DUMPSBOSS.

Certified Tester Test Automation Engineer

ISTQB CT-TAE

Version Demo

Total Demo Questions: 10

Total Premium Questions: 107

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Introduction and Objectives for Test Automation	3
Topic 2, Preparing for Test Automation	29
Topic 3, The Generic Test Automation Architecture	28
Topic 4, Deployment Risks and Contingencies	19
Topic 5, Test Automation Reporting and Metrics	14
Topic 6, Transitioning Manual Testing to an Automated Environment	7
Topic 7, Verifying the Test Automation Solution	7
Total	107

QUESTION NO: 1

Consider a TAS deployed into production. The SUT is a web application and the test suite consists of a set of automated regression tests developed via GUI. A keyword-driven framework has been adopted for automating the regression tests. The tests are based on identification at low-levels of the web page components (e.g class indexes, tab sequence indexes and coordinates) in the next planned release the SUT will be subject to significant corrective maintenance (bug-fixes) and evolution (new features) Maintenance costs to update the test scripts should be as low as possible and the scripts must be highly reusable.

Which of the following statements is most likely to be TRUE?

A. The keyword-driven framework is not suitable, it would be better to adopt a structured-scripting approach

B. False positive errors are likely to occur when running the automated tests on the new releases without modifying the test

C. The total execution time of the automated regression test suite will decrease for each planned release.

D. The keyword-driven framework introduces a level abstraction that is too high and makes it difficult what really happens

ANSWER: B

Explanation:

“False positive errors are likely to occur when running the automated tests on the new releases without modifying the test” is correct. Although a keyword-driven framework can support reuse and separation of test intent from implementation, the effective maintainability of GUI tests also depends heavily on the robustness of the object-identification strategy. Here, the tests identify web components through volatile, presentation-level properties such as class indexes, tab sequence indexes, and screen coordinates. Significant corrective maintenance and functional evolution can alter page layout, element order, styling, or the generated DOM structure without necessarily breaking the business behaviour being tested.

Consequently, existing automated tests may fail because they can no longer find or interact with the expected component, even when the corresponding requirement is still correctly implemented. Such a failure reports an apparent test problem or product defect that does not represent a real functional regression: a false positive. Reducing this risk requires maintainable, stable locators and appropriate abstraction of UI interactions, for example by using unique, intentional identifiers where available and centralising locator maintenance. This is consistent with the CT-TAE focus on maintainability, reusability, and controlling false alarms in test automation. See the ISTQB Certified Tester Test Automation Engineer certification overview and Selenium’s guidance on web-element locators.

QUESTION NO: 2

You are using a gTAA to create a TAS for a project. The TAS is aimed at automatically and executing test cases based on a use-case

Modeling approaching that uses UML as a modeling language. All the interaction between TAS and SUT will only be at the API and GUI level.

Which of the following components of the gTAA would you EXCLUDE from the TAS?

A. The test reporting component of the test execution layer.

B. The Test execution component of the test generation layer

C. The test execution (test engine of the test execution layer

D. The Command Line Interface (CLI) component of the test adaptation layer

ANSWER: D

Explanation:

The Command Line Interface (CLI) component of the test adaptation layer is the component to exclude because the stated test automation solution interacts with the system under test exclusively through APIs and graphical user interfaces. In the generic Test Automation Architecture (gTAA), the test adaptation layer contains the technology-specific means by which the

Test Automation System connects to and controls the SUT. A CLI adapter is appropriate only when command-line interaction is a required SUT interface or is otherwise needed to set up, control, observe, or tear down tests. Since neither such interaction nor a command-line interface is within the project scope, including this adapter would add unnecessary implementation, maintenance, and configuration work without supporting a required test interaction. The TAS should instead provide adaptations that support the declared API and GUI interfaces, allowing generated tests based on the UML use-case model to be executed through those channels. This reflects the gTAA principle that a concrete TAS is tailored from the generic architecture according to the project's actual test objectives, SUT interfaces, and technical constraints. The ISTQB Test Automation Engineer certification overview and its supporting syllabus resources describe gTAA as a basis for selecting and implementing only the architecture components needed for a specific automation solution. See ISTQB Certified Tester Test Automation Engineer and ISTQB CT-TAE certification resources.

QUESTION NO: 3

Designing the System Under Test (SUT) for testability is important for a good test automation approach and can also benefit manual test execution.

