



Free Questions for TAE by dumpssheet

Shared by Rodriguez on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

The GUI of a Customer Relationship Management (CRM) application has been delivered through internet Explorer with proprietary Active X and Java controls. This implementation enables rich client capabilities, but specific commercial automation tools are necessary to automate test cases at GUI of functional test cases. This is to demonstrate whether a small set of the commercial are able to properly recognize actions taken by a tester when interacting with GUI of the CRM application.

Which of the following scripting techniques would be MOST suitable in this scenario?

Options:

- A- Data-driven scripting
- B- Keyword-driven scripting
- C- Linear scripting
- D- Structure scripting

Answer:

D

Question 2

Question Type: MultipleChoice

Which of the following statements BEST describe aspects of the SUT to consider when designing a TAA?

Options:

- A- All the interaction between SUT and TAS should be logged with the highest level of detail
- B- All the internal test interfaces of the SUT should be removed prior to the product release
- C- All the interface of the SUT affected by the tests should be controllable by the TAA
- D- All the external test interfaces of the SUT should be removed prior to the product release

Answer:

A

Question 3

Question Type: MultipleChoice

Which of the following is NOT a technical design consideration for a TAA?

Options:

- A- The number of users for the SUT
- B- Availability of interfaces for the SUT to be testable
- C- Standards and Legal requirements, e.g data privacy
- D- Data used by the SUT, e.g configuration, users

Answer:

A

Question 4

Question Type: MultipleChoice

Which of the following BEST describes why it is important to separate test definition from test execution in a TAA?

Options:

- A- It allows developing steps of the test process without being closely tied to the SUT interface.
- B- It allow choosing different paradigms (e.g event-driven) for the interaction TAS and SUT
- C- It allows specify test cases without being closely tied to the tool to run them against the SUT
- D- It allows testers to find more defects on the SUT

Answer:

C

Question 5

Question Type: MultipleChoice

You are currently designing the TAA of a TAS. You have been asked to adopt an approach for automatically generating and executing test cases from a model that defines the SUT. The SUT is a state-based and event-driven that is described by a finite-state machine and exposes its functionality via an API. The behavior of the SUT depends on hardware and communication links that can be unreliable.

Which of the following aspects is MOST important when designing the TAA in this scenario?

Options:

- A- Looking for tools that allows direct denoting of exceptions and actions depending on the SUT events.
- B- Adopting a test definition strategy based on classification tree coverage for the test definition layer.
- C- Looking for tools that allow performing setup and teardown of the test suites and the SUT.
- D- Adopting a test definition strategy based on use case/exception case coverage for the definition layer.

Answer:

C

Question 6

Question Type: MultipleChoice

Consider the following layers of the gTAA structure:

- a. Test generation layer
- b. Test definition layer
- c. Test execution layer
- d. Test execution layer

Consider the following capabilities associated with these layers.

Acquire all the necessary resources before each test and release all after run, in order to avoid interdependences between test

Allow the automated test scripts on an abstract level to interact with components, configurations and interfaces of the SUT.

Design test directives that allow configuring the algorithms used to automatically produce the test cases a given model of the SUT.

Allow the definition and implementation of test cases and data by means of templates and/or guidelines.

Which of the following BEST matches each layer with the appropriate capability?

Options:

A- a-3, b-4, c-1, d-2

B- a-4, b-3, c-1, d-2

C- a-4, b-3, c-2, d-1

D- a-3, b-4, c-2, d-1

Answer:

C

Question 7

Question Type: MultipleChoice

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?

Options:

- 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

Question 8

Question Type: MultipleChoice

You are reviewing the testability of your SUT.

Which of the following BEST refers to the characteristic of OBSERVABILITY?

Options:

- A- The ability of the SUT to perform its intended function for a specified period of time
- B- The ability to exercise the SUT by entering inputs, triggering events and invoking methods
- C- The ability of the SUT to prevent unauthorized access to its components or data.
- D- The ability to identify states, outputs, intermediate result and error messages in the SUT

Answer:

D

Question 9

Question Type: MultipleChoice

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 properly after these updates.

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

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

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

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

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?

Options:

A- A and C

B- A and E

C- B and E

D- C and D

Answer:

Question 10

Question Type: MultipleChoice

You are working on a TAS for standalone application. The automated tests are developed based on a automation framework that allows interaction with GUI elements using on object orientated API. The GUI elements include menus, buttons, radio buttons, text toolbars and their properties.

Whilst automating a test, you have discovered that the GUI elements of some third party components are not identifiable by the automated tool you are using.

Which of the following is the FIRST step that you take to investigate this issue?

Options:

- A- Verify the testability support with the providers of the third party components
- B- Verify whether the GUI identification depends on the browser.
- C- Adopt an approach that uses the coordinates of the GUI elements instead
- D- Verify whether naming standards for variables and have been defined for the current automation solution

Answer:

A

Question 11

Question Type: MultipleChoice

You have been asked to determine a TAS for a new release of a SUT, test should be automated wherever. The new release will consist of 5 new interfaces and an amendment to 3 existing interfaces. The new and amended interface will be delivered incrementally in 3 sprints, each lasting 2 weeks.

What would be the BEST Test Automation Solution (TAS) design in this scenario?

Options:

- A-** Automate tests at both Component and System Level. Only do this automation once every interface has been fully developed or amended and manual testing has completed successfully.
- B-** Automate tests at one level only, System level. Use only the newly developed interfaces and do not create any customized interfaces/test hooks.
- C-** Automate the tests at two levels, Component and System level. Create customized hooks at Component level for interface not yet developed or amended. Only use the newly developed or amended interfaces to test at System level.

D- Automate a test at once level, component level, Create customized interface/test hooks for this level where the interface has not yet been developed or amended.

Answer:

A

To Get Premium Files for TAE Visit

<https://www.p2pexams.com/products/tae>

For More Free Questions Visit

<https://www.p2pexams.com/bcs/pdf/tae>

