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.

a) How the cost of repairing defects in the software can be minimized? Discuss with examples.

b) Differentiate between verification and validation. Can a proof of correctness of software be provided?

c) “Complete testing of software is just not possible.” State whether this statement is true or false and illustrate with an example.

d) Explain the boundary value analysis testing technique with the help of an example.

e) What is a satisfactory way of testing object oriented software?

f) Discuss main objectives of performing measurement in software engineering domain.

g) Give main features of the Capability Maturity Model (CMM).

(7x4)

2.

a) Discuss the role of software testing during software life cycle. Why software testing is very difficult?

b) Consider the program of determination of next date in a calendar. The input is a triple of day, month and year within the range 1≤month≤12, 1≤day≤31 and 1900≤year≤2005 respectively. The possible outputs would be next date or invalid input date. Design boundary values, robust and worst test cases for this program.

c) Explain with examples the difference between top down and bottom up integration testing. Why is the mixed mode integration testing preferred most of the times?

(6+6+6)

3.

a) What is software failure? How is it related with faults? Explain bath tub curve nature of the software reliability and compare it with the hardware reliability.

b) Discuss the defect amplification model of calculating latent errors in the software. Why it is preferred to identify and remove most of the errors during initial phases of software development?

c) How do you calculate error index by making classification of various errors/causes into different categories? Discuss applicability of the Pareto’s Principle in this case.

(6+6+6)

4.

a) Draw Control Flow Graph and calculate Cylomatic Complexity of the following program. Find various independent paths present in the following?

```
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;
                        }
}
```

b) Discuss the importance of path testing during white box or structural testing. What is code walk through? What do you mean by code not reachable? How do you find out not reachable code in a program?

(9+9)

5.

a) How do we measure function points of software? Compute the function point value for a software project with the following details.

Number of User Inputs=10
Number of User Outputs=20
Number of Enquiries=10
Number of Files=6
Number of External Interfaces=3

Assume the multipliers at their average values (4, 5, 4, 10, 7) and all the complexity adjustment factors at their moderate to average values (2.5).

b) What do you understand by static analysis of a program? How is the information generated during static analysis is useful?

c) Discuss test coverage metrics of procedural software. Can these be also used for object oriented program?

(6+6+6)

6.

a) How do you define software quality? Give a list of various software quality criteria and attributes.

b) "Software Quality Assurance (SQA) is an umbrella activity." Illustrate this statement and give various activities required for software quality assurance. Discuss the importance of software configuration management in modern quality paradigms.

c) What do you understand by the reliability of software? Give some metrics which can be determined from the analysis of source code of software and can be correlated with the reliability.

(6+6+6)

7.

a) List various software project management activities and give various options to achieve reliable cost and schedule estimates.

b) What are software reliability growth models? Using logarithmic Poisson execution time model calculate current failure intensity for a software which has initial failure intensity as 20 failures/hour. The failure intensity decay parameter is 0.02/failures and it has experienced 100 failures up to this point.

c) Give the structure of a software testing tool and discuss its various components in brief.

(6+6+6)