[Q11-Q30 TAE Certification - The Ultimate Guide [Updated 2022

TAE Certification - The Ultimate Guide [Updated 2022]
TAE Practice Exam and Study Guides - Verified By BraindumpsIT

Q11. 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?
* Observability: The SUT needs to provide interface that give insight into the system.
* Re-useability: The code written for the SUT must be re-useable for other similar system.
* Clearly defined architecture: The SUT Architecture needs to provide clear and understandable interfaces giving control and visibility on all test levels.
* Control: the SUT needs to provide interfaces that can be used to perform actions on SUT.

Q12. Which of the following statements BEST describe aspects of the SUT to consider when designing a TAA?
* All the interaction between SUT and TAS should belogged with the highest level of detail
* All the internal test interfaces of the SUT should be removed prior to the product release
* All the interface of the SUT affected by the tests should be controllable by the TAA
* All the external test interfaces of the SUT should be removed prior to the product release

Q13. Which of the following statements about the reuse of TAS artefacts is TRUE?
- Reusable TAS artefacts can include components (or parts of components) associated with different layers of the TAA
- 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
- Communications maintenance and improvements for reusing TAS artefacts are modify addressedduring the design of the TAA
- Reusable TAS artifacts associated with the definition layer of the TAA include the adaptors to the SUT components and/or interfaces

Q14. A TAS uses a commercial test automation tool and the default logs generated by the inconsistent formats such as different types of messages (pass/fail steps, screenshots, warnings, etc.) To solve this issue some custom logging functions have been created from the test scripts, making it possible tolog the different types of messages with the same format. However, this may cause a problem due to excessive size of the logs which can make it difficult to find the required information. Assume that all the default logs will be disabled when running theautomated tests and that some tests will not generate excessively sized logs.

Which of the following represents the BEST suggestion for implementing the custom logging functions?
- Implement the custom logging functions without saving timestamps
- Implement the custom logging functions to support different levels of tracing
- Implement the custom logging functions without saving stack traces
- Implement the custom logging functions to redirect the logs to multiple files

Q15. 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?
- The connectivity from the TAS to the dedicatedtest interfaces will not work
- The process of configuring the TAS will be error-phone due to manual intervention
- The automated test cases will not contain the expected result
- False alarms, that are unlikely to occur in the real world, will be observed during testing

Q16. Which of the following attributes should NOT be included in a test execution report associated with a suite of automated tests?
- Summary of the test execution results
- System/Application under test and its version
- Defect clusters identified during test execution
- Environment in which the tests have been executed

Q17. 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 thenecessary 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?
* a-3, b-4, c-1, d-2
* a-4, b-3, c-1, d-2
* a-4, b-3, c-2, d-1
* a-3, b-4, c-2, d-1

Q18. Consider a TAS that uses a keyword-driven framework. The SUT is a web application and there is a large set of keywords available for writingthe automated tests that relate to highly specific user actions linked directly to the GUI of the SUT. The automated test written with the keywords are statically analyzed by a custom tool which highlight&#8217;s repeated instances of identical sequence of keywords. The waiting mechanism implemented by the TAS for a webpage load is based on a synchronous sampling within a given timeout. The TAS allows checking a webpage load every seconds until a timeout value
* Changing the scripting approach to data-driven scripting
* Implementing keywords with a higher level of granularity
* Changing the wait mechanism to explicit hard-coded waits
* Establishing an error recovery process for TAS and SUT

Q19. Your goal is to verify completeness, consistency and correct behavior of an automated test suite. The TAS has been proven to successfully install in the SUT environment. All the preliminary checks toverify the correct functioning of the automated test environment and test tool configuration, installation and setup have successfully completed.

Which of the following is NOT a relevant check for achieving your goal in this scenario?
* Checking whether all the test cases contain the expected results
* Checking whether the post condition have been fulfilled for all the test cases
* Checking whether the loading of the TAS is repeatable in the SUT environment
* Checking whether all the test casesproduce repeatable outcomes

Q20. You are reviewing the testability of your SUT.

Which of the following BEST refers to the characteristic of OBSERVABILITY?
* The ability of the SUTto perform its intended function for a specified period of time
* The ability to exercise the SUT by entering inputs, triggering events and invoking methods
* The ability of the SUT to prevent unauthorized access to its components or data.
* The abilityto identify states, outputs, intermediate result and error messages in the SUT

Q21. Consider A TAS for testing adesktop application via its GUI. All the test cases of the automated test suite contain the same identical sequences of steps at the beginning (to create the necessary objects when doing a preliminary configuration of the test environment and at the end (to remove everything created -specifically for the test itself during the preliminary configuration of the test environment). All automated test cases use the same set of assertion functions from a shared library, for verifying the values in the GUI fields(e.g text boxes).

What is the BEST recommendation for improving the TAS?
* Implementing keywords with higher level of granularity
* Improving the architecture of the application in order to improve its testability
* Adopting a set of standard verification methods for use by all automated tests

* Implementing standard setup and teardown functions at test case level

Q22. 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?
* Data-driven scripting
* Keyword-driven scripting
* Linear scripting
* Structure scripting

Q23. A web application was released into production one year ago, it has regular release which follow a V-model lifecycle and testing is well-established and fully integration into the development lifecycle. You have been asked to implement a TAS for the regression test suite. The regression tests have been developed via the GUI and are expected to be run at least four times a month, for each planned release, for the whole operation solution life of the system (six years). Each screen of the GUI uses several third-party controls which are not compatible with the existing automation solutions. The environment for the automation will be stable, fully controllable and separated from other environments (development, staging, production).

What could be the MOST problematic for this TAS?
* Maturity of the test process
* Complexity to automate
* Frequency of use
* Sustainability of the automated environment

Q24. 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?
* The test reporting component of the test execution layer.
* The Test execution component of the test generation layer
* The test execution (test engine of the test execution layer
* The Command Line Interface (CLI) component of the test adaptation layer

Q25. Which of the following BEST describes why it is important to separate test definition from test execution in a TAA?
* It allows developing steps of the test process without being closely tied to the SUT interface.
* It allow choosing different paradigms (e.g event-driven) for the interaction TAS and SUT
* It allows specify test cases without being closely tied to the tool to run them against the SUT
* It allows testers to findmore defects on the SUT

Q26. 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 possibleand the scripts must be highly reusable.

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

* The keyword-driven framework is not suitable, it would be better to adopt a structured-scripting approach
* False positive errors are likely to occurwhen running the automated tests on the new releases without modifying the test
* The total execution time of the automated regression test suite will decrease for each planned release.
* The keyword-driven framework introduces a level abstraction that is too high and makes it difficult what really happens