

Lead2Passed



Lead2Passed

HOME

ALL VENDORS

★ GUARANTEE

? FAQ

TESTIMONIALS

Login / Register My Shopcart (1)

Input your exam code ...



Try before you buy

Download a free sample of any of our exam questions and answers

- ✓ Online Test Engine: Online Tool, Convenient, easy to study. Instant Online Access. Supports All Web Browsers.
- ✓ PDF format: Easy to read and print learning materials, our products are available in PDF file format.
- ✓ Desktop Test Engine: Installable Software Application. Simulates Real Exam Environment. Practice Offline Anytime.



Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.



365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.



Money Back Guarantee

Full refund if you fail the corresponding exam in 60 days after purchasing. And Free get any another product.



Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

<http://www.lead2passed.com>

Valid Certification Exam Dumps Materials and Study Guide -
Lead2Passed

Exam : **CTFL_Syll2018**

Title : ISTQB Certified Tester
Foundation Level (Syllabus
2018)

Vendor : ISQI

Version : DEMO

NO.1 Which test design technique can be used for effectively testing system requirements consisting of logical conditions?

- A. Boundary value analysis
- B. Use case testing
- C. Equivalence partitioning
- D. Decision table testing

Answer: D

Explanation of Correct answer: Decision table testing is a test design technique that can be used for effectively testing system requirements consisting of logical conditions. Decision table testing is based on the idea that the system behavior can be represented by a table, where each column corresponds to a possible combination of inputs and each row corresponds to the expected outputs for that combination. Decision table testing helps to identify and cover all possible scenarios and outcomes for a given requirement¹².

References: 1: Certified Tester Foundation Level Syllabus, Version 2018 v3.1.1, Section 4.3.2 2: A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Section 4.3.2

NO.2 Typically, exit criteria may consist of:

- A. Defining the amount, level of detail structure, and templates for the test documentation.
- B. Estimates of defect density or reliability measures.
- C. Adequacy of the test approaches taken.
- D. Discussions on disaster recovery.

Answer: B

Explanation:

These are examples of exit criteria, which are the conditions that must be met before testing can be completed.

Exit criteria may also include test coverage measures, residual risk assessment, or stakeholder approval. The other options are not exit criteria, but rather test planning activities.

NO.3 When an organization considers the use of testing tools, they should:

- A. Use a tool in order to help define a good test process because the tool will force process repeatability and therefore enforce good test process.
- B. Always start by bringing in automated test execution tools as these tools have the greatest return on investment and therefore should be introduced first.
- C. Perform analysis of the test process and then assess whether it can be supported through the introduction of tool support.
- D. Allow the developers to select the testing tools because tools are technical and developers have the appropriate skills to advise on test tool selection and configuration.

Answer: C

Explanation:

When an organization considers the use of testing tools, they should C. Perform analysis of the test process and then assess whether it can be supported through the introduction of tool support.

Testing tools can provide many benefits to the test process, such as increased efficiency, effectiveness, consistency, quality, etc.

However, testing tools also have some challenges and risks associated with them, such as cost,

learning curve, compatibility, maintenance, etc. Therefore, before introducing any testing tool, an organization should analyze their current test process and identify their needs, goals, expectations, constraints, etc., and then evaluate whether a tool can support them or not. A detailed explanation of testing tools can be found in Software Testing Foundations: A Study Guide for the Certified Tester Exam, pages 193-1983.

NO.4 Which of the following statements gives the BEST example of a trigger for maintenance testing?

- A. Retirement of a customer service application
- B. Business approval to fund a project to develop an e-commerce web application
- C. Completion of the requirements specifications for an application
- D. Completion of the designs for an application

Answer: A

Explanation:

Maintenance testing is triggered by changes, such as migration, retirement, or enhancement of a system¹.

Therefore, option A is the best example of a trigger for maintenance testing. Options B, C, and D are examples of triggers for development testing, not maintenance testing¹. References: 1, Section 2.3.2

NO.5 What is integration testing?

- A. Looking for faults in larger components or subsystems
- B. Another term for testing system integrity
- C. Specifying which components to integrate in which order
- D. Testing that the interfaces work correctly

Answer: D

Explanation:

Integration testing is the process of testing the interactions between different components or subsystems of a system. Integration testing verifies that the interfaces work correctly, meaning that they pass data and control correctly between components or subsystems, and that they handle errors and exceptions properly. Integration testing can also verify the functionality and quality of the integrated system or subsystem.

References: Certified Tester Foundation Level Syllabus, Section 5.1.

NO.6 Which of the following sentences about testing and debugging is correct?

