Free Questions for CTFL_Syll_4.0 by certsinside

Shared by Ewing on 15-04-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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

Options:

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

Explanation:

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].

Question 2

Question Type: **MultipleChoice**

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

Options:

A- Test planning and control, Test analysis and design, Test implementation and execution, Evaluating ex t criteria and reporting. Test closure activities

B- Test planning, Test analysis and design. Test implementation and control. Checking test coverage and reporting, Test closure activities

C- Test planning and control, Test specification and design. Test implementation and execution, Evaluating test coverage and reporting, Retesting and regression testing, Test closure activities

D- Test planning. Test specification and design. Test implementation and execution. Evaluating exit criteria and reporting. Retesting and test closure activities

Answer:

A

Explanation:

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

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

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

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

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

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

Reference= ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.2, page 15; ISTQB Glossary of Testing Terms, Version 4.0, 2018, pages 37-38; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 88, page 32.

Question 3

Question Type: MultipleChoice

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

Options:

A- The developer

B- The customer

C- The development manager

D- The test leader

Answer:

D

Explanation:

The test leader is the person who is responsible for planning, monitoring, and controlling the test activities and resources in a test project. The test leader should have the best knowledge of the test objectives, scope, risks, resources, schedule, and quality criteria.The test leader should also be aware of the test automation criteria, such as the execution frequency, the test support, the team education, the roles and responsibilities, and the devs and testers collaboration1. Based on these factors, the test leader can decide which tests are

suitable for automation and which are not, and prioritize them accordingly.The test leader can also coordinate with the test automation engineers, the developers, and the stakeholders to ensure the alignment of the test automation strategy with the test project goals and expectations.Reference= 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

Question 4

Question Type: MultipleChoice

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?

Options:

A- It is a product risk since any risk associated with development timeline is a product risk.

B- It is no longer a risk for the Test Manager since an independent party (the sub-contractor) is now managing it

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

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

Answer:

D

Explanation:

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

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

The other options are not correct because:

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.

Reference=

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

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

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

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

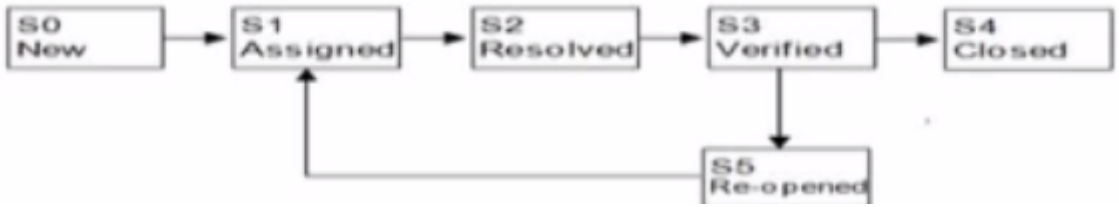
5ISTQB Certified Tester Foundation Level Syllabus v4.0, 2023, p. 101

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

Question 5

Question Type: MultipleChoice

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



Options:

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

Explanation:

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).Reference: ISTQB Certified Tester Foundation Level (CTFL) v4.0 Syllabus, Section 1.4.3, Page 17.

Question 6

Question Type: MultipleChoice

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

Options:

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

Explanation:

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.

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

Question 7

Question Type: MultipleChoice

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

Options:

A- Can't be determined

B- Covering all transitions at least once

C- Covering only start and end states

D- Covering all states at least once

Answer:

B

Explanation:

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

Question 8

Question Type: MultipleChoice

The following 4 equivalence classes are given:

x <= -100
-100 < x < 100
100 <= x < 1000
x >= 1000

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

Options:

A- -100; 100:1000; 1001

B- -500; 0; 100; 1000

C- -99; 99:101; 1001

D- -1000; -100; 100; 1000

Answer:

D

Explanation:

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

(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.

