KillTest

Mejor calidad. Mejor servicio



Preguntas de práctica

https://www.killtest.es

Actualizaciones gratuitas durante un año

Exam : **CTAL-TTA_SyII19_4.0**

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

Version : DEMO

1.Assume you are involved in testing a Health Insurance Calculation system.
At the main screen one can enter information for a new client. The information to be provided consists of last name, first name and date of birth. After confirmation of the information, the system checks the age of the potential new client and calculates a proposed premium.
The system also has the option to request information for an existing client, using the client's ID number.
A keyword-driven automation approach is being used to automate most of the regression testing.
Based on the information provided, which TWO of the options provided would be the MOST LIKELY keywords for this application? (Choose two.)
A. Remove_Client
B. Enter_Client
C. Print_Premium
D. Select_Client
E. Exclude_Client
Answer: B, D
Explanation:
Considering the functionalities described for the Health Insurance Calculation system, the keywords would represent the main actions that can be performed in the system. 'Enter_Client' would be a keyword for entering new client information, which is a primary feature of the system as described. 'Select_Client' would be used to retrieve information for an existing client using the client's ID number, which is another main functionality. Other options such as 'Remove_Client', 'Print_Premium', and 'Exclude_Client'
are not explicitly mentioned in the provided system functionalities, therefore, 'Enter_Client'
and 'Select_Client' are the most likely keywords for automation.

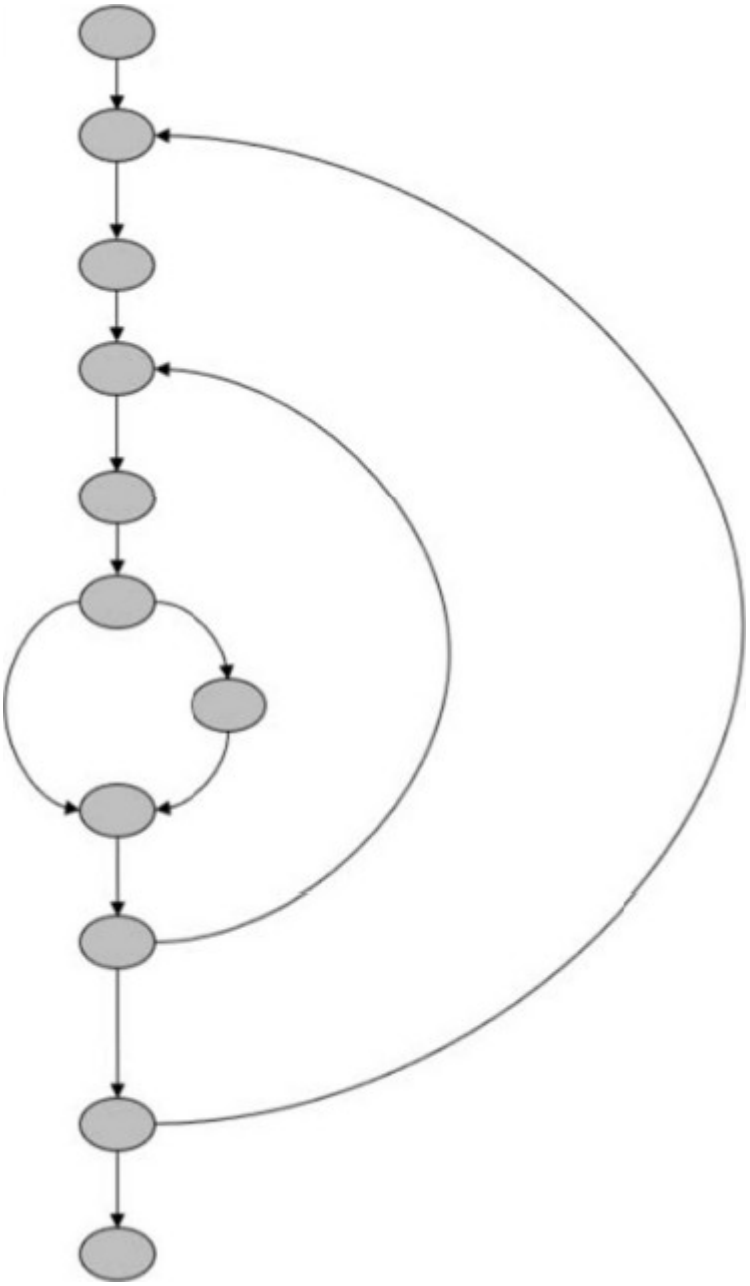
2.At which test level would reliability testing most likely be performed?
A. Static testing
B. Component testing
C. System testing
D. Functional acceptance testing
Answer: C
Explanation:
Reliability testing is aimed at verifying the software's ability to function under expected conditions for a specified period of time. It is typically conducted during system testing, where the software is tested in its entirety to ensure that all components work together as expected in an environment that closely simulates the production environment. Reliability testing is not typically associated with static testing, component testing, or functional acceptance testing, as these levels of testing do not address the overall behavior of the system over time.