Which of the following is NOT a consideration when designing for testability?

A. Observability: The SUT needs to provide interface that give insight into the system.

B. Re-useability: The code written for the SUT must be re-useable for other similar system.

C. Clearly defined architecture: The SUT Architecture needs to provide clear and understandable interfaces giving control and visibility on all test levels.

D. Control: the SUT needs to provide interfaces that can be used to perform actions on SUT.

ANSWER: B

Explanation:

"Re-useability: The code written for the SUT must be re-useable for other similar system." is the correct answer because code reuse across similar systems is a general software-development and maintainability objective, rather than a core SUT testability design consideration. Testability concerns how effectively testers and automated tests can set up conditions, operate the SUT, observe its behavior, and determine results. Designing a testable SUT therefore focuses on providing suitable interfaces and mechanisms through which tests can control the system and obtain meaningful, reliable visibility of its internal state, outputs, and interactions. A clearly structured architecture with understandable interfaces also supports these goals by making test access and observation practical at relevant test levels. Reusable production code may be valuable, but it does not inherently make a particular SUT easier to control, observe, isolate, or evaluate through automated testing. The ISTQB Test Automation Engineering syllabus treats SUT testability as an important technical factor in a sustainable automation solution and discusses the need for appropriate controllability and observability. See the ISTQB Certified Tester Test Automation Engineering certification page and the ISTQB glossary entry for testability.

QUESTION NO: 4

Your project is transitioning from manual to automated testing. You have decided to implement a pilot project so that lessons learned can inform future time estimates and schedules.

Which two of the following represent the types of test cases that are MOST suited to a test automation pilot project?

a) High added value test cases that require little effort to automate.

b) Test that are run infrequently as these will be simpler to automate

c) Reliability test cases that can show added value soon

d) Technically challenging test cases to provide the best validation of manual test conversion

e) Tests that are least Important to the business as these are safer to trial

A. a and b

B. a and c

C. b and d

D. c and e

ANSWER: B

Explanation:

“a and c” is correct because a test automation pilot should demonstrate useful benefits early while keeping implementation risk and effort manageable. High added value test cases that require little effort to automate are a strong pilot choice: they provide a favorable return on the initial automation investment and allow the team to establish the test automation architecture, coding conventions, execution process, reporting, and maintenance approach without being delayed by unnecessary complexity. The pilot can therefore produce realistic information for estimating subsequent automation work.

Reliability test cases that can show added value soon are also suitable because automation is especially valuable when tests must be executed consistently and repeatedly, or when they support confidence in the product’s stable behavior. Showing a measurable benefit early helps obtain stakeholder support and validates that the selected tools and approach fit the project. A pilot is not merely a technical experiment; it should provide evidence that the automation solution can deliver meaningful, repeatable results in the intended delivery context. This aligns with the CT-TAE emphasis on selecting and implementing an automation solution that provides maintainable, effective feedback and demonstrable value. See the ISTQB Certified Tester Test Automation Engineer certification page and the ISTQB glossary entry for test automation.

QUESTION NO: 5

Which of the following statements about the reuse of TAS artefacts is TRUE?

A. Reusable TAS artefacts can include components (or parts of components) associated with different layers of the TAA

B. To enable reuse of TAS artefacts, a good design for reuse is built into the TAA and to further action are needed during the TAS lifecycle

C. Communications maintenance and improvements for reusing TAS artefacts are modify addressed during the design of the TAA

D. Reusable TAS artifacts associated with the definition layer of the TAA include the adaptors to the SUT components and/or interfaces

ANSWER: A

Explanation:

Reusable TAS artefacts can include components (or parts of components) associated with different layers of the TAA. This is correct because reuse in test automation is not limited to a single type of asset or a single architectural layer. A Test Automation Architecture (TAA) structures the Test Automation Solution (TAS) into layers with distinct responsibilities, and artefacts from those layers can be designed and selected for use beyond their original implementation context. For example, reusable assets may range from test-definition-related structures to execution-supporting utilities, adapters, libraries, or shared mechanisms, provided their dependencies and interfaces are appropriately managed.

Considering reuse across layers is important when designing a maintainable and extensible TAS. It enables teams to retain valuable automation capabilities when tests, products, environments, or projects evolve, rather than rebuilding every supporting component. The CT-TAE syllabus treats reuse as a design consideration for TAS artefacts and recognizes that reusable components or portions of components may be associated with different TAA layers. This supports consistent automation practices and helps reduce duplication across the automated testing ecosystem. See the official ISTQB Certified Tester Test Automation Engineer certification page and the ISTQB certifications overview for the current syllabus materials and certification context.