- A. Re-testing checks that debugging has found and analyzed the failure
- B. Dynamic testing finds defects, while debugging removes failures
- C. Dynamic testing reveals failures, while debugging removes defects
- D. Like most development activities, debugging is usually done before testing starts

Answer: C

Explanation:

Testing and debugging are two different activities that are related to finding and removing defects and failures in software. Testing is the process of evaluating software by applying test cases to find failures and provide information on its quality. Debugging is the process of finding, analyzing, and removing the causes of failures in software. Testing reveals failures, which are deviations of the actual behavior of the software from its expected behavior. Debugging removes defects, which are

flaws in the software that cause failures.

References: Certified Tester Foundation Level Syllabus, Section 1.2.1

NO.7 Which of the following processes ensures that all items of testware are identified, version controlled, tracked for changes, so that traceability can be maintained throughout the test process?

- A.** Software traceability process
- B.** Incidence management process
- C.** Testing design process
- D.** Configuration management process

Answer: D

Explanation:

The process that ensures that all items of testware are identified, version controlled, tracked for changes, so that traceability can be maintained throughout the test process is D. Configuration management process.

Configuration management is a process that establishes and maintains the integrity and consistency of all items of testware throughout the test process. Configuration management involves identifying, storing, controlling, tracking, and auditing all items of testware (such as test plans, test cases, test scripts, test data, test results, etc.) and their versions and changes. Configuration management helps to ensure that only authorized and approved items of testware are used for testing and that traceability can be maintained between them. A detailed explanation of configuration management can be found in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, pages 101-1021.

NO.8 Which of the following software development models BEST exemplifies a model that does NOT support the principle of early testing?

- A.** The Waterfall model
- B.** The V-model
- C.** The Incremental development model
- D.** The Iterative development model

Answer: A

Explanation:

According to the syllabus, a software development life cycle (SDLC) model is a conceptual framework that describes the phases and activities involved in developing a software product. Different SDLC models have different advantages and disadvantages depending on the project context and objectives. The principle of early testing states that testing activities should start as early as possible in the software development life cycle and be focused on defined objectives. Early testing helps to prevent defects, reduce rework, lower costs, and improve quality. The answer A is correct because it best exemplifies a model that does not support the principle of early testing. The waterfall model is a sequential SDLC model that divides the development process into distinct phases, such as requirements analysis, design, implementation, testing, and maintenance.

Each phase must be completed before the next phase can begin, and there is no overlap or iteration between phases. Testing is done only after the implementation phase, which means that defects are detected late in the development cycle and are more expensive and difficult to fix. The other answers are incorrect because they exemplify models that support the principle of early testing. The V-model is an extension of the waterfall model that emphasizes verification and validation activities at each

phase of development. Testing is done in parallel with each corresponding development phase, which means that defects are detected early and feedback is provided to the developers. The incremental development model is an iterative SDLC model that divides the development process into smaller increments or iterations, each delivering a working software product or a subset of features. Testing is done at the end of each iteration, which means that defects are detected early and feedback is provided to the developers. The iterative development model is another iterative SDLC model that repeats the development process for each iteration, with each iteration producing an improved version of the software product or a subset of features. Testing is done throughout each iteration, which means that defects are detected early and feedback is provided to the developers. References: Certified Tester Foundation Level Syllabus, Section 1.1.1, page 9-10.

NO.9 Which activity in the fundamental test process includes test script generation?

- A. Test closure activities
- B. Test analysis and design
- C. Test planning and control
- D. Test implementation and execution

Answer: D

Explanation:

Test implementation and execution is the activity in the fundamental test process that includes test script generation. Test script generation is the process of creating executable test cases from test conditions and test data² defines this activity as follows:

Test implementation and execution has the following major tasks:

- * Develop and prioritize test cases, creating test data and writing test procedures.
- * Check test environment has been set up correctly.
- * Execute test cases, evaluate results and document deviations from expected results.
- * Use exit criteria to report on suitability of system under test.

Test closure activities (A) include finalizing and archiving test results, evaluating the test process, identifying areas for improvement, and celebrating achievements. Test analysis and design (B) include reviewing test basis, identifying test conditions, designing high-level test cases, and defining exit criteria. Test planning and control include defining test objectives, scope, strategy, resources, schedule, risks, and metrics.

NO.10 Which of the following statements contradicts the general principles of testing?

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

Answer: A

Explanation:

The general principles of testing state that testing can show the presence of defects, but not their absence.

Running the same test set more often will not increase the likelihood of finding new defects, unless the system or its environment changes. Therefore, statement A contradicts the general principles of testing. Statement B is true, as testing is context-dependent and should be tailored to the specific situation of a project. Statement C is also true, as early testing can help prevent defects and reduce

rework. Statement D is true, as it reflects the Pareto principle or the 80/20 rule, which states that most defects are found in a small subset of a system's modules.

References: ISTQB Certified Tester Foundation Level Syllabus 2018, Section 1.2

NO.11 Testing should provide sufficient information to stakeholders to make informed decisions about the release of the software or system being tested. At which of the following fundamental test process activity the sufficiency of the testing and the resulting information are assessed?

