

**THE BCS PROFESSIONAL EXAMINATION
Professional Graduate Diploma**

April 2001

EXAMINERS' REPORT

Software Engineering

There seemed to be a general improvement in the quality of answers submitted. In particular, handwriting, spelling and grammatical errors have been reduced and this is most welcomed. However, many candidates still fail to answer the questions and proceed to let us know everything they know about a related topic or a few keywords within the question. Secondly, there appeared to be more content in the answers submitted although there is still some evidence of answers set out as bullet points, and lecture notes format.

The good answers addressed the question asked. Poor answers failed to address the question as set. There was a sense of continuity, intellectual engagement, discussion or debate, a strong theme, thread or focus. Poor answers failed to address the question as set. Two or three key words were identified and descriptive and voluminous and sometimes irrelevant material produced. Some of the answers were almost impossible to read and many more were badly structured and hard to follow.

QUESTION ONE

"The most important aspect of software development is conformance to an appropriate process model." Referring to a Process Model of your choice ATTACK or DEFEND this statement. (25 marks)

The purpose of this question was to test the candidates' knowledge about the importance of a process model within software development. Candidates needed to provide an example process as the basis of their arguments. It was important that candidates covered all aspects of a software development process.

Answer Pointers

- a) Any process model was acceptable as the basis of an answer
- b) Candidates should have covered a range of topics when defending or attacking the claim such as:
 1. Good technical team
 2. Use of the right tools
 3. Use of the right method of computation
 4. Clear and stable set of requirements
 5. Good configuration management and change control
 6. Good project management (technical and personnel)
 7. Good planning

QUESTION TWO

The software development team you manage has been using structured programming and C for the last 20 years to develop real-time applications. A decision has been taken to move to an object-oriented development method and an object-oriented language. Write a report to senior management explaining the technical, managerial and resource problems of making such a dramatic change and providing the management with some viable approaches. You must include in your report recommendations for a suitable object-oriented development method and a suitable language. (25 marks)

The purpose of this question was to test candidates' knowledge of the implications of the change. In general, candidates did not cover enough of the issues and tended to spend too much time on issues with which they were familiar rather than covering other aspects of the problem.

Answer Pointers

a) Technical issues should have included; O-O is a completely different concept from structured programming and C, all technical staff will require education and training plus appropriate time to become fully conversant with the new development method and language.

b) The Management problem is twofold. Firstly that all management need to understand the implications of using the new technology plus the change in culture within the organisation, secondly they will have to manage the current system, plus the time lost through all the training and education plus the change over to the new system. Project timescales for early projects will have to be adjusted to allow for familiarisation.

c) Resourcing issues should have included cost of training, loss of technical and managerial resource during training, cost of additional consultancy, cost of software (and hardware) needed for technical staff to become competent.

An obvious approach is to move one step at a time; initially make people aware of O-O technology, train them and educate them, then either change the development method and keep the language or change the language and keep the development method. This latter solution will allow staff to be trained in groups whilst maintaining the development method. This would be viable for either Java or C++.

QUESTION THREE

a) Using appropriate examples, compare and contrast a formal development method with a structured development method. (14 marks)

c) What are the difficulties in trying to combine these two approaches? (11 marks)

The purpose of this question was to test the candidates' understanding of the benefits and weakness of each of the methods. This question was answered really well by a number of candidates although many candidates who tackled it gave poor answers.

Answer Pointers

- a) Candidates should have compared all the attributes of both approaches including formal mathematics versus more readable diagrams, rigorous design, formal proof, tool availability, understandability, availability of experts.
- b) The advantages and disadvantages of each method should have been discussed.
- c) The problems discussed should have included integrating formal mathematics with diagrammatic techniques and trying to gain benefit from the advantages of each technique.

QUESTION FOUR

Models for data, architecture, interface, and components are said to be the most fundamental outputs of the design stage in any software engineering project.

- a) **Briefly explain the purpose of EACH design model in the complete specification of a software design. (9 marks)**
- b) **Illustrate, using diagrams and appropriate explanations, the models produced by a typical commercial structured design method. (10 marks)**
- c) **Briefly discuss whether the object-oriented approach has made an impact on the fundamental outputs of the design phase. (6 marks)**

This question assesses the candidate's appreciation of the different facets of a design specification resulting from the design phase of a software engineering project. A variety of design methodologies and techniques were presented. Whilst many candidates were able to describe the different models within a design method, very few were able to illustrate their answers with example applications.

Answer Pointers

- a) Data models specify the data and data structures required; architecture models define relationships between sub-systems; interface models address the communication requirements of the system; and component models describe the process or procedural aspects of software
- b) Application of appropriate diagrams from a commercial method such as Yourdon (or SSADM) for: data design – dataflow and entity-relationship diagrams; architecture design – context diagram, structure diagram, control flow diagrams; interface design – interaction diagrams; component design – structure chart, process specs and STDs
- c) Focus on the issue of identifying objects such that components and data design are integral to the object. The architecture design is captured by object hierarchies; with interface design based on the exchange of messages between objects

QUESTION FIVE

As a Quality Assurance and Metrics consultant you have been asked by a software development company to advise on establishing and operating an in-house software quality model. Write a report that:

a) presents an overview of quantitative and qualitative models of software quality; (10 marks)

and

b) discusses the problems, difficulties, and potential pitfalls in applying these models. (15 marks)

This question assesses the candidate's knowledge and awareness of quality assurance models and their effectiveness. Many of the answers to this question lacked one or more of the following: a structured report format and style of presentation; ability to differentiate between the terms qualitative and quantitative metrics; a balanced discussion of the techniques and tools available in both cases. Thus, many answers focussed only on qualitative metrics and quality assurance procedures.

Answer Pointers

- a) A properly structured report, the contents of which include; background information on software quality and explanation of the key differences between qualitative and quantitative models. The report should highlight such things as: the "software crisis", mathematical and empirical models of quality for greater engineering precision and predictability and formal procedures for managing the development process.
- b) Discussion should focus on issues such as the problems in gathering empirical data for software metrics, the relative immaturity of models, the general lack of practical tools for commercial software projects and the varying educational background and software engineering experience of many developers.

Finally, advise on adopting qualitative models based on FTRs supported by proven quantitative measures for such things as designing individual components.