Confirmation testing is usually done before regression testing, which is the process of verifying that no new defects have been introduced in the software as a result of changes or fixes.

Therefore, option D is the correct answer.

ISTQB Certified Tester Foundation Level Syllabus v4.01, Section 2.4.1, page 28; ISTQB Glossary v4.

02, page 15.

NEW QUESTION: 102

Use Scenario 1 "Happy Tomatoes" (from the previous question).

When running test case TC_59, the actual result for $t = 35$ degrees Celsius is $OUTPUT = X$ instead of the expected output.

Which information should NOT be included in the defect report?

- A.** Identification of the test object and test environment
- B.** A concise title and a short summary of the defect being reported
- C.** Description of the structure of the test team
- D.** Expected results and actual results

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

Comprehensive and Detailed In-Depth Explanation: A defect report should contain relevant details to help developers reproduce and fix the defect efficiently. The essential elements include:

- * Test object & environment (A)- To ensure reproducibility.
 - * Title & summary (B)- For quick identification.
 - * Expected vs. actual results (D)- To describe the discrepancy.
- The structure of the test team (C) is irrelevant for defect tracking and resolution.

NEW QUESTION: 103

Which ONE of the following elements is TYPICALLY NOT part of a test progress report?

- A. Obstacles and their workarounds
- B. A detailed assessment of product quality
- C. Test metrics to show the current status of the test process
- D. New or changed risks

Answer: (SHOW ANSWER)

Comprehensive and Detailed In-Depth Explanation: A test progress report provides an overview of testing activities, metrics, and identified risks. It focuses on tracking testing progress rather than evaluating overall product quality (B), which is typically included in a test summary report after testing is completed.

- * (A) is correct because obstacles (challenges) are reported to ensure test execution stays on track.
 - * (C) is correct as test metrics help stakeholders track execution progress.
 - * (D) is correct because new or changed risks impact test focus and priorities.
- A test progress report tracks execution and informs stakeholders about ongoing testing activities.

NEW QUESTION: 104

Which of the following statements is assigned correctly as a project or product risk?

- A. User experience feedback does not meet product expectations: this is a project risk.
- B. The system architecture does not adequately support some of the non-functional requirements: this is a product risk.
- C. The gradual expansion of the project scope results in significant rework: this is a product risk.
- D. The software does not perform its intended functions according to the specification: this is a project risk.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 105

Consider the following statements about risk-based testing.

- I) Risk-based testing has the objective to reduce the level of project risks.
 - II) Tests should be prioritized to find the critical defects as early as possible.
 - III) Non-testing activities may also help to reduce risk.
 - IV) Risks have to be reassessed on a regular basis.
 - V) The project stakeholders can give useful input to determine the risks.
- A. I, III, IV and V are true. II is false.
 - B. II, III, IV and V are correct. I is false.

C. C. I, II and IV are true.III and V are false.

D. II, III and V are true.I and IV are false.

Answer: (SHOW ANSWER)

The following statements about risk-based testing are correct:

* II) Tests should be prioritized to find the critical defects as early as possible. Risk-based testing involves prioritizing tests based on risk level, which reflects both the likelihood and impact of defects or failures.

Tests with higher risk level should be executed earlier than tests with lower risk level, in order to find and fix critical defects as soon as possible.

* III) Non-testing activities may also help to reduce risk. Risk-based testing does not only involve testing activities, but also other activities that can help mitigate risks, such as reviews, inspections, audits,

* simulations or prototyping.

NEW QUESTION: 106

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

A. The report is detailed and includes a status summary of defects by priority or budget

B. The report is detailed and includes specific information on defects and trends

C. The report is high-level and includes a status summary of defects by priority or budget

D. The report is high-level and includes specific information on defects and trends

Answer: C (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.

Reference:

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

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam!

PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:

<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As

Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 107

Which of the following statements is true?

- A.** Functional testing focuses on what the system should do while non-functional testing on the internal structure of the system
- B.** Non-functional testing includes testing of both technical and non-technical quality characteristics
- C.** Testers who perform functional tests are generally expected to have more technical skills than testers who perform non-functional tests
- D.** The test techniques that can be used to design white-box tests are described in the ISO/IEC 25010 standard

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

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.

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

02, page 40.

NEW QUESTION: 108

Which of the following is not an example of a typical generic skill required for testing?

