DUMPSBOSS.

ISTQB Certified Tester Advanced Level - Technical Test Analyst (V4.0)

iSQI CTAL-TTA_Syll19_4.0

Version Demo

Total Demo Questions: 5

Total Premium Questions: 53

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, The Technical Test Analyst’s Tasks in Risk-Based Testing	4
Topic 2, White-Box Test Techniques	8
Topic 3, Static and Dynamic Analysis	11
Topic 4, Quality Characteristics for Technical Testing	19
Topic 5, Reviews	4
Topic 6, Tools and Automation	7
Total	53

QUESTION NO: 1

Consider the pseudo code provided below regarding a customer request for cash withdrawal from an ATM.

If the customer has sufficient funds in their account

OR the customer has the credit granted

THEN the ATM machine pays out the requested amount to the customer

Which of the following test cases would be the result of applying multiple condition testing, but would NOT be the result of applying modified condition/decision testing?

A. TC 1: Customer has sufficient funds. Credit has not been granted.

B. TC 2: Customer does not have sufficient funds. Credit has been granted.

C. TC 3: Customer does not have sufficient funds. Credit has not been granted.

D. TC 4: Customer has sufficient funds. Credit has been granted.

ANSWER: D

Explanation:

TC 4: Customer has sufficient funds. Credit has been granted. is correct. The decision has two atomic conditions joined by OR: sufficient funds and credit granted. Multiple condition testing requires coverage of every feasible combination of the atomic-condition truth values. Therefore, it includes the case in which both conditions are true, and the ATM pays the requested amount.

Modified condition/decision testing (MC/DC) instead requires evidence that each individual condition can independently affect the decision outcome while the other condition is held at a value that permits that effect to be observed. For an OR decision, sufficient funds is demonstrated as independently affecting the result by comparing a case with neither condition true against a case with sufficient funds true and credit not granted. Credit granted is demonstrated by comparing the neither-true case against a case with credit granted and insufficient funds. These comparisons meet MC/DC without requiring the both-true combination. Consequently, the both-true test is required by multiple condition testing but is not needed in the MC/DC test set. This distinction follows the structural-test coverage concepts used in the ISTQB Advanced Level Technical Test Analyst syllabus; see ISTQB Technical Test Analyst certification and the ISTQB glossary definition of modified condition/decision testing.

QUESTION NO: 2

A Technical Test Analyst is defining static-analysis entry criteria for components that will be changed frequently after release. The objective is to identify code properties that could make future changes difficult to understand, modify, or test.

Which TWO measures are MOST useful for this objective? (Choose two.)

A. Cyclomatic complexity of the component

B. Mean Time Between Failures of the system

C. Degree of coupling between the component and other modules

D. Number of records processed per hour

E. Number of concurrent users in production

ANSWER: A C

Explanation:

Cyclomatic complexity is useful because a highly complex control-flow structure tends to require more effort to understand, modify, and test. High coupling between modules is also a significant maintainability concern because a local change may require changes, retesting, or coordination across several dependent modules.

Mean Time Between Failures is an operational reliability measure and cannot be established from static source-code analysis. The number of records processed per hour is a performance measure, while the number of concurrent users is a workload characteristic. Neither directly indicates the maintainability of a source-code component.

QUESTION NO: 3

The following characteristics were identified during an early product risk-assessment for a software system:

• the software system needs to manage synchronization between various processes

• microcontrollers will be used that will limit product performance

• the hardware that will be used will make use of timeslots

• the number of tasks supported in parallel by the software system is large and are often highly complex.

Based on the information provided, which of the following non-functional test types is MOST appropriate to be performed?

A. Maintainability testing

B. Security testing

C. Time-behaviour testing

D. Portability testing

ANSWER: C

Explanation:

Time-behaviour testing is the most appropriate test type because the identified risks are fundamentally about whether the system can perform required work within defined time constraints. Coordinating synchronization among multiple processes, operating on resource-constrained microcontrollers, using hardware time slots, and supporting numerous complex tasks concurrently all create a significant risk that processing will be delayed, scheduled incorrectly, or fail to meet deadlines. These are especially important concerns in embedded and real-time systems, where correctness depends not only on producing the right result but also on producing it at the required time.

Time-behaviour testing evaluates time-related quality characteristics such as response time, processing time, throughput, capacity under concurrent demand, and compliance with timing deadlines. Tests should therefore measure the timing of critical operations and process synchronization while representative numbers of parallel tasks execute under realistic hardware constraints. This gives stakeholders objective evidence that the system's scheduling and resource use meet its specified timing needs. The ISO/IEC 25010 product-quality model identifies time behaviour as a performance-efficiency subcharacteristic, focused on whether response and processing times and throughput rates meet requirements. See ISO 25010 overview and the ISTQB Technical Test Analyst certification page.

QUESTION NO: 4

A network-management component runs with elevated operating-system privileges and processes commands received from external devices. Static analysis is being introduced before component integration.

Which TWO findings should be given the highest security-related priority? (Choose two.)

A. A received command length is used to copy data into a fixed-size buffer without a bounds check.

B. A source file contains several methods that exceed the team's preferred length guideline.

C. An administrative password is stored as a literal value in the source code.

D. Comments use inconsistent punctuation and capitalization.

E. A temporary variable is declared within the smallest applicable block.

ANSWER: A C

Explanation:

A write to a fixed-size buffer without validating the received length can allow memory corruption when externally supplied data is malformed or deliberately hostile. Use of a hard-coded administrative credential can allow unauthorized access if the code or deployed artefact is obtained. Both findings directly expose significant security risks in a privileged component.

Long source files and inconsistent comment formatting can affect maintainability, but they do not directly create the described security exposure. A variable with a limited scope is normally preferable to a variable with unnecessarily broad scope.

QUESTION NO: 5

Within an embedded software project, the maintainability of the software is considered to be critical. It has been decided to use static analysis on each delivered software component.

Which of the following metrics is NOT a maintainability metric typically used with static analysis?

A. Number of Lines of Code (LOG)

B. Number of Function Calls

C. Mean Time Between Failures

D. Comment Frequency

ANSWER: C

Explanation:

Mean Time Between Failures is the correct answer because it measures an operational reliability characteristic: the average elapsed time between failures of a repairable system. Establishing this value requires failure data gathered while the system is executing in its intended environment, or from representative reliability testing. It therefore concerns the dependability experienced in operation rather than structural properties of source code that can be measured without executing it.

Static analysis supports assessment of maintainability by examining code and its structure, including properties that affect how readily software can be understood, modified, reviewed, and tested. In an embedded project, applying static analysis to each delivered component provides early, repeatable evidence about such internal code attributes before integration or operational use. This is particularly useful where changes may be costly and where component-level quality controls are required.

The ISTQB Technical Test Analyst certification explicitly covers static analysis and the technical quality characteristics assessed through it; see the ISTQB Technical Test Analyst certification page. The ISO/IEC 25010 quality model also distinguishes maintainability from reliability as separate product-quality characteristics: ISO 25010 overview.