- A. Implementation and execution
- B. Requirements specification
- C. Evaluating exit criteria and reporting.
- D. Analysis and design

Answer: C

Explanation:

The fundamental test process activity where the sufficiency of testing and resulting information are assessed is Evaluating exit criteria and reporting. This activity involves checking whether the test objectives have been met and whether there are any unresolved issues or risks that could affect the release or deployment decision.

This activity also involves preparing and communicating a test summary report that summarizes the test activities and results and provides recommendations and feedback for improvement. You can find more information about Evaluating exit criteria and reporting in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 3, Section 3.5.

NO.12 Which of the following lists represents the correct sequence of the main activities of the fundamental test process (leaving out the activity of control which should take place in parallel to all the other activities)?

- A. Planning, analysis and reporting, design and implementation, execution, test closure activities, evaluating exit criteria.
- B. Planning, analysis, design and implementation, execution, logging, test closure activities, evaluating exit criteria.
- C. Planning, analysis and design, execution, logging and reporting, regression testing
- D. Planning, analysis and design, implementation and execution, evaluation exit criteria and reporting, test closure activities

Answer: D

NO.13 The following sentences refer to the Standard for Software Test Documentation' specification (IEEE 829).

Which sentence is correct?

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

Answer: D

Explanation:

The 'Standard for Software Test Documentation' specification (IEEE 829) is a standard that defines a set of documents that can be used to document the test process and its outcomes. The standard provides an outline for each document, specifying its purpose, content, and format. However, the standard does not prescribe how to apply it in different contexts or projects. It is up to each organization or project to decide how to adapt the standard to their specific needs and situation. Therefore, the standard is not a rigid or mandatory requirement that must be followed strictly by all testers. Rather, it is a flexible and adaptable guideline that can be used as a reference or a starting point for creating test documentation regimes. You can find more information about IEEE 829 and test documentation in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 51.

NO.14 Which of the following statements describes regression testing?

I. Retesting of a fixed defect

II Testing of an already tested program

III. Testing of new functionality in a program

IV. Regression testing applies only to functional testing

V Tests that do not have to be repeatable. because they are only used once

A. II

B. II, IV, V

C. I,IV

D. I, III, IV

Answer: A

Explanation:

Regression testing is testing of an already tested program after modification to discover any defects introduced or uncovered as a result of changes in software or its environment. Regression testing ensures that previously working functionality still works after changes are made³ defines regression testing as follows:

Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

Regression Testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

I, III, IV, and V are incorrect statements about regression testing. Retesting of a fixed defect (I) is not regression testing, but confirmation testing or rework testing, which verifies that a specific defect has been resolved. Testing of new functionality in a program (III) is not regression testing, but functional testing or new feature testing, which verifies that new requirements are met by new code.

Regression testing applies only to functional testing (IV) is not true, as regression testing can also apply to non-functional aspects such as performance or security. Tests that do not have to be repeatable, because they are only used once (V) are not regression testing, but exploratory testing or ad hoc testing, which are based on learning and discovery rather than predefined test cases.

NO.15 The ISTQB fundamental test process consists of 5 main activities To which of these belongs the task

"Identifying necessary test data"?

- A. Evaluating test criteria and reporting
- B. Test implementation and execution
- C. Test planning and control
- D. Test analysis and design

Answer: D

Explanation:

Test analysis and design is the activity in the fundamental test process that includes identifying necessary test data. Test data are the inputs that are used to execute the test cases and verify the expected results² defines this activity as follows:

Test analysis and design has the following major tasks:

- * Reviewing the test basis (such as requirements, risk analysis reports, design documents or code).
- * Identifying test conditions based on analysis of test items, specifications, behavior and structure of the software.
- * Designing high-level test cases based on test conditions and designing techniques.
- * Evaluating testability of requirements and system under test.
- * Defining exit criteria.

Evaluating exit criteria and reporting (A) is part of the test closure activities, where the results of testing are evaluated against the defined objectives. Test implementation and execution (B) is where the test cases are executed using the identified test data and deviations from expected results are documented. Test planning and control is where the overall approach and resources for testing are defined and monitored.

NO.16 Which TWO of the following test tools would be classified as test execution tools? [K2] a. Test data preparation tools b. Test harness c. Review tools d. Test comparators e. Configuration management tools

- A. a and b
- B. c and d
- C. c and e
- D. b and d

Answer: D

Explanation:

The test tools that would be classified as test execution tools are D. b and d. Test execution tools are tools that automate the execution of test cases or test scripts, and compare the actual results with the expected results.

Test execution tools can also record and replay user actions, generate test data, and report test results. Test harness and test comparators are examples of test execution tools. A test harness is a tool that creates a test environment for a component or system under test, by simulating the required dependencies, such as stubs, drivers, or mock objects. A test comparator is a tool that compares the actual outputs of a component or system under test with the expected outputs, and reports any differences or anomalies. A detailed explanation of test execution tools can be found in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, pages 111-1121.

