[Feb-2024 ISTQB-CTFL Certification with Actual Questions from Test4Engine [Q12-Q33

[Feb-2024] ISTQB-CTFL Certification with Actual Questions from Test4Engine
Updated ISTQB-CTFL Dumps PDF - ISTQB-CTFL Real Valid Brain Dumps With 78 Questions!

QUESTION 12

Which of the following is a key characteristic of informal reviews?
* Kick-off meeting
* Low cost
* Individual preparation
* Metrics analysis

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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 9.

QUESTION 13

Which of the following is NOT an objective of testing?
* Finding defects
* Providing information for decision-making
* Gaining confidence about the level of quality of the software
* Analyzing and removing the cause of failures

Analyzing and removing the cause of failures is not an objective of testing, but rather a task of development or maintenance. A failure is an event or behavior that deviates from the expected or specified result of a system under test. A failure is caused by an error (also known as a mistake or a fault) in the software code, design, or specification. Analyzing and removing the cause of failures is a process of locating and fixing errors in the software code, design, or specification, which is also known as debugging or defect resolution. Analyzing and removing the cause of failures does not aim to find or report defects, but rather to correct or prevent them. The other options are objectives of testing. Finding defects is one of the main objectives of testing, as it helps to improve the quality and reliability of the software product. Providing information for decision-making is another objective of testing, as it helps to support decision making and risk management. Gaining confidence about the level of quality of the software is another objective of testing, as it helps to assure that the software product meets its requirements and customer or user needs and expectations. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 3.

QUESTION 14

ST is a Software Testing organization which utilizes a testing knowledge base. Access to ST knowledge base can be either full or limited. Access level is determined based on ST certification and testing experience as follows:

1. If ST certified, with less than 5 years testing experience – allow limited access

2. If ST certified, 5-10 years of testing experience – allow full access

3. If not ST certified with 5-10 years of testing experience – allow limited access.

What would be the results for:

A – ST certified. 12 years of testing experience

B – Not ST certified. 7 years of testing experience

C – Not ST certified. 3 years of testing experience
* A – unknown

B – limned access

C- unknown
* A – full access

B – limited access

C – unknown
* A – full access

B – limited access

C – limited access
* A – unknown

B – full access

C – unknown
The correct answer can be derived by applying the given rules to each case:

A is ST certified and has 12 years of testing experience, which is more than 10 years. Therefore, A does not match any of the rules and the result is unknown.

B is not ST certified and has 7 years of testing experience, which is between 5 and 10 years. Therefore, B matches rule 3 and the result is limited access.

C is not ST certified and has 3 years of testing experience, which is less than 5 years. Therefore, C does not match any of the rules and the result is unknown. Verified Reference: This question does not require any external references, as it is based on logical reasoning.

QUESTION 15

A program got 100% decision coverage in a test. Which of the following statements is then guaranteed to be true?
* Every executable statement Is covered.
* Every output equivalence class has been tested.
* Every input equivalence class has been tested.
* The “dead” code has not been covered.
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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 36.

QUESTION 16

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.
* I, III. Iv
* II, v
* I, Ill
* II. IV, V

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.

QUESTION 17

A mid-size software product development company has analyzed data related to defects detected in its product and found out that detects 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?
* Automate existing test suits so that lesser time is spent on execution of each test, and more tests can be executed m the available time thus leading to a lower probability of defects slipping by
* 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
* Analyze the product modules containing maximum defects, and get them thoroughly tested and defects fixed as a one-time activity
* If required, tram 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

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

QUESTION 18

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?
* V-Model
* Agile
* Waterfall
* Prototyping

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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 18.

QUESTION 19

Which of the following is NOT an experience-based technique?
* Boundary value analysis.
* Error guessing
* Exploratory testing
* Fault attack

Boundary value analysis is not an experience-based technique, but rather a specification-based technique (also known as black-box technique). Experience-based techniques are 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. Some examples of experience-based techniques are error guessing, exploratory testing, fault attack, checklist-based testing, etc. Specification-based techniques are techniques that rely on the tester’s analysis and interpretation of the requirements or specifications of the system under test to derive and select test cases based on some criteria or rules. Some examples of specification-based techniques are equivalence partitioning, boundary value analysis, decision table testing, state transition testing, etc. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 31.

QUESTION 20

Which of the following is a function of a dynamic analysis tool?
* Provide support for traceability of tests, test results and incidents to source documents
* Monitor the allocation, use and de-allocation of memory during run-time of a program
* Execute programs step-by-step in order to reproduce failures and find corresponding defects
* Provide support for release of baselines consisting of configuration items

A dynamic analysis tool is a tool that performs analysis of a software product based on its behavior during execution. A dynamic analysis tool can monitor various aspects of a program’s run-time performance, such as memory usage, CPU load, response time, or resource leaks. A dynamic analysis tool can monitor the allocation, use and de-allocation of memory during run-time of a program, which can help detect defects such as memory leaks, buffer overflows, or memory corruption. A dynamic analysis tool cannot provide support for traceability of tests, test results and incidents to source documents, as this is a function of a test management tool. A dynamic analysis tool cannot execute programs step-by-step in order to reproduce failures and find corresponding defects, as this is a function of a debugging tool. A dynamic analysis tool cannot provide support for release of baselines consisting of configuration items, as this is a function of a configuration management tool. Verified Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 6, page 56-57.