- A.** Be able to apply test-driven development
- B.** Be able to use test management tools and defect tracking tools
- C.** Be able to communicate defects and failures to developers as objectively as possible
- D.** Possess the necessary social skills that support effective teamwork

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

Test-driven development is not an example of a typical generic skill required for testing, but rather an example of a specific technical skill or a development practice that may or may not be relevant for testing, depending on the context and the objectives of the testing activities. Test-driven development is an approach to software development and testing, in which the developers write automated unit tests before writing the source code, and then refactor the code until the tests pass. Test-driven development can help to improve the quality, the design, and the maintainability of the code, as well as to provide fast feedback and guidance for the developers. However, test-driven development is not a skill that is generally expected or needed for testers, especially for testers who are not involved in unit testing or who do not have access to the source code. The other options are examples of typical generic skills required for testing, which are skills that are applicable and beneficial for testing in any context or situation, regardless of the specific testing techniques, tools, or methods used. The typical generic skills required for testing include:

Be able to use test management tools and defect tracking tools: These are tools that help testers to plan, organize, monitor, and control the testing activities and resources, as well as to record,

track, analyze, and resolve the defects detected during testing. These tools can improve the efficiency, the effectiveness, and the communication of the testing process, as well as to provide traceability, metrics, and reports for the testing outcomes.

Be able to communicate defects and failures to developers as objectively as possible: This is a skill that involves the ability to report and describe the defects and failures found during testing in a clear, concise, accurate, and unbiased manner, using relevant information, evidence, and terminology, without making assumptions, judgments, or accusations. This skill can facilitate the collaboration, the understanding, and the resolution of the defects and failures between the testers and the developers, as well as to prevent conflicts, misunderstandings, or blame games. Possess the necessary social skills that support effective teamwork: These are skills that involve the ability to interact, cooperate, and coordinate with other 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. These skills can include communication, negotiation, leadership, motivation, feedback, conflict resolution, etc. These skills can enhance the quality, the productivity, and the satisfaction of the testing process, as well as to foster a positive and constructive testing culture. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 1.1.1, Testing and the Software Development Lifecycle ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 1.1.2, Testing and Quality ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 1.2.1, Testing Principles ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches ISTQBGlossary of Testing Terms v4.0, Test-driven Development, Test Management Tool, Defect Tracking Tool, Defect Report, Failure, Social Skill2

NEW QUESTION: 109

For each test case to be executed, the following table specifies its dependencies and the required configuration of the test environment for running such test case:

Test Case	Dependencies	Configuration
TC1		CONF2
TC2	TC4	CONF2
TC3	TC4	CONF1
TC4		CONF1
TC5	TC1	CONF2

Assume that CONF1 is the initial configuration of the test environment Based on this assumption, which of the following is a test execution schedule that is compatible with the specified dependencies and allows minimising the number of switches between the different configurations of the test environment^

A. TC4, TC3, TC2, TC1, TC5.

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

Answer: A (LEAVE A REPLY)

To determine the optimal test execution schedule that minimizes the number of configuration switches and respects the dependencies, we start with the initial configuration, CONF1.

- * TC4: It has no dependencies and runs on CONF1 (initial configuration).
- * TC3: Depends on TC4 and runs on CONF1. Since TC4 is already executed, we can proceed with TC3.
- * TC2: Depends on TC4 and runs on CONF2. We switch to CONF2 after TC3.
- * TC1: No dependencies and runs on CONF2. Since we are already in CONF2, we can execute TC1 next.
- * TC5: Depends on TC1 and runs on CONF2. Since TC1 is already executed, we can proceed with TC5 without additional configuration switches.

By following this sequence (TC4, TC3, TC2, TC1, TC5), we respect the dependencies and minimize the number of configuration switches

NEW QUESTION: 110

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

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

Answer: A (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: 111

Calculate the measurement error SD for the following estimates done using three point estimation technique- Most optimistic effort (a) -120 person days Most likely effort (m) -180 person days Most pessimistic effort (b)

- 240 person days

A. 197

B. 120

C. 20

D. 180

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

NEW QUESTION: 112

A software module to be used in a mission critical application incorporates an algorithm for secure transmission of data.

Which review type is most appropriate to ensure high quality and technical correctness of the algorithm?

A. Walkthrough

B. Technical Review

C. Management Review

D. Informal Review

Answer: B ([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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, Chapter 3, page 29-30.