NO.17 When a test case is created based on a Use Case, what type of test is created?

- A. Structural Test

- B. Functional Test
- C. Performance test
- D. Regression Test

Answer: B

Explanation:

Functional test is the type of test that is created when a test case is based on a use case. A use case is a description of how a system interacts with one or more actors (users or other systems) to achieve a specific goal or function. A functional test is a test that verifies that a system or software component performs its specified functions according to its requirements. Functional tests can be derived from use cases by identifying test scenarios and test cases that cover the main flow and alternative flows of each use case² explains this as follows:

Use cases are one of the most commonly used techniques for analyzing and modeling functional requirements for a system. A use case describes how an actor interacts with a system to accomplish a specific goal.

Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions are tested by feeding them input and examining the output. Use cases can be used as a source for deriving functional tests by identifying test scenarios and test cases that cover the main flow and alternative flows of each use case.

A, C, and D are incorrect answers. Structural test, performance test, and regression test are not types of tests that are created when a test case is based on a use case. Structural test is a type of test that is based on the internal structure and logic of the code rather than the functionality or requirements. Performance test is a type of test that measures the speed, responsiveness, scalability, or stability of a system under various workloads or conditions. Regression test is a type of test that verifies that previously working functionality still works after changes are made to the system or its environment.

NO.18 "Experience based" test design techniques, typically...

- A. Use decision tables to generate the Boolean test conditions to be executed.
- B. Identify the structure of the system or software at the component, integration or system level.
- C. Use the skill, intuition and experience of the tester to derive the test cases, using error guessing and exploratory testing.
- D. Establish traceability from test conditions back to the specifications and requirements.

Answer: C

Explanation:

Experience-based test design techniques are techniques that use the skill, intuition, and experience of testers to derive test cases, using error guessing and exploratory testing¹. Error guessing is a technique that uses common sense and previous experience to guess where defects might occur in a system¹. Exploratory testing is an approach that involves simultaneous learning, test design, and test execution¹. Experience-based test design techniques are typically used when there is insufficient information or time to apply other more formal techniques¹. They do not use decision tables, identify the structure of the system or software, or establish traceability from test conditions back to the specifications and requirements.

NO.19 Which of the following would typically be identified using static analysis by tools? [K1]

- A. Spelling mistake on an error message
- B. A potential infinite loop

- C. Memory leakage
- D. A variable set to the wrong value

Answer: B

Explanation:

A potential infinite loop is a type of defect that can be typically identified using static analysis by tools¹. Static analysis is a technique that examines the source code or other software artifacts without executing them, and can detect defects, vulnerabilities, code smells, and deviations from standards early in the development process¹. Static analysis tools are software tools that automate the static analysis technique and provide various features and functionalities to support it¹. Static analysis tools can identify a potential infinite loop by analyzing the control flow and data flow of the source code and checking for conditions or statements that may cause an endless repetition or iteration¹. A potential infinite loop can cause serious problems in software performance, functionality, reliability, and security¹. Therefore, a potential infinite loop is a type of defect that can be typically identified using static analysis by tools.

NO.20 Which of the following would be appropriate test objectives for user acceptance testing of the first release of a new software product aimed at a general market and built using Agile methods? [K2] a. To identify as many defects as possible b. To maximise code coverage c. To ensure the product works as expected d. To assess the overall quality of the product e. To determine the reliability of the product

- A. b and c
- B. a and d
- C. b and e
- D. c and d

Answer: D

Explanation:

* The test objectives for user acceptance testing of the first release of a new software product aimed at a

* general market and built using Agile methods are D. c and d. User acceptance testing is a level of testing that involves users or customers validating that the software product meets their needs and expectations before accepting it for use or deployment. User acceptance testing can have different objectives depending on the context and purpose of the software product. For a new software product aimed at a general market and built using Agile methods, some possible objectives are:

* To ensure the product works as expected, which means verifying that the product meets the functional and non-functional requirements agreed upon by the users or customers and the development team during each iteration or sprint of the Agile development process.

* To assess the overall quality of the product, which means evaluating how well the product satisfies the quality attributes that are important for the users or customers, such as usability, performance, reliability, security, etc.

A detailed explanation of user acceptance testing can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 79-80.

NO.21 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: A

Explanation:

Provides developers with information to isolate the failure is the incident report objective that this excerpt satisfies, because it gives details about how to recreate the failure using a specific test file and location . An incident report is a document that records any event that deviates from the expected or desired behaviour of the software under test . One of its objectives is to provide developers with information to isolate the failure, which means to identify and locate the cause of the failure in the software . The other options are not incident report objectives that this excerpt satisfies. Option B is wrong, because this excerpt does not belong to an incident report, but rather to a test report. A test report is a document that summarizes the results and outcomes of testing activities . Option C is wrong, because this excerpt does not provide project managers with information on the project risks, which means to identify and assess the potential threats or uncertainties that may affect the project objectives or outcomes . Option D is wrong, because this excerpt does not provide test managers with information to report test progress, which means to measure and communicate how much of the planned testing activities have been completed or remain to be done .

