

# SE 317, Quiz 1

Name: Aina Qistina Binti Azman

Net ID: 457 464 051

## Chapter 1: Why Do We Test Software? (3 Questions)

1. What is the definition of:

- a. Software Fault?
- b. Software Error?
- c. Software Failure?

Answer:

- a. Software Fault: A static defect in the software.
- b. Software Error: An incorrect internal state that is the manifestation of some fault. This includes syntax error, logic error, calculations, boundary, etc.
- c. Software Failure: External, incorrect behavior with respect to the requirements or another description of the expected behavior.

2. The acronym *IV&V* stands for "Independent Verification and Validation". Provide the definition of Verification & Validation (IEEE).

Answer:

- Verification: The process of determining whether the products of a given **phase** of the software development process fulfill the requirements established during the previous phase.
- Validation: The process of evaluating software at the **end** of software development to ensure compliance with the intended usage.

3. Beizer discussed the goals of testing in terms of the “test process maturity levels” of an organization, where the levels are characterized by the testers’ goals. Match the below purposes to its correct level.

- a. The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.
- b. The purpose of testing is to show correctness.
- c. The purpose of testing is to show that the software does not work.
- d. There is no difference between testing and debugging.
- e. Testing is a mental discipline that helps all IT professionals develop higher-quality software.

Answer:

Level 0	d. There is no difference between testing and debugging.
Level 1	b. The purpose of testing is to show correctness.
Level 2	c. The purpose of testing is to show that the software does not work.
Level 3	a. The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.
Level 4	e. Testing is a mental discipline that helps all IT professionals develop higher-quality software.

## Chapter 2: Model-Driven Test Design (4 Questions)

4. Choose the conditions that are necessary for a failure to be observed.

- ☐ Infection
- ☐ Propagation
- ☐ Reveal
- ☐ Reachability
- ☐ All of the above

Answer:

- All of the above

5. A test engineer is an IT professional who is in charge of one or more technical test activities. What are their tasks as a test engineer?

- ☐ Producing test values
- ☐ Designing test inputs
- ☐ Fix problems found in the software being tested
- ☐ Running test scripts
- ☐ Analyzing results
- ☐ All of the above

Answer:

- Producing test values
- Designing test inputs
- Running test scripts
- Analyzing results

6. List three advantages of coverage criteria.

Answer:

1. Make regression testing easier.
2. Provide traceability from software artifacts to tests.
3. Can be well supported with powerful tools.
4. Gives testers a "stopping rule".
5. Maximize the "bang for the buck".

7. Label the general activities below whether it is:

- Test design
- Test automation

- Test execution
- Test evaluation

Activities	Type of activities
pass/fail	
model/structure → test requirements → refined requirements/test specs	
test results	
test scripts → test cases	

Answer:

Activities	Type of activities
pass/fail	Test Evaluation
model/structure → test requirements → refined requirements/test specs	Test Design
test results	Test Execution
test scripts → test cases	Test Automation

## Chapter 3: Test Automation (4 Questions)

8. What is test automation? Provide the definition of test automation and three of its advantages.

Answer:

- Definition of Test Automation

The use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.

- Advantages of Test Automation
  - Reduces human error
  - Reduces cost

- Reduces variance in test quality from different individuals
- Significantly reduces the cost of regression testing

9. Most test automation frameworks support:

- ☐ Test sets to easily organize and run tests
- ☐ Assertions to evaluate expected results
- ☐ The ability to share common test data among tests
- ☐ The ability to run tests from either a command line or a GUI
- ☐ All of the above

Answer:

- All of the above

10. Explain the purpose and process of using the `@Before` annotation in JUnit tests.

Answer:

The `@Before` annotation in JUnit is used to define a method that will always be executed first before each test method in a test class. This is generally used for common setup code that is needed by multiple test methods. This practice helps to avoid code duplication and improve readability.

11. How does the assertion `fail(String)` works? Provide an example of how the `fail` method might be used to test exceptional behavior.

Answer:

The `fail` method in JUnit is used to indicate that a test has failed if a certain section of code is reached. This method is particularly useful in scenarios where the test should not reach a particular point if the code is functioning correctly. `fail` method takes a `String` parameter that provides a summary message, explaining why the test failed to the test engineer.