QUESTION NO: 6

Consider a TAS that exclusively uses the APIs of a SUT. To make this work, significant changes have been required to the SUT by adding a set of dedicated test interfaces to the APIs. All the automated tests will use these test interfaces when interacting with the SUT. Assume that you are currently verifying the correctness of the automated test environment and test tool setup.

Which of the following would you expect to be the MOST specific risk associated with this scenario?

A. The connectivity from the TAS to the dedicated test interfaces will not work

B. The process of configuring the TAS will be error-phone due to manual intervention

C. The automated test cases will not contain the expected result

D. False alarms, that are unlikely to occur in the real world, will be observed during testing

ANSWER: D

Explanation:

False alarms, that are unlikely to occur in the real world, will be observed during testing is the most specific risk because the automated tests interact with dedicated test interfaces rather than solely through the interfaces and conditions used by real users or production integrations. Adding these interfaces requires changes to the SUT and introduces behavior, states, error handling, timing, configuration, or data-access paths that may exist only for testing. Consequently, an automated test can report a failure caused by the dedicated interface or its interaction with the TAS, even though the corresponding production behavior is correct. Such a result is a false alarm: it consumes investigation effort, reduces confidence in automated-test results, and can obscure meaningful failure information. During verification of the test automation environment and tools, it is therefore important to establish that failures reported through these test-specific interfaces can be reliably distinguished from actual SUT defects and that the interfaces faithfully support the intended test purpose. This aligns with the CT-TAE syllabus emphasis on identifying technical and deployment risks related to the TAS, SUT interfaces, and reliability of automated test execution. See the official ISTQB Certified Tester Test Automation Engineer certification page and the ISTQB CT-TAE syllabus resources.

QUESTION NO: 7

You have been asked to automate a set of functional tests at system Test level via the CLI of the SUT for the first release of a software system. The automated tests will be delivered to the learn in change of maintenance testing, who will use them for part of the regression testing. They have the following requirements.

1. The automated tests must be as fast and cheap to maintain as possible

2. The cost of adding new automated tests must be as low as possible

3. The automated tests must have a high level of independence from the tool itself

Which of the following scripting techniques would be MOST suitable?

A. Data-driven scripting

B. Keyword-driven scripting

C. Linear scripting

D. Structure scripting

ANSWER: B

Explanation:

Keyword-driven scripting is the most suitable technique because it separates the business-level description of a test from the lower-level automation implementation. Tests are assembled from keywords that express meaningful actions, while the keyword implementations contain the tool-specific commands needed to interact with the system through its CLI. This separation means that a change to the automation tool, its libraries, or the CLI interaction details can usually be addressed centrally in the keyword implementation layer rather than by editing every regression test.

For maintenance testing, this structure supports fast and economical upkeep: common actions are implemented once and reused across many tests. It also reduces the effort needed to add coverage, because new automated tests can be created primarily by combining existing keywords and supplying relevant test data, instead of repeatedly writing detailed automation code. The resulting test specifications are more readable and remain focused on test intent, which makes them practical for a maintenance team to understand and extend. This aligns with the CT-TAE emphasis on selecting automation approaches that provide maintainability, reusability, abstraction, and independence from a particular test automation technology. See the ISTQB CT-TAE certification overview and the ISTQB certifications and syllabus resources.

QUESTION NO: 8

As a TAE you are evaluating a functional test automation tool that will be for several projects within your organization. The projects require that tool to work effectively and efficiently with SUT's in distributed environments. The test automated tool also needs to interface with other existing test tools (test management tool and defect tracking tool.) The existing test tools subject to planned updates and their interface to the test automated tool may not work property after these updates.

Which of the following are the two LEAST important concerns related to the evaluation of the test automation in this scenario?

A) Is the test automation tool able to launch processors and execute test cases on multiple machines in different environments?

B) Does the test automation tool support a licensing scheme that allows accessing different sets?

C) Does the test automation tool have a large feature set, but only part of the features will be sets?

D) Do the release notes for the planned updates on existing specify the impacts on their interfaces to other tools?

Does the test automation tool need to install specific libraries that could impact the SUT?