NEW QUESTION: 113

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

- 1 - The valid ZIP code format is 5 numeric digits.
- 2 - The code has to exist in the post office's official ZIP code list

Using equivalence classes partitioning, how many test cases are required to test this field?

- A. 8**
- B. 3**
- C. 6**
- D. 4**

Answer: ([**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: 114

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

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

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

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

References:

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

NEW QUESTION: 115

In a review, which of the following is the responsibility of the manager?

- A. Measuring the performance of each reviewer
- B. Organizing the logistics of the review meeting
- C. Ensuring that sufficient time is available for review
- D. Performing detailed review based on past experience

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

NEW QUESTION: 116

Which of the following statements about the value of maintaining traceability between the test basis and test work products is not true?

- A. Traceability can be useful for assessing the impact of a change to a test basis item on the corresponding tests
- B. Traceability can be useful for determining how many test basis items are covered by the corresponding tests
- C. Traceability can be useful for determining the most suitable test techniques to be used in a testing project
- D. Traceability can be useful to support the needs required by the auditing of testing

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

Traceability is the ability to trace the relationships between the items of the test basis, such as the requirements, the design, the risks, etc., and the test artifacts, such as the test cases, the test results, the defects, etc. Traceability can provide various benefits for the testing process, such as improving the test coverage, the test quality, the test efficiency, and the test communication.

However, not all the statements given are true about the value of maintaining traceability between the test basis and test work products. The statement that is not true is option C, which says that test objectives should be the same for all test levels, although the number of tests designed at various levels can vary significantly. This statement is false, because test objectives are the goals or the purposes of testing, which can vary depending on the test level, the test type, the test technique, the test environment, the test stakeholder, etc. Test objectives can be defined in terms of the test basis, the test coverage, the test quality, the test risk, the test cost, the test time, etc. Test objectives should be specific, measurable, achievable, relevant, and time-bound, and they should be aligned with the project objectives and the quality characteristics. Test objectives should not be the same for all test levels, as different test levels have different focuses, scopes, and perspectives of testing, such as component testing, integration testing, system testing, and acceptance testing. The other statements are true about the value of maintaining traceability between the test basis and test work products, such as:

Traceability can be useful for assessing the impact of a change to a test basis item on the corresponding tests: This statement is true, because traceability can help to identify which tests are affected by a change in the test basis, such as a new requirement, a modified design, a revised risk, etc., and to determine the necessary actions to update, re-execute, or re-evaluate the tests. Traceability can also help to estimate the effort, the cost, and the time needed to implement the change and to verify its impact on the software system.

Traceability can be useful for determining how many test basis items are covered by the corresponding tests: This statement is true, because traceability can help to measure the test coverage, which is the degree to which the test basis is exercised by the test cases. Traceability can help to identify which test basis items are covered, partially covered, or not covered by the tests, and to evaluate the adequacy, the completeness, and the effectiveness of the testing process. Traceability can also help to identify the gaps, the overlaps, or the redundancies in the test coverage, and to prioritize, optimize, or improve the test cases.

Traceability can be useful to support the needs required by the auditing of testing: This statement is true, because traceability can help to provide evidence, documentation, and justification for the testing activities, results, and outcomes. Traceability can help to demonstrate that the testing process follows the standards, the regulations, the policies, and the best practices that are applicable to the software system, the project, or the organization. Traceability can also help to verify that the testing process meets the expectations, the needs, and the satisfaction of the users and the stakeholders. Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 sources and documents:

ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 1.2.2, Testing Policies, Strategies, and Test Approaches¹ ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 2.1.1, Test Planning¹ ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 2.1.2, Test Monitoring and Control¹ ISTQBCertified Tester Foundation Level Syllabus v4.0, Chapter 2.1.3, Test Analysis and Design¹ ISTQBGlossary of Testing Terms v4.0, Traceability, Test Basis, Test Artifact, Test Objective, Test Level, Test Coverage, Test Quality, Test Risk, Test Cost, Test Time²

NEW QUESTION: 117

4 equivalence classes are given for integer values:

$0 < x < 100$

$100 \leq x \leq 200$

$200 < x < 500$

$x \geq 500$

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

A. 50; 100; 200. 1000

B. 0. 1.99, 100.200,201.499, 500;

C. 0.50; 100; 150.200.350.500;

D. 50; 100; 250; 1000

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

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

NEW QUESTION: 118

Match the Work Product with the category it belongs to:

Work Product: 1. Risk register 2 Risk information

3. Test cases

4. Test conditions

Category of work products:

A . Test planning work products

B . Test analysis work products.