NO.22 Consider the following pseudo code

1. Begin
2. Read Gender
3. __Print "Dear"
4. If Gender = 'female'
5. Print ("Ms")
6. Else
7. __Print ("Mr")
8. Endif
9. End

How many test cases are needed to achieve 100 per cent decision coverage?

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

Answer: B

Explanation:

Decision coverage is a structure-based test technique that involves testing every decision point in the code by testing all possible outcomes or branches of each decision point. Decision coverage requires some knowledge of the internal structure or implementation of the software system; it focuses on how the system does what it does rather than what it does.

The pseudo code given in the question has one decision point at line 4, where the value of Gender is compared to 'female'. This decision point has two possible outcomes or branches:

- * True branch: If Gender = 'female', then print "Ms" at line 5.
- * False branch: If Gender != 'female', then print "Mr" at line 7.

To achieve 100 per cent decision coverage, we need to test both outcomes or branches of the decision point at least once. Therefore, we need two test cases that cover both scenarios:

* TC1: Gender = 'female' -> Print "Dear Ms"

* TC2: Gender = 'male' -> Print "Dear Mr"

The other options are not correct because they do not provide enough or too many test cases to achieve 100 per cent decision coverage. For example:

* A. 1: This option is not enough because it only provides one test case that covers one outcome or branch of the decision point, but not the other.

* C. 3: This option is too many because it provides three test cases that cover both outcomes or branches of the decision point, but one of them is redundant or unnecessary.

* D. 4: This option is too many because it provides four test cases that cover both outcomes or branches of the decision point, but two of them are redundant or unnecessary.

You can find more information about decision coverage and structure-based testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 4, Section 4.3.

NO.23 Which of the following accurately defines the integration testing test level? [K2]

A. Test basis includes software and system design, test objects include interfaces, and tests concentrate on the interactions between different parts of a system

B. Test basis includes component requirements, test objects include database modules, and tests concentrate on the behaviour of the system as a whole.

C. Test basis includes business processes, test objects include system configuration and configuration data, and tests concentrate on establishing confidence in the system

D. Test basis includes use cases, test objects include user procedures and tests concentrate on a high level model of system behaviour

Answer: A

Explanation:

Integration testing is a test level that accurately defines the integration testing test level¹. Integration testing is a type of testing that verifies that different components or systems work together correctly and consistently¹. The test basis for integration testing includes software and system design documents that specify how the components or systems are integrated and interact with each other¹. The test objects for integration testing include interfaces between components or systems that enable data exchange or communication¹. The tests for integration testing concentrate on the interactions between different parts of a system and check for functional, performance, reliability, security, compatibility, and interoperability issues¹. Therefore, integration testing is a test level that accurately defines the integration testing test level.

NO.24 Which of the following is a possible reason for introducing a defect in software code?

A. Rushing to meet a tight deadline to turn code over for testing

B. Improper system testing

C. Improper unit testing

D. Focus on static testing over dynamic testing

Answer: A

Explanation:

According to the syllabus, a defect is a flaw in a component or system that can cause it to fail to perform its required function. A defect can be introduced in any phase of the software development

life cycle, such as requirements, design, coding, testing, or deployment. A possible reason for introducing a defect in software code is rushing to meet a tight deadline to turn code over for testing. This can lead to careless mistakes, incomplete functionality, or poor quality code. The answer A is correct because it is an example of a possible reason for introducing a defect in software code. The other answers are incorrect because they are not reasons for introducing defects in software code, but rather consequences or detection methods of defects.

References: [Certified Tester Foundation Level Syllabus], Section 1.3.1, page 12-13.

NO.25 Which of the following BEST distinguishes the terms "validation" and "verification"?

- A.** Validation is confirmation through the provision of objective evidence that the specified requirements have been met while verification is confirmation through the provision of objective evidence that the requirements for a specific intended use have been met
- B.** Verification is confirmation through the provision of subjective evidence that the specified requirements have been met while validation is confirmation through the provision of subjective evidence that the designs for a specific intended use have been met
- C.** Validation is confirmation through the provision of subjective evidence that the specified requirements have been met while verification is confirmation through the provision of subjective evidence that the designs for a specific intended use have been met
- D.** Verification is confirmation through the provision of objective evidence that the specified requirements have been met while validation is confirmation through the provision of objective evidence that the requirements for a specific intended use have been met

Answer: D

Explanation:

According to the syllabus, verification is the process of evaluating a product or component to determine whether it satisfies its specified requirements. Validation is the process of evaluating a product or component to determine whether it fulfills its intended use and user expectations. Verification answers the question "Are we building the product right?" while validation answers the question "Are we building the right product?" Both verification and validation involve providing objective evidence, which means factual, unbiased, and observable information that supports the evaluation. The answer D is correct because it best distinguishes the terms "validation" and "verification". The other answers are incorrect because they either confuse the terms "validation" and "verification" or use subjective evidence, which means personal, biased, or unobservable information that does not support the evaluation.

