Free Questions for CTAL-TTA by go4braindumps

Shared by Melendez on 14-02-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

As a technical test analyst, you are involved in a risk analysis session using the Failure Mode and Effect Analysis technique. You are calculating risk priorities. Which of the following are the major factors in this exercise?

Options:

A- Severity and priority

B- Functionality, reliability, usability, maintainability, efficiency and portability

C- Likelihood and impact

D- Financial damage, frequency of use and external visibility

Answer:

C

Explanation:

Failure Mode and Effect Analysis (FMEA) is a structured approach to identify and address potential failures in a system, product, process, or service. The major factors involved in calculating risk priorities in FMEA are typically the severity of the potential failure, its likelihood of occurrence, and the ability to detect it. These factors are usually combined to form a Risk Priority Number (RPN) for each potential failure mode identified. However, the specific factors mentioned in the options like functionality, reliability, usability, maintainability, efficiency, and portability are quality characteristics that could be considered in an FMEA analysis but are not directly used for calculating risk priorities. Likewise, financial damage, frequency of use, and external visibility might influence the severity or impact of a failure, but they are not standard factors in calculating risk priorities in the context of FMEA. Therefore, the most relevant factors for calculating risk priorities in an FMEA context would typically be the likelihood of the failure occurring and its potential impact, which aligns with option C: Likelihood and impact.

It's important to note that while these explanations are based on general principles and practices related to fault seeding and FMEA, the specifics might vary slightly in different contexts or with different methodologies.

Question 2

Question Type: **MultipleChoice**

Which of the following statements about fault seeding tools is correct?

Options:

A- Fault seeding tools insert defects into the source code to check the effectiveness of testing.

B- Fault seeding tools insert defects into the source code to test the input checking capabilities of the software.

C- Fault seeding tools insert defects into the source code to support the application of specification-based test design techniques.

D- Fault seeding tools insert defects into the source code to check the level of maintainability of the software.

Answer:

A

Explanation:

Fault seeding is a method used to evaluate the effectiveness of a testing process. Tools designed for fault seeding intentionally insert known defects into the source code, which are then supposed to be discovered during testing. The main purpose is not to check the input checking capabilities, support specification-based test design techniques, or assess maintainability of the software, but rather to gauge how well the testing process can identify and capture defects. By comparing the number of seeded faults that are found against the total number of faults inserted, test teams can get an insight into the effectiveness of their testing strategies and coverage. This method helps in understanding the detection capabilities of testing efforts and in identifying potential areas for improvement in test processes.

Question 3

Consider the pseudo code for the Answer program:

```
01      MAIN "The ANSWER Program"
02
03      a, b, c, d: integer
04      BEGIN
05      Read (a)
06      Read (b)
07      d = 5
08
09      WHILE a < d
10        c = b + d
11        a = 2
12        b = a + 10
13
14        IF b = 12 THEN print (b)
15        ELSE print (a)
16        a = a + 2
17        ENDIF
18
19      ENDWHILE
20
21      Print ("the answer is")
22      Print (c)
23
24      END
```

Which of the following statements about the Answer program BEST describes the control flow anomalies to be found in the program?

Options:

A- The Answer program contains no control flow anomalies.

B- The Answer program contains unreachable code.

C- The Answer program contains unreachable code and an infinite loop.

**D-** The Answer program contains an infinite loop.

Answer:

C

Explanation:

The provided pseudo code for the Answer program shows a WHILE loop that will always execute because the condition for the loop to terminate (a >= d) is never met within the loop's body. This results in an infinite loop. Additionally, since the value of 'b' is initialized with 'a + 10' and 'a' starts from a value that is read and then set to 2, 'b' will never be equal to 12. Therefore, the 'THEN' branch of the IF statement, which includes 'print(b)', is unreachable. These are control flow anomalies because they represent logic in the code that will not function as presumably intended.

