THE BCS PROFESSIONAL EXAMINATION
Professional Graduate Diploma

April 2002

EXAMINERS' REPORT

Software Engineering

The general readability of student scripts have improved and presented fewer difficulties this year. However, spelling, grammar, and the organisation of answers in some centres remain problematic. Further improvements could be made in producing answers that address the question as set, and is structured in a manner that gives a clear focus and a strong sense of continuity, intellectual engagement, discussion, or debate. Some candidates simply set their answers out as a set of bullet points, mere descriptions of products or processes; or the reproduction of a collection of lecture notes or memorised textbook materials.

Question 1

Discuss the key principles of software engineering for the development of software systems that, when applied, can lead to solutions that are modifiable, efficient, reliable, and understandable.

(25 marks)

Answer Pointers

This question assesses the candidate's awareness of the fundamental principles underlying software engineering that can give rise to the quality characteristics of modifiability, efficiency, reliability, and understandability. A good answer will:

a) identify the principles of **abstraction** with information hiding, **modularity** with localisation, and **uniformity** with completeness, and confirmability, as key principles in software development.

b) provide clear definitions and illustration of, and differentiation between, each principle. For example, **abstraction** is the ability to extract essential properties and model from different perspectives whilst information hiding is the concealment of that which is not essential to the use or description of that which is being modelled; **modularity** enables the management of structural complexity by decomposition into networks of smaller, more easily understood units. Thus, localisation in this context will mean that side effects from a unit's behaviour is confined to a well-defined area. Finally, **uniformity** is about consistent notation, structures, and program flows with common patterns such as sequence, iteration, and selection structures permitting completeness and confirmability where essential components can be identified and are testable.

c) provide clearly reasoned arguments about the application and limitations of these principles in the software industry. Thus, it could be argued that *"In the software* industry today, the realisation of the objectives is very dependent on the degree of interaction between developer and user. The less involved the user, the more likely it is that the system will lacking in one or more areas. For example, whilst the developer may be able to demonstrate consistency, it may be that others are

more able to judge completeness. Abstraction needs to remain a medium by which user and developer can share a common view of the intended system, and modularity the mechanism for a common understanding of its component behaviour."

Examiner's Comments

This was a popular question alongside Q2, but was poorly answered, and many failed. Whilst the intention of this question was to focus on programming concepts, many students' interpretation were that it was concerned with the software development life cycle model. Thus, the answers provided in many instances were extensions to Q2.

There are a number of issues that have been raised as a result. Firstly, there was no shared understanding of the term "key principles" across centres, thus a wide variety of ideas were presented by students. Secondly, many of the answers seemed to indicate that candidates were less able to discuss technical issues to do with program development, and more apt to present management generalisation of the software lifecycle process. Finally, many candidates found it difficult to deconstruct the question and, identify and discuss the key issues. Thus, many of the answers were mere reproduction of memorised lecture material.

As a result of these observations, marks were also awarded for plausible answers that identified and discussed key principles within the context of the software lifecycle. But, very few were even able to benefit from this concession.

Question 2

2. *a)* Describe the SPIRAL software development process and comment critically on its suitability for a modern software development project. (15 marks)

b) **Your Managing Director has asked you to recommend a life cycle model that he should implement for a RAD (Rapid Application Development) project recently contracted to your firm. Sketch the plan of a life cycle model you would choose, and identify three or four suitable milestones that would measure progress. Make sure you include justifications for the recommendations you plan to make. (10 marks)**

Answer Pointers

A very popular question.

Key points of a good answer to (a) were the appreciation of risk, the possibility of stopping a project that exhibited too much risk, and knowledge of the phases (determine objectives, identify and assess risk, develop and verify next-level product, and plan next phase) plus anywhere-near decent grasp of the phase specific processes in each step of the Spiral Model.

A good answer to (b) depended on giving reasons for the life cycle model that was chosen.

Examiner's Comments

Most students missed the utility of stopping a project of risk proved too great.

For the element about suitability, very few students related the Spiral Model to its home territory, that of developing new products using innovative or untested technology. A reasoned argument about 'modern suitability' was acceptable providing rationale was clear. One of the best answers commented that risk was encapsulated and mitigated within each product release of the DSDM method. This neatly described the shift of effort from combating high-tech risks of the 80s to combating business-tech risks of the new Millennium.

Few candidates were able to justify their selection using the criteria of RAD – use of timeboxing, use of high productivity tools coupled with frequent iterations that produced a product each time, concentration on business issues rather than engineering issues, and maintaining management visibility.

Question 3

A vending machine dispenses a wide range of products. The temperature of the machine is monitored on a regular basis and maintained between 15 and 21 degrees Celsius. If the temperature is outside of this range the machine will shut down.

