UNIVERSITY OF LIMERICK

COLLEGE OF INFORMATICS AND ELECTRONICS

Department of Computer Science and Information Systems

End-of-Semester Assessment Paper

Academic Year:	**2004/05**	Semester:	**Semester 1**
Module Title:	**Software Quality**	Module Code:	**CS4157**
Duration of Exam:	**2 $\frac{1}{2}$ Hours**	Percent of Total Marks:	**100**
Lecturer:	**Michael English and John Dawson**	Paper marked out of:	**100**

Instructions to Candidates:

- Section A is compulsory. Section B is compulsory. Attempt two questions from Section C.

- Use separate answer book for section A.

- Note that the examiner can take into account the quality of presentation and exposition as well as the content.

- Calculators are not allowed.

SECTION A

Answer *all three* questions
Question 1 carries 5 marks
Question 2 carries 10 marks
Question 3 carries 10 marks

Q1 Answer *either* a) *or* b).

(a) Total Quality Management uses the concepts of:

- Fitness to Standard
- Fitness to Use
- Fitness of Cost
- Fitness to Latent Requirements

Briefly describe all four, in a software quality context. (5 marks)

(b) The Capability Maturity Model (CMM) is a well-known Software Process Improvement model, characterised by a five-level view of process maturity.

Identify and briefly describe the *first two* of these five levels. (4 marks)

The CMM has sometimes been criticised; briefly describe any one criticism. (1 mark)

Q2 Answer *either* a) *or* b).

(a) It is claimed that iteration can be a useful approach to software development; identify its key features and briefly describe reasons for this claim. (10 marks)

(b) Describe how the SPICE model for Software Process Improvement uses the concepts of *Process*, *Process Category* and *Process Attribute* to assess software process. (10 marks)

Q3

(a) Briefly, distinguish between software product quality and software process quality and say how they are related. (1 mark)

(b) Suggestions for improving the results of software development projects include:

- Software development models
- Project management practices
- Software process improvement initiatives.

With reference to any *one* of these three, describe your choice and how it might improve the results of software development projects. (9 marks)

SECTION B

Q4 is compulsory (31 marks)

Q3

(a) The International Standard, ISO 9126, describes software quality characteristics under six broad headings. List and define these characteristics. (8 marks)

(b) The literature contains a number of reports which study the impact of software testing. Discuss one such study. (7 marks)

(c) Given the tables provided in Appendix B which are used to calculcate the minimum number of test cases needed to achieve statement, branch or structured testing, answer each of the following:

- Why is $\mu(WHILE) = 1$ for both statement and branch testing?
- For the sequencing function explain why:
  $\mu(F_1, \ldots, F_n) = max(\mu(F_1), \ldots \mu(F_n))$ for branch testing.
- Why is $\mu(WHILE(F)) = 1$ for statement testing?

(8 marks)

(d) List and explain the four different types of software maintenance. (8 marks)

SECTION C

Attempt 2 Questions from this section. All questions carry 22 marks

Q5

(a) What characteristics do a good test case and a successful test possess? (4 marks)

(b) For the testing technique Linear Code Sequence and Jump, (LCSAJ), explain the terms start point, end point and target point. (3 marks)

(c) List the steps involved in applying the LCSAJ technique to a piece of code. (4 marks)

(d) Determine all the LCSAJs for the following piece of pseudo-code. To do this create a table and for each start point determine all end points and target points. For any five LCSAJs determine appropriate test cases. Remember in this case that a test case is a sequence of salaries terminated by -1. (11 marks)

```
START
Read Salary
Tax=0
While Salary does not equal -1
   If Salary <= 15000 Then
      Tax=0
   EndIf
   If 15000<= Salary <= 30000 Then
      Tax=Salary * 0.20
   EndIf
   If Salary > 30000
      Tax=Salary * 0.35
   EndIf
   Print Tax
   Read Salary
End While
END
```

Q6

(a) Chidamber and Kememer define six metrics for Object-Oriented Design. Discuss each of these metrics paying particular attention to their usefulness in evaluating object-oriented systems. (9 marks)

(b) Explain the concepts of flowgraphs, prime flowgraphs, sequencing and nesting of flowgraphs and decomposition trees. (10 marks)

(c) How can recursive definitions of functions be used to define a set of software metrics to measure testability based on these concepts. (3 marks)

Q7

(a) What is the objective of the third phase of the Sugiyama algorithm? (2 marks)

(b) Apply the first and second phase of the Sugiyama Algorithm to the following directed graph with nodes labelled $a, b, c, d, m, n, p, q, r, x$ and edges:
$\langle a, b\rangle, \langle x, c\rangle, \langle p, a\rangle, \langle r, m\rangle, \langle m, q\rangle, \langle n, x\rangle, \langle d, m\rangle, \langle x, q\rangle$
$\langle x, b\rangle, \langle r, a\rangle, \langle p, x\rangle, \langle d, a\rangle, \langle n, a\rangle, \langle m, c\rangle, \langle r, c\rangle, \langle n, b\rangle$.
Note: In the second phase apply 1 down-pass and 1 up-pass only. (14 marks)

(c) Girard and Koschke have used graph theory as a basis for identifying components in legacy systems. Discuss this statement mnetioning at least 3 core ideas from this paper. (6 marks)

Appendix A

Test Strategy	$\mu(P_1)$	$\mu(IFT)$	$\mu(IFTE)$	$\mu(WHILE)$
Structured Testing	1	2	2	2
Branch Testing	1	2	2	1
Statement Testing	1	1	2	1

Table 1: **Metric values for primes**

Test Strategy	$\mu(F_1, \ldots, F_n)$
Structured Testing	$\Sigma_{i=1}^{n} \mu(F_i) - n + 1$
Branch Testing	$max(\mu(F_1), \ldots, \mu(F_n))$
Statement Testing	$max(\mu(F_1), \ldots, \mu(F_n))$

Table 2: **Sequencing Function**

Test Strategy	$\mu(IFTE(F_1, F_2)$	$\mu(IFT(F))$	$\mu(WHILE(F))$
Structured Testing	$\mu(F_1) + \mu(F_2)$	$\mu(F) + 1$	$\mu(F) + 1$
Branch Testing	$\mu(F_1) + \mu(F_2)$	$\mu(F) + 1$	1
Statement Testing	$\mu(F_1) + \mu(F_2)$	$\mu(F)$	1

Table 3: **Nesting Function**