Free Questions for ISTQB-CTFL by dumpshq

Shared by Daniels on 12-12-2023

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

In maintenance testing, what is the relationship between impact analysis and regression testing?

Options:

A- Impact analysis requires a regression testing for only the tests that have detected faults in previous SW release

B- There is no relationship between impact analysis and regression testing.

C- Impact analysis requires a regression testing for all program elements which were newly integrated (new functionalities).

D- The impact analysis is used to evaluate the amount of regression testing to be performed.

Answer:

D

Explanation:

In maintenance testing, the relationship between impact analysis and regression testing is that the impact analysis is used to evaluate the amount of regression testing to be performed. Maintenance testing is a type of testing that is performed on an existing software product after it has been delivered or deployed, in order to ensure that it still meets its requirements and functions correctly after a

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

Question 2

Question Type: **MultipleChoice**

Which of the following should be included in a test status report?

I Estimation details

II Total number of open and closed defects

III Actual effort spent

IV Defect reports

V Number of executed, failed, blocked tests

Options:

A- III.V

B- II, III

C- I. II. IV

D- II, III.V

Answer:

D

Explanation:

The following should be included in a test status report: total number of open and closed defects, actual effort spent, and number of executed, failed, and blocked tests. A test status report is a document that provides information on the results and status of testing activities for a given period or phase. A test status report should include information that is relevant, accurate, and timely for the intended audience and purpose. Some of the information that should be included in a test status report are: total number of open and closed defects, which can indicate the defect trend and defect density of the software product; actual effort spent, which can indicate the productivity and efficiency of the testing process; number of executed, failed, and blocked tests, which can indicate the test progress and

test coverage of the software product. The following should not be included in a test status report: estimation details, defect reports, and impact analysis. Estimation details are not part of a test status report, but rather part of a test plan or a test estimation document. Estimation details provide information on the expected time, resources, and costs for testing activities, not on the actual results or status of testing activities. Defect reports are not part of a test status report, but rather separate documents that provide detailed information on individual defects found during testing. Defect reports include information such as defect description, defect severity, defect priority, defect status, defect resolution, etc. Defect reports can be referenced or summarized in a test status report, but not included in full. Impact analysis is not part of a test status report, but rather part of a risk assessment or prioritization process. Impact analysis provides information on the potential effects or consequences of a change or a defect on the software product or project. Impact analysis can be used to evaluate the amount or scope of testing to be performed, but not to report the results or status of testing activities. Verified Reference:A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 141.

Question 3

Question Type: MultipleChoice

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

Options:

A- Every executable statement Is covered.

B- Every output equivalence class has been tested.

C- Every input equivalence class has been tested.

D- The 'dead' code has not been covered.

Answer:

A

Explanation:

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 4

Question Type: MultipleChoice

How can testing contribute to higher quality?

Options:

A- Testing help to measure the quality of software.

B- Testing ensures that remaining defects are documented.

C- Testing removes errors in the software.

D- Testing eliminates the risk with software.

Answer:

A

Explanation:

Testing can contribute to higher quality by helping to measure the quality of software. Quality is defined as the degree to which a component or system satisfies specified requirements and customer or user needs and expectations. Testing is a process of evaluating a component or system by applying inputs and observing outputs, and comparing them with expected results. Testing can help to measure the quality of software by providing information on its functionality, performance, usability, security, reliability, etc. Testing can also help to identify and report defects in software, which can lead to improvement actions and quality assurance activities. The other options are not accurate descriptions of how testing can contribute to higher quality. Testing does not ensure that remaining defects are documented, but rather that detected defects are reported. Testing does not remove errors in software, but rather finds defects in software behavior or quality. Testing does not eliminate the risk with software, but rather assesses and manages the risk with software. Verified Reference:A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 3.

Question 5

Question Type: MultipleChoice

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

Options:

A- Test schedules

B- Cost of testing performed so far

C- Confidence of testers in tested code

D- Number of unfixed defects

Answer:

A

Explanation:

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

Question 6

Question Type: MultipleChoice

Why should you choose a test technique?

Options:

A- Because you need to match the way you test to the content of the product under test

B- Because of the time constraints that usually accompany a test project

C- Because this way you cover the full scope of the product's functionality

D- Because choosing a test technique is a common practice in software testing

Answer:

A

Explanation:

You should choose a test technique because you need to match the way you test to the content of the product under test. A test technique is a method or process for deriving and selecting test cases based on some criteria or rules. Different test techniques are suitable for different types of software products, depending on their characteristics, functionalities, requirements, specifications, risks, etc. Choosing a test technique helps to ensure that the test cases are relevant, effective, and efficient for the product under test. The other options are not correct reasons to choose a test technique. Time constraints are not a factor for choosing a test technique, but rather for prioritizing or optimizing testing activities. Covering the full scope of the product's functionality is not a guarantee of choosing a test technique, but rather a goal of testing. Choosing a test technique is not a common practice in software testing, but rather a professional skill and responsibility. Verified Reference:A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 31.

Question 7

Question Type: MultipleChoice

Which of the following is NOT a product risk?

Options:

A- Poor software usability

B- Failure-prone software is delivered

C- Problems in defining the right requirements

D- Software does not perform the intended functions

Answer:

C

Explanation:

Problems in defining the right requirements is not a product risk, but rather a project risk. A product risk is a risk that affects the quality or performance of the software product itself, such as poor usability, failure-prone functionality, security vulnerabilities, compatibility issues, etc. A project risk is a risk that affects the management or delivery of the software project itself, such as unrealistic schedule, insufficient resources, unclear scope, changing requirements, etc. The other options are examples of product risks, as they relate to the software product's characteristics or features. Verified Reference:A Study Guide to the ISTQB Foundation Level 2018 Syllabus - Springer, page 12.