A product will only be dispensed if the customer has tendered at least the minimum charge, selected an available product, and then pressed the "dispense" button. The machine will then return any change, but no greater than what was owed to the customer. Finally, the customer can cancel an order at any time by pressing the'cancel' button and the money tendered will be returned.

Using an appropriate development method demonstrate how you would model the process, data, and the timing control aspects of the vending machine outlined.

[Note. Marks will be awarded for clear and relevant sets of diagrams with supporting annotation and descriptions.]

(25 marks)

Answer Pointers

This question assesses the candidate's theoretical understanding and practical awareness of software development methods. A good answer should demonstrate knowledge of methods and notation for the representation of process, data, and timing. Further, different diagramming models should be applied to the specific aspects of the vending machine. For example*, using SSADM*

- Develop process models (Data Flow Diagrams)context diagram and Level 1 DFD showing typical processes such as: dispense, monitor/maintain, select, cancel; and flows – temperature, product selected, amount tendered;
- Develop logical data models (Entity Relationship Diagrams) where the entities include: machine, product, temperature, customer, order, charge, money; and relationship such as one-to-many (customer->order)
- Develop Entity Life History models. For example, an order made by a customer which is processed, dispensed, and closed. Another could be the machine

whose temperature is self-monitored, and shuts down when the temperature is out of range.

- Diagram(s) and annotations that represent the complete system either from a time, data, or process-based perspective using the components identified earlier. For example, the dispense process should have flows inwards of the product selected, minimum tendered (flag), and the dispense signal; and a flow outwards entitled dispensed (flag perhaps). Likewise, the 'product selected' flow would be the outflow from the select process.

Examiner's Comments

This was the least popular question on the paper, but many more candidates attempted it, and provided reasonably good answers compared to previous years. This is encouraging and most welcome considering that this type of question is one of the most challenging, as it requires the student to demonstrate analytical, modelling, and problem-solving skills under closed-book examination conditions.

Question 4

A set of qualities for a software product has been described by the following set of discrete, separate 'abilities':
functionality, reliability, usability, efficiency, maintainability and portability.

Briefly define any FIVE of these with short descriptions that clearly identify what is meant by each and how they differ one from another.

(25 marks)

Examiner's Comments

A very popular question.

Not enough thinking went into some of these answers. Too many adopted a process mind-set, and described HOW to achieve, for example, functionality (extensive requirements gathering) , or efficiency (lots of testing). Good answers described attributes of, for example, WHAT were usability (easy to learn, easy to operate, easy to understand) or WHAT attributes of maintainability (easy to study, easy to test, easy to add new bits).

Question 5

a) ISO 9001 has many paragraphs that describe compliance to quality. Compliance with ISO 9001 requires compliance with all the systems described in the standard. This form of quality compliance has been described as 'top down'. On the other hand, a method of complying with one system or subsystem at a time, as a way of obtaining gradual improvement, has been described as 'bottom-up'.

Comment on each of these descriptions. Give reasons for your answers. (15 marks)

b) **You are asked to set up a training session for some staff because your manager is concerned that there is insufficient awareness of quality procedures in your firm. Describe, with reasons, four or five topics you would set for such a training programme and identify, with reasons, the particular sequence in which you would introduce them. (10 marks)**

BCS Software Engineering – Professional Graduate Diploma - Additional Information for Q5

Systemic Requirements
Establish your quality system
Document your quality system

Management Requirements
Support quality
Satisfy your customers
Establish a quality policy
Carry out quality planning
Control your quality system
Perform management reviews

Resource Requirements
Provide quality resources
Provide quality personnel
Provide quality infrastructure
Provide quality environment

Realisation Requirements
Control realisation planning
Control customer processes
Control product development
Control purchasing function
Control operational activities
Control monitoring devices

Remedial Requirements
Perform remedial processes
Monitor and measure quality
Control non-conforming products
Analyse quality information
Make quality improvements

Examiner's Comments

(a) Few answers, even fewer sensible answers. Many answers became word-blind, and gave a description of top-down testing versus bottom-up testing. Very few answers found merit in the idea that improving quality a little at a time was better than testing quality on 20 fronts all at once. Whenever an answer was supported by good argument and examples, it gained high marks.

(b) The question was asking candidates to select topics from the supporting material provided and justify their selection and sequencing. Very few did so. Most preferred to offer their own views of what to deliver. Where these views were supported by reasoned arguments about relevance, they were admitted and awarded high marks. Too many candidates simply itemised what they would teach about quality, and gave no reasons for its selection or sequencing.