C Test design work products

D . Test monitoring and control work products

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

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

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

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

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

NEW QUESTION: 119

Which of the following statements is not correct?

A. Looking for defects in a system may require Ignoring system details

B. Identifying defects may be perceived as criticism against product

C. Looking for defects in system requires professional pessimism and curiosity

D. Testing is often seen as a destructive activity instead of constructive activity

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

* Looking for defects in a system does not require ignoring system details, but rather paying attention to them and understanding how they affect the system's quality, functionality, and usability. Ignoring system details could lead to missing important defects or testing irrelevant aspects of the system.

* Identifying defects may be perceived as criticism against product, especially by the developers or stakeholders who are invested in the product's success. However, identifying defects is not meant to be a personal attack, but rather a constructive feedback that helps to improve the product and ensure its alignment with the requirements and expectations of the users and clients.

* Looking for defects in system requires professional pessimism and curiosity, as testers need to anticipate and explore the possible ways that the system could fail, malfunction, or behave unexpectedly.

Professional pessimism means being skeptical and critical of the system's quality and reliability, while curiosity means being eager and interested in finding out the root causes and consequences of the defects.

* Testing is often seen as a destructive activity instead of constructive activity, as it involves finding and reporting the flaws and weaknesses of the system, rather than creating or enhancing it. However, testing
* is actually a constructive activity, as it contributes to the system's improvement, verification, validation, and optimization, and ultimately to the delivery of a high-quality product that meets the needs and expectations of the users and clients.

NEW QUESTION: 120

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

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

Answer: B ([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: 121

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

I Stubs and drivers are often used

II The test environment should correspond to the production environment III Finding defects is not the main focus IV Testing can be based on use cases V Testing is normally performed by testers

VI Testing for functional and non-functional characteristics

A. Component - VI

Integration - IV

System - I

Acceptance - III

B. Component - IV

Integration - I

System - VI

Acceptance - V

C. Component - I

Integration - V

System - II

Acceptance - IV

D. Component - V

Integration - II

System - IV

Acceptance - VI

Answer: D (LEAVE A REPLY)

The relationship between impact analysis and regression testing in maintenance testing is that impact analysis is used to evaluate the amount of regression testing to be performed.

Maintenance testing is a type of testing that is performed on an existing software product after it has been delivered or deployed, in order to ensure that it still meets its requirements and functions correctly after a change or a modification. Maintenance testing can be triggered by various reasons, such as corrective maintenance (fixing defects), adaptive maintenance (adapting

to new environments), perfective maintenance (improving performance), preventive maintenance (avoiding future problems), etc. Impact analysis is a technique that is used to assess the extent and nature of changes introduced by maintenance activities on the software product or project. Impact analysis helps to identify which parts of the software product are affected by the changes, which parts need to be modified or updated accordingly, which parts need to be retested or verified for correctness or compatibility, etc. Regression testing is a type of testing that verifies that previously tested software still performs correctly after a change or a modification. Regression testing helps to detect any side effects or unintended consequences of maintenance activities on the software product's functionality or quality. Regression testing can be performed at various levels and scopes depending on the impact analysis results. Therefore, in maintenance testing, impact analysis is used to evaluate the amount of regression testing to be performed. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 20.

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (409 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)

NEW QUESTION: 122

Which of the following best describes the way in which statement coverage is measured?

- A.** Measured as the number of decision outcomes executed by the tests, divided by the total number of decision outcomes in the test object.
- B.** It is not possible to accurately measure statement coverage.
- C.** Measured as the number of statements executed by the tests, divided by the total number of executable statements in the code.
- D.** Measured as the number of lines of code executed by the test, divided by the total number of lines of code in the test object.

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

Statement coverage is a metric used in white-box testing that measures the percentage of executable statements in the code that have been executed by the test cases. It is calculated as the number of statements executed by the tests divided by the total number of executable statements in the code, providing an indication of how much of the code has been tested.

NEW QUESTION: 123

The following 4 equivalence classes are given:

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

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

Answer: D (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.

Reference:

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

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

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

NEW QUESTION: 124

Your organization's test strategy states that it is desirable to use more than one method for estimating test effort. You are responsible for estimating test effort for the next project. Based on historical data, the development-to-test effort ratio is 5:3.