QUESTION 21

Which of the following statements about reviews are TRUE?

I In walkthroughs the review meeting is typically led by the author.

II Inspection is characterized by an open-ended review meeting

III Preparation before the review meeting is part of informal reviews

IV Management rarely participates in technical review meetings
* II, Ill
* I, II
* I, IV
* III, IV

The following statements about reviews are true:

I) In walkthroughs the review meeting is typically led by the author. A walkthrough is a type of review that has a predefined objective and agenda but no formal process or roles. A walkthrough is typically led by the author of the work product under review, who guides the participants through a scenario and solicits feedback.

IV) Management rarely participates in technical review meetings. A technical review is a type of review that has a predefined objective and agenda but no formal process or roles. A technical review is typically performed by peers with technical expertise in order to evaluate technical aspects of a work product. Management rarely participates in technical review meetings, as they may not

have sufficient technical knowledge or skills to contribute effectively. The following statements about reviews are false:

II) Inspection is characterized by an open-ended review meeting. An inspection is a type of review that follows a defined process with formal entry and exit criteria and roles and responsibilities for participants. An inspection is characterized by a structured review meeting with a fixed duration and agenda.

III) Preparation before the review meeting is part of informal reviews. Preparation before the review meeting is part of formal reviews, such as inspections or technical reviews. Preparation involves checking

QUESTION 22

In which of the following cases you would NOT execute maintenance testing?
* Retirement of the software or system
* Modifications to a released software or system
* Migration of the system data to a replacement system
* Update to the Maintainability requirements during the development phase

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 Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 2, page 18-19.

QUESTION 23

Which of the following can be considered a VALID exit criterion?

I Estimates of defect density or reliability measures.

II. The completion and publication of an exhaustive Test Report.

III. Accuracy measures, such as code, functionality or risk coverage.

IV Residual risks such as lack of code coverage in certain areas.
* I, III, IV
* I, II, III
* III, IV
* II, III, IV

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

I) Estimates of defect density or reliability measures. These are quantitative measures that indicate how many defects are present in the software product or how likely it is to fail under certain conditions. These can be used as exit criteria to ensure that the software product meets a certain level of quality or performance before moving to the next phase or releasing it to the customer.

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

IV) Residual risks, such as lack of code coverage in certain areas, unresolved defects or unknown factors. These are qualitative

measures that indicate the remaining risks or uncertainties associated with the software product after testing. These can be used as exit criteria to ensure that the residual risks are acceptable or manageable before moving to the next phase or releasing it to the customer. The following factor is not a valid exit criterion:

II) The completion and publication of an exhaustive Test Report. This is not a valid exit criterion, as it does not reflect the quality or completeness of the testing process or product. A test report is a document that summarizes the results and outcomes of a test activity or phase. A test report can be used as an input for deciding whether to exit a test activity or phase, but it is not a condition that defines when to exit. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, Chapter 2, page 13; Chapter 6, page 58-59.

QUESTION 24

Which of the following statements is LEAST likely to be describing component testing?
* It identifies defects in modules and classes.
* Simulators and stubs may be required.
* It mainly tests interfaces and interaction between components.
* It may be applied using a test-first approach.

Component testing (also known as unit testing or module testing) is a level of testing that focuses on verifying the functionality and quality of individual software components (such as modules, classes, functions, methods, etc.). Component testing mainly tests interfaces and interaction between components, as well as internal logic and data structures of the components. Component testing may be applied using a test-first approach (such as test-driven development or behavior-driven development), where tests are written before the code is implemented. Component testing does not identify defects in modules and classes, as this is a result of component testing, not an objective. Simulators and stubs may be required for component testing, as they can simulate or replace missing or incomplete components or external systems that are needed for testing. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 19.

QUESTION 25

In what way do Configuration Management effects testing?
* Without proper configuration management, test planning cannot proceed.
* Proper configuration management ensures that testers can uniquely identity the tested item
* Configuration management is important for developers, not for testers
* There is very little influence of configuration management practices on the test project.

Configuration management is a process that establishes and maintains consistency among work products throughout their life cycle. Configuration management affects testing in various ways, such as:

Proper configuration management ensures that testers can uniquely identify the tested item, which can help traceability, reproducibility and accountability.

Proper configuration management ensures that testers have access to consistent versions of software components and testware, which can help reliability, compatibility and efficiency.

Proper configuration management ensures that testers can track changes and defects in software components and testware, which can help verification, validation and reporting.

Proper configuration management ensures that testers can control the configuration of the test environment, which can help stability, security and performance. Configuration management is not a prerequisite for test planning, as test planning can proceed without configuration management, although it may be less effective or accurate. Configuration management is not important for developers only, but for testers as well, as it affects the quality and consistency of the testing process and products. Configuration management has a significant influence on the test project, as it affects various aspects of testing, such as traceability, reproducibility, reliability,

compatibility, efficiency, verification, validation, reporting, stability, security and performance. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, Chapter 6, page 60-61.