Question 4

Consider the following control flow graph:
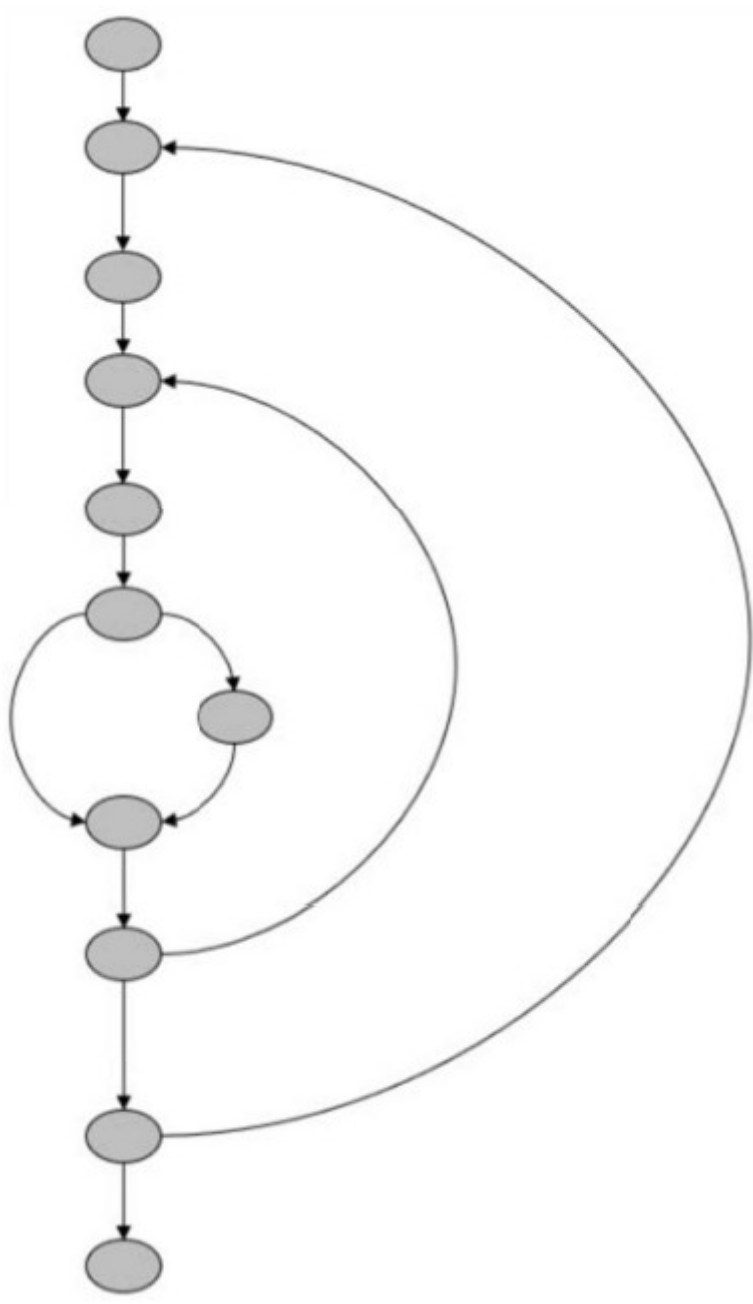
The control flow represents a software component of a car navigation system. Within the project the maximum cyclomatic complexity to be allowed is set at 5.

Which of the following statements is correct?

Options:

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.

Question 5

At which test level would reliability testing most likely be performed?

Options:

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.

Question 6

Question Type: **MultipleChoice**

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.)

Options:

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.

Question 7

Within the world of consumer electronics, the amount of embedded software is growing rapidly. The amount of software in high-end television sets has increased by a factor of about eight over the last six years. In addition, the market of consumer electronics has been faced with a 5 -10% price erosion per year. The price of a product is, among a number of other things, determined by the microcontroller used. Therefore, the use of ROM and RAM remains under high pressure in consumer electronic products, leading to severe restrictions on code size.

You are a Technical Test Analyst involved in the review of the architecture of this project.

Which of the following issues would be MOST important to focus on during the review and when verifying the correct implementation?

Options:

A- Connection pooling

B- Caching

C- Transaction concurrency

D- Lazy instantiation

Answer:

D

Explanation:

The key context here is the challenge of managing limited resources, particularly ROM and RAM, due to severe restrictions on code size in consumer electronics. Lazy instantiation is a design pattern that defers the creation of an object until the first time it is needed. This approach can significantly reduce the application's memory footprint by avoiding unnecessary pre-allocation of memory, which is particularly valuable in systems where memory resources are constrained. In reviewing the architecture for such a system, it's crucial to ensure that objects are only created when necessary and that memory is optimally managed. Hence, the focus on lazy instantiation would be most important to ensure that the system uses resources efficiently and remains within the restricted code size.