

UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

BEng Honours Degree in Computing Part III
BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
BEng Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degree in Mathematics and Computer Science Part III
for Internal Students of the Imperial College of Science, Technology and Medicine

*This paper is also taken for the relevant examinations for the
Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the
Associateship of the Royal College of Science*

PAPER C302=I3.8

SOFTWARE ENGINEERING - METHODS

Friday 12 May 2000, 14:30
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions

1.

- a. A software project is sometimes divided into four stages.

Inception
Elaboration
Construction
Transition

Explain briefly what happens during each stage.

- b. Frequently it is argued that risks in software projects are reduced by means of incremental and iterative design. Explain briefly what you understand by incremental and iterative design.
- c. Explain, with a suitable diagram, how risk changes through the four stages of a software project defined in your answer to part a, for both incremental and iterative design and the waterfall model.
- d. Explain, with the aid of a suitable diagram, what is meant by *design breakage*. Illustrate your answer with reference to both the waterfall model and iterative and incremental design.
- e. How relevant is iterative and incremental design to large scale software project management?

The five parts of the question carry respectively, 20%, 15%, 15%, 20% and 30% of the marks.

2.

- a. Describe a possible format for a *quality profile*.
- b. Briefly outline (e.g. in 3 or 4 paragraphs) how such a quality profile would be used during system development.
- c. Provide a partial quality profile for a word processing program (such as Microsoft Word). This partial profile should cover 7 features and 3 attributes.

The three parts of the question carry, respectively, 30%, 35%, 35% of the marks.

3.

- a Consider a simple language with: assignments, A, binary conditionals, C(____), loops, L(____), and binary sequencing, S(____).

For this language, define hierarchically a measure which gives the number of simple paths through a piece of code. (Recall that a simple path is one which does not traverse the same edge twice. The measure must count all such paths.)

Use this measure to calculate (showing working) the number of simple paths for the following program fragment:

```
x := in
;
if (x > 0)
then
    y := x
else
    y := 0-x
endif
;
while(y > 0)
do
    x := x + y
enddo
;
out := x
```

- b Describe what paths are required to be covered by the *all du paths* strategy.

Find these paths for the above program fragment.

What probable error in the program does this analysis expose?

- c Contrast the *all du* path strategy with the *all paths* strategy.

What are the strength and weaknesses of each?

The three parts of the question carry, respectively, 30%, 50% and 20% of the marks.

4

- a Briefly describe, with examples, the most important aspects of the five kinds of "scale" used in measurement.

State the *principle of decomposition* as applied to measures and explain how it can be applied to the measurement of software.

- b What characterises *adaptive techniques* for software unit testing? What advantages are claimed for such techniques and what problems remain with them?

Use the path prefix approach to find a test suite for 100% branch coverage of the following program fragment. (Consider the tuple (x,y) as input and output. Begin with input (0,0).)

```
1.  if x > 0
    then
2.      while x > 0
        do
3.          x := x+1
4.          y := y+1
        enddo
    else
5.      while x < 0
        do
6.          x := x-1
7.          y := y+1
        enddo
    endif
    ;
8.  x := y
```

- c Let P_n be the number of predicate nodes in a piece of code. The number of tests required by the path prefix approach for branch coverage is at most P_n+1 . The "McCabe" strategy of "linearly independent path" coverage also requires this number of tests. Explain the difference between these two coverage strategies.

The three parts of the question carry, respectively, 25%, 50% and 25% of the marks.