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

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

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

Question 9

Question Type: **MultipleChoice**

In which of the following test documents would you expect to find test exit criteria described9

Options:

A- Test design specification

B- Project plan

C- Requirements specification

D- Test plan

Answer:

D

Explanation:

Test exit criteria are the conditions that must be fulfilled before concluding a particular testing phase.These criteria act as a checkpoint to assess whether we have achieved the testing objectives and are done with testing1. Test exit criteria are typically defined in the test plan document, which is one of the outputs of the test planning phase. The test plan document describes the scope, approach, resources, and schedule of the testing activities.It also identifies the test items, the features to be tested, the testing tasks, the risks, and the test deliverables2.According to the ISTQB Certified Tester Foundation Level Syllabus v4.0, the test plan document should include the following information related to the test exit criteria3:

The criteria for evaluating test completion, such as the percentage of test cases executed, the percentage of test coverage achieved, the number and severity of defects found and fixed, the quality and reliability of the software product, and the stakeholder satisfaction.

The criteria for evaluating test process improvement, such as the adherence to the test strategy, the efficiency and effectiveness of the testing activities, the lessons learned and best practices identified, and the recommendations for future improvements.

Therefore, the test plan document is the most appropriate test document to find the test exit criteria described. The other options, such as test design specification, project plan, and requirements specification, are not directly related to the test exit criteria.The test design specification describes the test cases and test procedures for a specific test level or test type3.The project plan describes the overall objectives, scope, assumptions, risks, and deliverables of the software project4.The requirements specification describes the functional and non-functional requirements of the software product5. None of these documents specify the conditions for ending the testing process or evaluating the testing outcomes.Reference=ISTQB Certified Tester Foundation Level Syllabus v4.0,Entry and Exit Criteria in Software Testing | Baeldung on Computer Science,Entry And Exit Criteria In Software Testing - Rishabh Software,Entry and Exit Criteria in Software Testing Life Cycle - STLC [2022 Updated] - Testsigma Blog,ISTQB releases Certified Tester Foundation Level v4.0 (CTFL).

Question 10

Question Type: MultipleChoice

A test manager decided to skip static testing since he believes bugs can be found easily by doing dynamic testing. Was this decision right or wrong?

Options:

A- The decision was wrong. Ensuring quality mandates that static testing is performed after performing the dynamic testing.

B- The decision was right. Static testing is usually redundant if a product is planned to go through a full-cycle of dynamic testing.

C- The decision was right. Most of the bugs are easier to identify during the dynamic testing.

D- The decision was wrong. Static testing can find defects early in the development process, reducing the overall cost of testing and development

Answer:

D

Explanation:

Static testing is a form of testing that does not involve executing the software or system under test. It includes activities such as reviews, inspections, walkthroughs, and analysis of documents, code, and models. Static testing can find defects early in the development process, before they become more expensive and difficult to fix in later stages. Static testing can also improve the quality of the software or system by preventing defects from being introduced in the first place. Static testing can complement dynamic testing, which involves executing the software or system under test and checking the results against expected outcomes. Dynamic testing can find defects that static testing may miss, such as performance, usability, or integration issues. However, dynamic testing alone is not sufficient to ensure quality, as it may not cover all possible scenarios, inputs, or paths. Therefore, a test manager who decides to skip static testing is making a wrong decision, as he or she is ignoring the benefits of static testing and relying solely on dynamic testing, which may not be effective or efficient enough to find and prevent defects.Reference= ISTQB Certified Tester Foundation Level Syllabus, Version 4.0, 2018, Section 2.1.1, page 14; ISTQB Glossary of Testing Terms, Version 4.0, 2018, page 36; ISTQB CTFL 4.0 - Sample Exam - Answers, Version 1.1, 2023, Question 3, page 9.

Question 11

Question Type: MultipleChoice

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

Options:

A- Static techniques can be used before all code is ready for execution

B- Static techniques find more detects then dynamic techniques.

C- Static techniques can be used by inexperienced users.

D- Static techniques are always cheaper than dynamic techniques.

Answer:

A

Explanation:

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.Reference= 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