References: [Certified Tester Foundation Level Syllabus], Section 1.2.1, page 10-11.

NO.26 A client-server system for a web development must support a minimum of 200 enquiries per hour. In peak times, it must be available 24 hours x 7 days due to the critical nature of the application, and must have a response time lower than 20 seconds during peak loads. Which of the following set of test types would be most appropriate to verify the non-functional requirements expressed in this statement?

- A.** Performance, Usability, Regression.
- B.** System, Load, Fail-over.
- C.** Performance, Load, Stress.
- D.** Load, Volume and Component.

Answer: C

Explanation:

Performance, Load, Stress. Performance testing is a type of non-functional testing that measures how well the system performs under various conditions, such as response time, throughput, resource utilization, etc. Load testing is a type of performance testing that simulates a high volume of user requests or transactions on the system and measures its behavior under normal and peak load conditions. Stress testing is a type of performance testing that simulates extreme or abnormal conditions on the system and measures its behavior under stress or overload conditions. These test types would help verify the requirements such as minimum number of enquiries per hour, availability, and response time under peak loads. A detailed explanation of performance, load, and stress testing can be found in *Software Testing Foundations: A Study Guide for the Certified Tester Exam*, pages 187-1913.

NO.27 Which of the following is a factor that could negatively influence the success of the deployment of a tool within an organization?

- A.** Introducing the tool to all the test teams of the organization at the same time
- B.** Collecting usage information of the tool from the early projects.
- C.** Providing training and coaching for new users of the tool
- D.** Adapting and improving the processes to the usage of the tool

Answer: A

Explanation:

Introducing the tool to all the test teams of the organization at the same time is a factor that could negatively influence the success of the deployment of a tool within an organization, as it may cause resistance, confusion, or overload among the test teams. A better approach would be to introduce the tool gradually and incrementally, starting with a pilot project or a small group of users, and then expanding to other test teams based on the feedback and results. suggests this as follows:

A pilot project should be conducted before introducing a new test tool into an organization in order to learn more about how to use it effectively and efficiently in your context and how it will interact with other tools and processes. A pilot project should involve a small group of users who are willing and able to experiment with the tool and provide feedback. The pilot project should have clear objectives, scope, duration, and success criteria.

B, C, and D are factors that could positively influence the success of the deployment of a tool within an organization. Collecting usage information of the tool from the early projects (B) can help to evaluate the benefits and drawbacks of the tool, identify areas for improvement, and justify the investment. Providing training and coaching for new users of the tool can help to increase their confidence and competence in using the tool, reduce errors and frustration, and enhance their productivity and satisfaction. Adapting and improving the processes to the usage of the tool (D) can help to optimize the workflow and performance of the test teams, align the tool with the organizational goals and standards, and increase the value and quality of the testing activities.

NO.28 Which of the following test techniques is structure-based?

- A.** Control flow testing
- B.** Use case testing
- C.** State transition testing
- D.** Decision table testing

Answer: A

Explanation:

Test techniques are methods or procedures that can be used to design, execute, or evaluate test cases. Test techniques can be classified into two categories: specification-based and structure-based. Specification-based test techniques, also known as black-box test techniques, are based on the requirements, specifications, or expectations of the system under test. They do not require any knowledge of the internal structure or implementation of the system. Some examples of specification-based test techniques are use case testing, state transition testing, decision table testing, etc. Structure-based test techniques, also known as white-box test techniques, are based on the code, architecture, or design of the system under test. They require some knowledge of the internal structure or implementation of the system. Some examples of structure-based test techniques are control flow testing, data flow testing, branch testing, statement testing, etc. You can find more information about test techniques in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 41.

NO.29 Which one of the following is an example of how product risk analysis can influence the testing approach?

- A.** The complex business rules could result in costly failures, so decision table testing will be used for test case design.
- B.** Performance failures were much lower than expected, so more test analysis will be conducted in this area.
- C.** There is a lack of automation skills in the test team, so training for the automation tool will be rolled out.
- D.** There are no product risks recorded around security, so security testing will be given priority as a contingency measure.