QUESTION 26

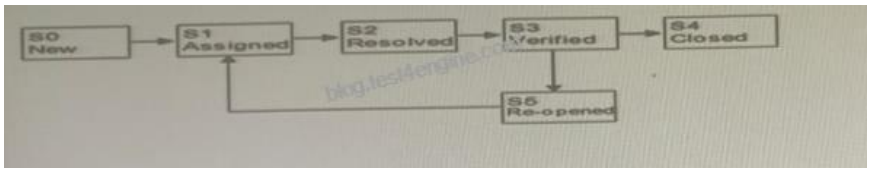
The following sentences refer to the ‘Standard for Software Test Documentation’ specification (IEEE 829).

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

The IEEE 829 standard is a widely used specification for test documentation, but it is not mandatory or universal. Most test documentation regimes follow this spec to some degree, with changes done to fit a specific situation or organization. The standard does not require any approval from management, marketing or development for any deviation, nor does it depend on the type of project (military or consumer market). The standard also does not guarantee high quality test documentation regimes, as it only provides a general outline and format, not the actual content or quality criteria. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 16.

QUESTION 27

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



* S0->S1->S2->S3->S4
* S0->S1->S2->S3->S5>S1
* S0->S1->S2->S3->S5->S1->S2->S3
* S0->S1->S2->S3->S5->S3->S4

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 Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 4, page 40-41.

QUESTION 28

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
* I III IV and V are true. II is false.
* II, III IV and V are correct. I is false.
* I, II and IV are true. III and V are false.
* II, Ill and V are true. 1 ants Iv are false.

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.

QUESTION 29

Software was found lo take much more time than the stated requirement of less than one second to save a file. Upon investigation it was found that there was an unnecessary check inside a loop which was slowing down the file-save operation. The software not being able to meet the desired response time is an example of

* It is not a defect
* Defect
* Error
* Failure

A failure is an event in which a component or system does not perform a required function within specified limits. A failure is observable by the software users or other stakeholders. A failure is caused by one or more defects in the software. In this case, the software not being able to meet the desired response time is an example of a failure, as it deviates from the stated requirement and affects the user experience. It is not a defect, which is a flaw in the software that causes the failure. It is not an error, which is a human action that produces an incorrect result. It is not a non-defect, as it clearly violates a specified requirement. Verified Reference: [A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer], Chapter 1, page 4.

QUESTION 30

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:

* Identification of unreachable code
* Report on adherence to the coding standards
* Number of comment lines
* Number of source code lines

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 Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, page 8.

QUESTION 31

Which of the following statements about testware are correct?

I When closing the test activities, all the testware resources can be uninstalled and released II All the testware should be subject to Configuration Management III. The testware. at the end of the project, should be transferred to the organization responsible for maintenance IV The developers are responsible for the correct installation of the testware

* II, Ill
* I, Ill
* I, IV
* II, IV

Testware is a term that refers to all artifacts produced during the testing process, such as test plans, test cases, test scripts, test data, test results, defect reports, etc. The following statements about testware are correct:

II) All the testware should be subject to Configuration Management. Configuration management is a process that establishes and maintains consistency among work products throughout their life cycle. Configuration management applies to all testware, as it

helps ensure their quality and consistency, track their changes and defects, control their versions and access rights, and link them to other artifacts.

III) The testware at the end of the project should be transferred to the organization responsible for maintenance. Maintenance testing is testing performed on a software product after delivery to correct defects or improve performance or other attributes. Maintenance testing requires testware from previous testing activities or phases, such as test cases, test data, test results, etc. Therefore, the testware at the end of the project should be transferred to the organization responsible for maintenance testing, such as support team or maintenance team. The following statements about testware are incorrect:

I) When closing the test activities, all the testware resources can be uninstalled and released. This statement is incorrect, as some testware resources may still be needed for future testing activities or phases, such as maintenance testing or regression testing. Therefore, when closing the test activities, some testware resources should be archived and stored for future use, while others can be uninstalled and released.

IV) The developers are responsible for the correct installation of the testware. This statement is incorrect, as the testers are responsible for the correct installation of the testware. The testers should ensure that they have access to all necessary testware resources and that they are installed and configured properly before starting the test execution. Verified Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus – Springer, Chapter 6, page 58-61.

QUESTION 32

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

I Stubs and drivers are often used

II The lest 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
* Component – VI

Integration – IV

System -1

Acceptance – 111
* Component – IV

Integration -1

System – VI

Acceptance – V
* Component-I

Integration – V

System – II

Acceptance – IV
* Component – V

Integration – II

System – IV

Acceptance – VI
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.

QUESTION 33

Which of the following statements about Experience Based Techniques (EBT) is correct?
* EBT use tests derived from the test engineers’ previous experience with similar technologies.
* EBT is based on the ability of the test engineer to implement various testing techniques.
* EBT is done as a second stage of testing, after non-experienced-based testing took place.
* EBT require broad and deep knowledge in testing but not necessarily in the application or technological domain.

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