The initial estimate for the development effort is 450 person-days.

Which ONE of the following options corresponds to the estimated test effort using the ratio-based method?

- A. 750 person-days
- B. 180 person-days
- C. 720 person-days
- D. 270 person-days

Answer: D (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation:

The ratio-based estimation method relies on historical relationships between development and testing effort.

Using the given 5:3 ratio:

$$\text{Test Effort} = \left(\frac{3}{5} \times \text{Development Effort} \right) = \left(\frac{3}{5} \times 450 \right) = 270 \text{ person-days}$$

Thus, the correct answer is 270 person-days (D).

* (A) and (C) are incorrect as they overestimate the effort.

* (B) underestimates the testing effort based on the ratio.

Ratio-based estimation helps allocate resources effectively based on past project data.

Reference: ISTQB CTFL v4.0 Syllabus, Section 5.1.4 - Effort Estimation in Testing

NEW QUESTION: 125

Which of the following statements is incorrect regarding the involvement of testers in the software development lifecycle (SDLC)?

A. Testers should contribute to all activities in the SDLC and participate in design discussions.

B. Testers should be involved from the beginning of the SDLC to increase understanding of design decisions and detect defects early.

C. Testers should only be involved during the testing phase.

D. Testers' involvement is essential; developers find it difficult to be objective.

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

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: 126

A calculator software is used to calculate the result for 5+6.

The user noticed that the result given is 6.

This is an example of;

A. Mistake

B. Fault

C. Error

D. Failure

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

According to the ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 18, a failure is "an event in which a component or system does not perform a required function within specified limits". In this case, the calculator software does not perform the required function of calculating

the correct result for 5+6 within the specified limits of accuracy and precision. Therefore, this is an example of a failure.

The other options are incorrect because:

* A mistake is "a human action that produces an incorrect result" (page 25). A mistake is not an event, but an action, and it may or may not lead to a failure. For example, a mistake could be a typo in the code, a wrong assumption in the design, or a misunderstanding of the requirement.

* A fault is "a defect in a component or system that can cause the component or system to fail to perform its required function" (page 16). A fault is not an event, but a defect, and it may or may not cause a failure. For example, a fault could be a logical error in the code, a missing specification in the design, or a contradiction in the requirement.

* An error is "the difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition" (page 15). An error is not an event, but a difference, and it may or may not result in a failure. For example, an error could be a rounding error in the calculation, a measurement error in the observation, or a deviation error in the condition.

References = ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 15-18, 25; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 96, page 34.

NEW QUESTION: 127

Which of the following sentences describe a product risk?

- A. The application might not be able to provide the expected responsiveness under a load of up to 300 concurrent users
- B. Failure in acquiring an adequate test automation tool
- C. A wrong configuration of the test environment that causes incidents related to the environment and not to the software under test
- D. The development team lacks knowledge of the technology on which the product is based

Answer: A (LEAVE A REPLY)

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: 128

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

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

What would be your FIRST recommendation to the company?

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

Answer: B (LEAVE A REPLY)

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

Training or replacing the teams responsible for development and testing of the modules containing maximum number of defects may also help improve their skills or performance, but this is not the first recommendation, as it may not be necessary or appropriate for all teams. Verified

References: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer], Chapter 2, page 19; Chapter 4, page 45.

NEW QUESTION: 129

Which of the following statements about test activities that form the overall test process is correct?

- A.** Test implementation realization answers the question of how to perform the test, and the test procedure compares the results of the test with the expected results.
- B.** Test design develops and prioritizes test procedures and may create automated test scripts.
- C.** Test planning determines what is to be tested based on the test basis.
- D.** Test analysis evaluates the test base and the test elements to identify inconsistencies.

Answer: D (LEAVE A REPLY)

Valid ISTQB-CTFL Dumps shared by PrepPdf.com for Helping Passing ISTQB-CTFL Exam! PrepPdf.com now offer the **newest ISTQB-CTFL exam dumps**, the PrepPdf.com ISTQB-CTFL exam **questions have been updated** and **answers have been corrected** get the **newest** PrepPdf.com ISTQB-CTFL dumps with Test Engine here:
<https://www.preppdf.com/ISTQB/ISTQB-CTFL-prepaway-exam-dumps.html> (**409** Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)