**Interview question**

**1.** **Why do you need a test case?**

Know which features need to be tested.

Know how many test cases are executed.

Know how many features and stable functions there are.

Easily hand over to other members.

Save time when updating software versions.

Know which function needs more work depending on the number of errors found in the function.

—--------Võ Kim Tuyến—--------

**2.** **Test case format ?**

There are columns Id, Category, Title, Preconditions, Steps, Expected results, Data, status, Note, including the 3 most important columns: Title, Steps, Expected results.

—--------Võ Kim Tuyến—--------

**3.** **What is test case priority? Why do you need this field?**

Priority is the order of when a bug should be fixed. The higher the priority, the sooner it should be resolved. Bugs that make software unusable are given higher priority than a small faulty function.

Why: so the development team can know which bugs are more serious and need to be fixed first.

—--------Võ Kim Tuyến—--------

**4.** **What are best practices for writing test cases?**

- Define Clear Objectives: Each test case should have a clear purpose. Define what is being tested and the expected outcome.

- Maintain Simplicity and Clarity: Write test cases in simple and clear language. Avoid complex sentences and jargon to ensure that anyone reading the test case can understand it.

- Test cases are created with end user in mind, more than development view

- Do not repeat test cases, Do not assume

- Ensure Coverage: Aim for comprehensive coverage of all possible scenarios, including positive, negative, edge cases, and boundary conditions.

- Setting priority when writing test cases is crucial for several reasons including: Focus on Critical Areas, Risk Management, Improved Coverage, Enhance overall application quality and customer satisfaction

- Peer review

- - Categorize/group set of test cases

- - Implement testing techniques

—--------Vũ Phương Thảo—--------

**5.** **How many test cases can you execute/write in a day?**

The number of test cases one can execute or write in a day can vary greatly depending on several factors, such as the complexity of the test cases, the nature of the application being tested, the tools and resources available, and the individual's experience and expertise. So there are definitely no specific numbers for this question

—--------Vũ Phương Thảo—--------

**6.** **What is test data?**

Test data is a set of data used in software testing to simulate real-world inputs. Its purpose is to ensure applications function correctly under various conditions and to validate performance, reliability, and security. Different types of test data, such as valid, invalid, boundary, and synthetic data, are used to cover diverse testing scenarios and ensure thorough application testing.

—--------Vũ Phương Thảo—--------

**7.** **How many test design techniques do you know? Short description and example?**

Several test design techniques commonly used in software testing:

**Equivalence Partitioning:**

*Description*: Divides input data into partitions of equivalent data.

*Example*: If a function accepts numbers from 1 to 1000, test cases might include inputs like 0, 1, 500, 1000, and 1001.

**Boundary Value Analysis:**

*Description*: Tests boundaries between partitions of input values.

*Example*: For a function accepting numbers from 1 to 1000, test cases might include 1, 2, 999, 1000, and values just outside these boundaries (0, 1001).

**Decision Table Testing:**

*Description*: Tests combinations of inputs and outputs based on decision rules.

*Example*: For a login system, combinations could include valid and invalid usernames and passwords to test different outcomes.

**State Transition Testing:**

*Description*: Tests the system's behavior based on transitions between different states.

*Example*: For an ATM, states could include idle, card inserted, PIN entered, amount entered, transaction complete, etc.

**Pairwise Testing (Combinatorial Testing):**

*Description*: Tests all possible pairs of input parameters.

*Example*: If a system has parameters A, B, and C with possible values, pairwise testing ensures each pair (A, B), (A, C), and (B, C) is tested.

**Pairwise Testing (Combinatorial Testing):**

*Description*: Tests all possible pairs of input parameters.

*Example*: If a system has parameters A, B, and C with possible values, pairwise testing ensures each pair (A, B), (A, C), and (B, C) is tested.

**Error Guessing:**

*Description*: Testers use their intuition and experience to anticipate where errors might occur.

*Example*: Based on experience, a tester might deliberately enter special characters in a text field to see if the system handles them correctly.

**Exploratory Testing:**

*Description*: Simultaneous test design and execution where tests are dynamically designed and executed.

*Example*: Exploring a new feature in an application without a predefined test script to uncover bugs or issues.

—--------Nguyễn Thị Quỳnh Quyên—--------