Example:

```

@Test
public void testExceptionThrown(){

    try{

        aMethodThatSupposeToThrowException();

        //If the exception is not thrown, the following line will be executed
        fail("Expected exception was not thrown");
    }

    catch(ExpectedExceptionType e){
        //Test passes if the expected exception is caught
        assertEquals("Expected exception message",e.getMessage());
    }
}

```

## Chapter 7: Embedded Systems & Graph Testing (5 Questions)

12. *Figure 12.a* below shows three graphs. Identify

- a) all the nodes in each of the graph
- b) the initial nodes of each graph
- c) the number of edges of each graph

Graph A

Graph B

Graph C

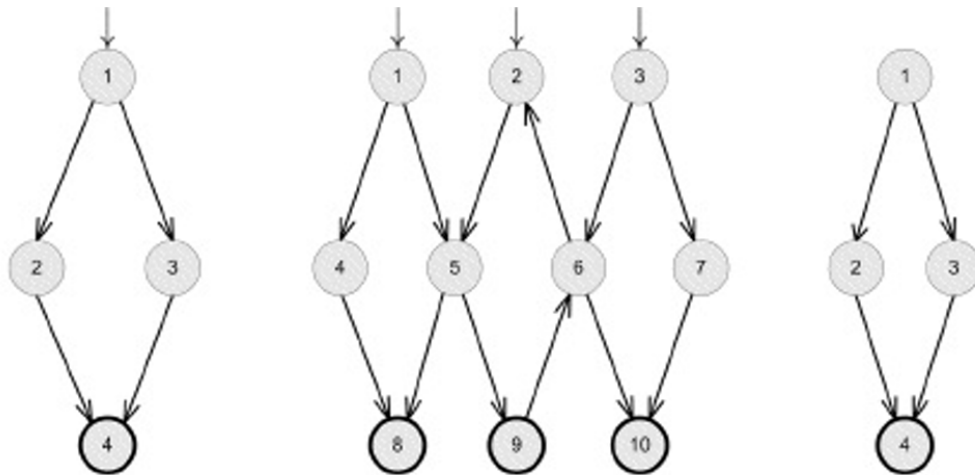


Figure 12.a

Answer:

- Graph A
  - a) Nodes = {1, 2, 3, 4}
  - b) Initial Nodes = {1}
  - c) Number of Edges = 4
  
- Graph B
  - a) Nodes = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
  - b) Initial Nodes = {1, 2, 3}
  - c) Number of Edges = 12
  
- Graph C
  - a) Nodes = {1, 2, 3, 4}
  - b) Initial Nodes = {}
  - c) Number of Edges = 4

13. From Figure 12.a, there is a graph that is not useful for generating test cases. What graph is it and why?

Answer:

Graph C is not useful for generating test cases. The reason why it is not useful for generating test cases is because there is no starting point to begin the execution or traversal. In the context of software testing, an initial node represents the starting state from which the tests begin. Without an initial node, it is impossible to determine where the testing process should start, resulting it to be impractical to generate meaningful test cases.

14. Given the definition below:

*"TR contains at least one round-trip path for each reachable node in G that begins and ends a round-trip path."*

Which criterion does the above definition matches to?

- ☐ Complete Round Trip Coverage (CRTC)
- ☐ Simple Round Trip Coverage (SRTC)
- ☐ Complete Path Coverage (CPC)
- ☐ Specified Path Coverage (SPC)
- ☐ none of the above

Answer:

Simple Round Trip Coverage (SRTC)

15. Provide the definition of

- a. Tour
- b. Tour with Sidetrips
- c. Tour with Detours

Answer:

- a. Tour: Test path  $p$  is said to tour subpath  $q$  if and only if  $q$  is a subpath of  $p$
- b. Tour with Sidetrips: Test path  $p$  is said to tour subpath  $q$  with sidetrips if and only if every edge in  $q$  is also in  $p$  in the same order.



- c. Tour with Detours: Test path  $p$  is said to tour subpath  $q$  with detours if and only if every node in  $q$  is also in  $p$  in the same order.

16. Provide the example of flow to a) sidetrips and b) detours based on the given graph of Figure 16. The example for tour has been provided.

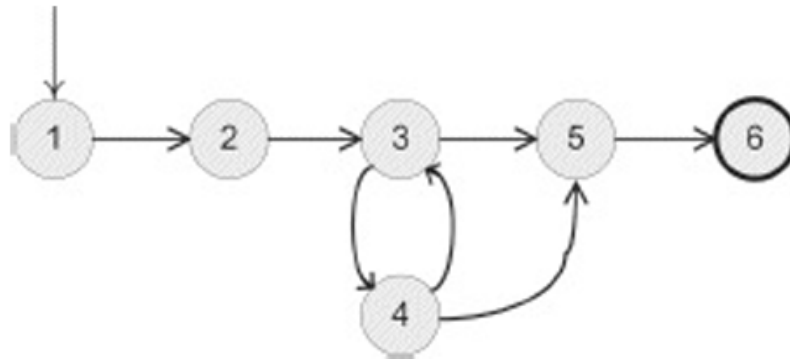


Figure 16

- Tour = {1, 2, 3, 5, 6}

Answer:

a) Sidetrips = {1, 2, 3, 4, 3, 5, 6}

b) Detours = {1, 2, 3, 4, 5, 6}