Answer: A

Explanation:

Product risk analysis is an approach to testing that aims to reduce the level of product risks and inform stakeholders of their status, starting in the initial stages of a project¹. It involves the identification of product risks and the use of risk levels to guide the test process¹. Product risks are uncertain situations that can affect the quality or value of the software product². One of the factors that can influence the testing approach is the test design technique to be applied³. Test design techniques are methods to derive and select test cases based on the test objectives, test conditions, and test basis². Decision table testing is a test design technique that is suitable for testing complex business rules that have logical conditions and outcomes². Therefore, if the product risk analysis identifies that the complex business rules could result in costly failures, then decision table testing will be used for test case design to cover all the possible combinations of conditions and outcomes². The other options are not examples of how product risk analysis can influence the testing approach, because they are either not related to product risks or not based on risk levels. They are:

* Performance failures were much lower than expected, so more test analysis will be conducted in this area (B). This is not related to product risk analysis, but to test monitoring and control, which is the process of comparing actual progress against the plan and reporting the status². Test analysis is the activity of analyzing the test basis and defining test objectives². If the performance failures were much lower than expected, then more test execution or evaluation might be needed, not more test analysis.

* There is a lack of automation skills in the test team, so training for the automation tool will be rolled out. This is not related to product risk analysis, but to test management, which is the process

of planning, monitoring, and controlling the test activities and resources². Test automation is the use of software to perform or support test activities². The lack of automation skills in the test team is a project risk, not a product risk, and it should be addressed by providing adequate training, tools, and resources for the test team².

* There are no product risks recorded around security, so security testing will be given priority as a contingency measure (D). This is not based on risk levels, but on the absence of risk identification, which is a poor practice in product risk analysis. Security testing is a type of testing that determines whether the software protects data and maintains functionality as intended². Product risk analysis should identify the potential security risks and assign them appropriate risk levels based on their probability and impact¹. Security testing should be prioritized based on the risk levels, not on the lack of risk identification, which could lead to overlooking or underestimating the security risks¹.

References:

- * ISTQB Foundation Level 2018 syllabus
- * Risk in Software Testing
- * Risk-Based Testing
- * Test Approach

NO.30 Test script TransVal 3.1 tests transaction validation via screen TRN 003B. According to the specification (PID ver 1.3 10b iv) the validation screen should not accept future dated transactions. Test script TransVal 3.1 passes. Test script eod 1.4 tests end of day processing and is run after the execution of TransVal 3.1 using data entered during that test Which of the following is the BEST detail on an incident report? [K3]

A. Title. End of Day failure.. Reproducible. Yes. Description. Script eod 1.4 fails when the first transaction of the day is a future dated transaction. Screen shot of the failure attached.

B. Title. Transaction input screen validation..Reproducible. Yes. Description. Script eod 1..4 fails . Screen shot of the failure attached. Validation of transaction entry on screen TRN-003B should not allow future dated transactions - see PID ver 1.3 para 10b iv.

C. Title. Screen TRN-003B validation of transaction date.. Reproducible. No. Description. When a future dated transaction is processed by the end of day process, a failure can occur. This does not always happen. Screen shot of the failure attached.

D. Title. Screen TRN-003B validation of transaction date. Reproducible. Yes. Description. Script eod 1.4 fails when the first transaction of the day is a future dated transaction. Screen shot of the failure attached. Validation of transaction entry on screen TRN-003B should not allow future dated transactions

- see PID ver 1.3 para 10b iv.

Answer: D

Explanation:

* An incident report is a document that records any event occurring during testing that requires investigation¹. An incident report should contain sufficient information to enable reproduction of the incident and resolution of the defect¹. According to IEEE 829 Standard for Software Test Documentation, an incident report should contain the following information:

- * Identifier: A unique identifier for the incident report
- * Summary: A brief summary of the incident
- * Incident description: A description of the incident, including:
- * Date: The date when the incident was observed

- * Author: The name of the person who reported the incident
 - * Source: The software or system lifecycle process in which the incident was observed
 - * Version: The identification of configuration items of the software or system
 - * Test case: The identification of the test case that caused the incident
 - * Execution phase: The phase of test execution when the incident was observed
 - * Environment: The hardware and software environment in which the incident was observed
 - * Description: A description of the anomaly to enable reproduction of the incident
 - * Expected result: The expected result of the test case
 - * Actual result: The actual result of the test case
 - * Reproducibility: An indication of whether the incident can be reproduced or not
 - * Impact analysis: An analysis of the impact of the incident on other aspects of the software or system
 - * Incident resolution: A description of how the incident was resolved, including:
 - * Resolution date: The date when the incident was resolved
 - * Resolver: The name of the person who resolved the incident
 - * Resolution summary: A brief summary of how the incident was resolved
 - * Status: The current status of the incident (e.g., open, closed, deferred)
 - * Classification information: A classification of the cause and effect of the incident for metrics and reporting purposes
- Therefore, among the options given in this question, only D provides the best detail on an incident report. It contains a clear title, a reproducibility indicator, a description that includes both expected and actual results, a reference to the specification document, and a screen shot of the failure. The other options are either missing some important information or providing inaccurate or irrelevant information

NO.31 A company purchased a new system which deals with all financial transactions in the company Which test types call for involvement of an expert from the financial department?

