B4.3-R3: SOFTWARE TESTING AND QUALITY MANAGEMENT

NOTE:

1. **Answer question 1 and any FOUR questions from 2 to 7.**
2. **Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1. State whether the following statements are **TRUE** or **FALSE.** In each case, justify your answer.

a) During code review one detects errors, whereas during code testing one detects failures.

b) The effectiveness of a test suite in detecting errors can be determined by counting the total number of test cases present in the test suite.

c) Usually compliance to coding standards are verified during system testing.

d) Modern quality assurance paradigms are centered around carrying out thorough product testing.

e) A pure top-down integration testing does not require the use of any stub modules.

f) A satisfactory way to test object-oriented programs is to test all the methods supported by the different classes individually.

g) Path coverage is a stronger testing technique compared to statement coverage technique.

(7x4)

2.

a) Explain, how the different defects in a system can be classified. Why is it necessary to classify the defects into several classes?

b) What are stress and volume testing? What is the difference between these two types of testing? How are they performed? Give some examples of stress and volume testing.

c) Which type of testing: unit, integration or system testing is best suited to test usability of a software product? How is usability of a software product tested?

(6+6+6)

3.

a) What is the difference between a coding standard and a coding guideline? Why are these considered important in a software development organization?

b) List two coding standards each for (i) enhancing readability of the code, (ii) reuse of the code.

c) Define three metrics to measure software reliability. Do you consider these metrics entirely satisfactory to provide measure of the reliability of a software system? Justify your answer.

(6+6+6)

4. Consider the following program segement.

```
void sort(int a[], int n){
     int i,j;

      for(i=0;i<n-1;i++)
          for(j=i+1;j<n;j++)
           if(a[i]>a[j])
           {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
           }
     }
```

a) Draw the control flow graph for above program segment.

b) Determine the cyclomatic complexity for above program. (Show all the intermediate steps in your computation.)

c) How is the cyclomatic complexity metric useful?

(6+6+6)

5.

a) Explain what do you understand by client-server software. What are its advantages over the traditional software architecture? How can a client-server software be effectively tested?

b) What do you understand by Key Process Area (KPA), in the context of SEI CMM? Would there be any problem if an organization tries to implement higher level SEI CMM KPAs before achieving lower level KPAs? Justify your answer using suitable examples.

c) What do you mean by process metric? Explain at least one process metric and how this metric can be collected. Also explain, how defects can be effectively tracked for a software product.

(6+6+6)

6.

a) What problems would you face if you are developing several versions of the same product according to a client's request and not using any configuration management tools?

b) What is a real-time system? Why effective testing of real-time and embedded systems is considered more difficult than testing traditional systems? Explain a satisfactory scheme to test real-time and embedded systems.

c) Design the black-box test suite for a program that accepts two strings and checks if the first string is a substring of the second string and displays the number of times the first string occurs in the second string.

(6+6+6)

7.

a) What do you understand by the term integration testing? Which types of defects are uncovered during integration testing? What are the different types of integration testing methods that can be used to carry out integration testing of a large software product? Compare the merits and demerits of these integration testing strategies.

b) Distinguish between software verification and software validation. When do you perform verification and validation in the context of software life cycle? Can one be used in place of the other?

c) What do you understand by data flow testing? How is data flow testing performed? Is it possible to design data flow test cases manually? Explain your answer.

(6+6+6)