3.You are working on project where re-use of software is an objective. You are involved in the project as a Technical Test Analyst and have been given the task to develop a checklist for code reviews.
Which question from the list below should you implement as part of the code review checklist?
A. Are all modules, data, and interfaces uniquely identified?
B. Can each item be implemented with the techniques, tools, and resources available?
C. Is it possible during acceptance testing to verity whether the item has been satisfied?

D. Are all variables defined with meaningful, consistent and clear names?
Answer: A
Explanation:
For a project where the reuse of software components is a key objective, it is essential to ensure that all modules, data, and interfaces are uniquely identified. This facilitates the tracking, maintenance, and reuse of these components in different parts of the software or in other projects. Unique identification helps in preventing confusion and conflicts that may arise due to the reuse of components that have the same name but different functionalities or interfaces.

4.Which of the following statements about performance testing tools is NOT correct?
A. Typical metrics and reports provided by performance testing tools include the number of simulated users throughout the test, and the number and type of transactions generated by the simulated users, and the arrival rate of the transactions.
B. Significant factors to consider in the implementation of performance testing tools include the flexibility of the tool to allow different operational profiles to be easily implemented, and the hardware and network bandwidth required to generate the load.
C. Performance testing tools typically drive the application by simulating user interaction at the graphical user interface level to more accurately measure response times.
D. Performance testing tools generate a load by simulating a large number of virtual users following their designated operational profiles to generate specific volumes of input data.
Answer: C
Explanation:
The statement about performance testing tools that is NOT correct is that they typically drive the application by simulating user interaction at the graphical user interface (GUI) level to more accurately measure response times. In practice, performance testing tools often simulate user interactions at a protocol or service level rather than the GUI level. This approach allows the tools to generate a high load by simulating many virtual users, which would be challenging to achieve with GUI-level interactions due to the higher resource consumption and slower execution speed associated with GUI automation. Performance testing tools are designed to assess the performance of a system under a particular load and are not primarily focused on the user interface. They simulate multiple users accessing the system simultaneously, which helps in identifying bottlenecks, understanding the system's behavior under load, and determining how the system scales with increasing load. The tools typically simulate user requests to the server, bypassing the GUI to directly test the backend, APIs, or other service endpoints. This method allows for more efficient and scalable testing, enabling the simulation of thousands of users without the overhead of rendering the GUI.

5.Consider the following control flow graph:



The control flow represents a software component of a car navigation system. Within the project the maximum cyclomatic complexity to be allowed is set at 5.
Which of the following statements is correct?
A. No defect needs to be reported since the cyclomatic complexity of the component is calculated at 3.
B. No defect needs to be reported since the cyclomatic complexity of the component is calculated at 4
C. No defect needs to be reported since the cyclomatic complexity of the component is calculated at 5.
D. A defect needs to be reported since the cyclomatic complexity of the component is calculated at 6.
Answer: D
Explanation:
Cyclomatic complexity is a measure of the number of linearly-independent paths through a program's source code, which is often used as a measure of the complexity of a program. The control flow graph provided represents the logic of a software component and has more than 5 nodes with decision points, indicating that the complexity would exceed the maximum allowed value of 5. The calculation for

cyclomatic complexity is V(G) = E - N + 2P, where E is the number of edges, N is the number of nodes, and P is the number of connected components. In this case, the calculated cyclomatic complexity exceeds the allowed threshold, thus a defect should be reported.