- A.** Component testing
- B.** Acceptance tests
- C.** Maintenance testing
- D.** System tests

Answer: B

Explanation:

Acceptance tests are the test types that call for involvement of an expert from the financial department for a new system that deals with all financial transactions in a company. Acceptance tests are tests conducted to determine if the requirements of a specification or contract are met by a system or software component prior to its delivery or deployment. Acceptance tests are usually performed by end users or customers who have domain knowledge and expertise in evaluating if the system meets their needs and expectations¹ defines acceptance tests as follows:

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Acceptance Testing is also known as User Acceptance Testing (UAT), End-User Testing, Operational Acceptance Testing (OAT) or Field (Acceptance) Testing.

Acceptance Testing is performed by end users or customers who have domain knowledge and expertise in evaluating if the system meets their needs and expectations.

A, C, and D are incorrect answers. Component testing, maintenance testing, and system testing are

not test types that call for involvement of an expert from the financial department for a new system that deals with all financial transactions in a company. Component testing is testing of individual software components in isolation from other components, usually done by developers. Maintenance testing is testing of a modified system or component after changes have been made to it, usually done by testers. System testing is testing of an integrated system as a whole to verify that it meets specified requirements, usually done by testers.

NO.32 What is Test Design?

- A. The overall process of testing a system to verify that it meets specified requirements.
- B. The process of selecting test techniques to exercise the functions implemented by the tested system
- C. The process of transforming general testing objectives into tangible test conditions and test cases
- D. An approach to testing to reduce the level of product risks by focusing on high-risk areas in the product

Answer: C

Explanation:

Test design is the process of transforming general testing objectives into tangible test conditions and test cases that can be executed and verified against expected results. Test design involves identifying test techniques, coverage criteria, input data, expected outcomes and other test parameters.

Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus1, Chapter 3, Section 3.2.1, page 53.

NO.33 What does the term 'Pesticide paradox' refer to?

- A. The phenomena where a piece of code that has a lot of bugs is likely to have more hidden, yet unfound
- B. The decreasing efficiency of debugging when done in code that has many bugs
- C. Reduced effectiveness of test cases that are repeated and focused on the same scenarios
- D. The redundancy of testing the same objects in both black and white box techniques

Answer: C

Explanation:

The term 'Pesticide paradox' refers to the phenomenon where the effectiveness of test cases that are repeated and focused on the same scenarios decreases over time because they tend to find the same defects or no defects at all. This is because the system under test becomes more resistant or immune to the existing test cases, just like pests become more resistant or immune to pesticides over time. To overcome the pesticide paradox, test cases should be regularly reviewed and updated to cover new or changed requirements, scenarios, risks, or defects. Test cases should also be designed to cover different aspects and perspectives of the system under test, such as functionality, usability, performance, security, etc. You can find more information about the pesticide paradox in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Section 4.11.

NO.34 Which of the following factors will MOST affect the testing effort required to test a software product? [K1]

- A. The number of staff available to execute tests
- B. The level of detail in the test plan

C. The requirements for reliability and security in the product

D. The test estimation method used

Answer: C

Explanation:

The testing effort required to test a software product depends on various factors, such as the size and complexity of the product, the quality of the requirements and design documents, the testability of the product, the test strategy and scope, the test environment and tools, the skills and experience of the testers, and the quality expectations and standards of the stakeholders¹. Among these factors, the requirements for reliability and security in the product will most affect the testing effort required to test a software product¹. Reliability and security are quality attributes that measure how well a software product performs its intended functions under specified conditions and protects itself from unauthorized access or harm¹. Testing for reliability and security requires more rigorous and thorough testing techniques, such as reliability testing, security testing, penetration testing, stress testing, etc¹. These techniques may require more time, resources, tools, and skills to perform effectively¹. Therefore, the requirements for reliability and security in the product will most affect the testing effort required to test a software product.

NO.35 When should testers be involved in reviewing a UAT specification? [K1]

A. At the beginning of the project

B. As soon as requirements have been approved

C. As soon as the UAT specification has been drafted

D. At any time before UAT begins

Answer: C

Explanation:

Testers should be involved in reviewing a UAT specification as soon as the UAT specification has been drafted¹. UAT stands for user acceptance testing, which is a type of testing that verifies that the software product meets the acceptance criteria and expectations of the end users or customers¹. A UAT specification is a document that defines the scope, objectives, approach, and criteria for UAT¹. Testers should be involved in reviewing a UAT specification as soon as the UAT specification has been drafted, as this can help to ensure that the UAT specification is clear, complete, consistent, testable, and aligned with the user requirements¹. Testers can also provide feedback and suggestions to improve the UAT specification and avoid potential issues or conflicts during UAT execution¹. Therefore, testers should be involved in reviewing a UAT specification as soon as the UAT specification has been drafted.