A. A and C

B. A and E

C. B and E

D. C and D

E. B and C

ANSWER: E

Explanation:

B and C are the two least important concerns in relation to the stated evaluation priorities. The scenario emphasizes technical suitability for distributed execution, interoperability with the organization's existing test-management and defect-tracking tools, and resilience to planned changes in those integrated tools. Evaluation should therefore concentrate on whether the tool can support the required execution architecture and whether its interfaces can be maintained reliably as the surrounding toolchain evolves. It must also consider whether the tool's installation or instrumentation affects the system under test, because such effects can compromise test validity or operational suitability.

By comparison, whether a licensing arrangement permits access to different sets is not described as a decisive operational or integration requirement. Likewise, a broad feature set of which only a subset will be used is not, by itself, a key selection concern; the relevant issue is whether the required subset supports the defined test-automation objectives. ISTQB's test automation guidance treats tool selection as a context-dependent assessment focused on fit with technical environment, interfaces, execution needs, and organizational requirements. See the ISTQB Certified Tester Test Automation Engineer certification page and the ISTQB Glossary for ISTQB terminology and certification resources.

QUESTION NO: 9

You have been asked to implement test automation for a project that is not meeting its deadlines. After further analysis you discover that the manual testers are not able to keep up with the new feature testing because the regression testing is taking

75% of their time. As a result, the new features are being released with many defects and customers are complaining about the quality.

Given this information, what metric SHOULD you be tracking to show the value of test automation for this project?

A. Percentage of code covered by the test automation.

B. Equivalent Manual Test Effort for the automated tests.

C. Number of defects found by test automation.

D. Percentage of builds accepted/rejected by the automated tests.

ANSWER: B

Explanation:

Equivalent Manual Test Effort for the automated tests. is the most relevant metric because the stated project problem is that regression testing consumes most of the manual testers' capacity. Measuring the manual effort that would have been required to execute the automated regression tests quantifies the capacity released by automation. That released capacity can be redirected to testing new features, exploratory testing, and investigating risks earlier, helping the team address the immediate cause of late, defect-prone releases.

This measure should be based on a realistic, repeatable estimate of the time required for a person to prepare, execute, monitor, and report the same regression tests manually. Tracking it over releases makes the benefit visible in business terms that stakeholders can understand: tester time made available and the potential for faster feedback. It also supports a balanced assessment when considered alongside the ongoing cost of maintaining the automated tests, because automation has implementation and maintenance costs as well as execution benefits.

The ISTQB Test Automation Engineer certification describes automation as supporting efficient execution and faster feedback, while emphasizing that automation objectives and measures should align with project needs. Here, reduced manual regression effort directly aligns with that need. See the ISTQB Certified Tester Test Automation Engineer certification information and the ISTQB Certified Tester Foundation Level resources for the underlying testing and automation context.

QUESTION NO: 10

You are working on a web-based application called Book Vault that allows people to upload books and order books. This application must be available on all major browsers.

You have been testing the application manually and management have asked you to consider automating some of the tests.

You have investigated a number of commercial and free tools which can automate tests at a web browser level and one tool in particular meets your requirements and you have implemented a trial version.

You have basic programming skills and the main goal is to automate a few functional tests to see if the tool is compatible with the application and can recognise the objects and controls.

Which scripting technique would be MOST suitable in this scenario in order to meet the objectives?

A. Structured scripting

B. Capture-replay scripting

C. Data-driven scripting

D. Model-based scripting

ANSWER: B

Explanation:

Capture-replay scripting is most suitable because the immediate objective is a short, practical tool evaluation rather than the creation of a maintainable, scalable automation solution. The tester needs to automate a small number of representative

functional flows and establish whether the trial tool can drive the application in the required browsers, identify the relevant web objects, and interact reliably with its controls. Recording the tester's actions and replaying them provides a rapid proof of capability with little programming effort, which fits the stated level of scripting experience.

In this context, capture-replay gives useful early evidence about object recognition, synchronization behavior, browser interaction, and whether the tool can reproduce key user journeys such as uploading or ordering a book. It is therefore appropriate for an initial technical trial or proof of concept. Once compatibility has been demonstrated and automation is adopted more broadly, the recorded scripts would normally be refactored or replaced using a more maintainable scripting approach. ISTQB's Test Automation Engineer certification scope includes selecting and applying appropriate automation approaches and technologies; see the ISTQB Certified Test Automation Engineer certification page and the CT-TAE syllabus.