THE BCS PROFESSIONAL EXAMINATIONS
Professional Graduate Diploma

April 2006

EXAMINERS' REPORT

Software Engineering (now renamed Software Engineering 2)

General Comments

Most of the scripts produced by candidates this year were well structured and readable. However, in some centres the general standard of the written work, and the level of comprehension (of the question set as well as the subject matter), needs particular attention.

Above all, this is a technical paper about Software Engineering. Questions seek to test candidates' knowledge and ability to apply that knowledge. It is a poor answer that simply describes recalled information. It is a good answer that can then show application of the recalled knowledge.

Question 4 and 5 proved to be two of the three most popular choices amongst candidates, but exhibited the lowest performance average overall. Both questions are characterised by four sub-sections appropriately weighted. In terms of time management, many candidates appeared to use the allotted time to answer only one or two parts of the question. In a few instances, the age-old problem persists in which some candidates just simply attempt to recognise one or two keywords without regard for the context and scenario in which the question was presented. On reflection, given the diversity of the cohort across centres, multiple-section questions present difficulties to many candidates, and perhaps should be limited to a maximum of two or three sub-sections.

In some centres the currency of knowledge in terms of software engineering practice, analysis and development is of concern. Again, we must emphasise the importance of candidates studying not only a recommended text, but reading more widely in terms of publications within the profession.

Similar to the previous year, a significant minority of students have answered more questions than was required in the examination, and this practice reduced the time they have for each question and limited their opportunity to obtain the full range of marks for the questions attempted.

Finally, similar to the previous year, some student had not completed the front cover with the number of the question and part-questions attempted. This made it very difficult to find, then determine whether it was a continuation of a previous question, and assess and record the mark for that question as a whole.

Question 1

1. *a)* Briefly describe TWO testing strategies for a software module that takes as input, and acts on, a set of six elements of data. The internal workings of the module are not available for testing. A detailed specification of the module's functionality is available, and this declares that all input data are validated. The ranges of each element of data are known. Some data elements, in a group, act together to describe a functional scenario. **(12 marks)**

b) Design a test strategy for a function with the following definition:

The function will process a scientific computation. Input data will be a set of parameters between 4 and 10 numbers, where each number will have 5 digits and be greater than 10,000. Input data validation will ensure the inputs are correct before computation begins. **(13 marks)**

Answer Pointers

a) Two possible strategies address the ranges of the data elements, and the way that groups of them act together.

Range: a strategy would be to probe the edges of each range to ensure the ranges are properly validated: 2 marks
this implies two pairs of test values, one pair at the lower edge just outside and just inside; 2 marks
and one pair on either side of the upper edge. 2 marks

Data inputs acting together as groups: the specification of the module should be studied to determine a logical description of the set of valid input groups. By complement, an invalid group will also be defined. 3 marks
The test data is then constructed with one test corresponding to each logical group, and a test for an invalid logical group. 3 marks
Alternative strategies that deal with ranges and groups are acceptable.

b) The input parameters vary between groups of 4 to groups of 10, suggesting there exists logical groups (1) invalid group less than 4, (2) a valid group of between 4 and 10, and (3) an invalid group of greater than 10. 6 marks

The actual values themselves suggest a Boundary Value approach to range; an invalid sample less than 10,000, the minimum valid number (i.e. 10,000), the maximum valid number (i.e. 99,999) and an invalid number out of range at the top, i.e. greater than 99,999. 7 marks

Examiner's Comments

The framing of this question sought to find higher skills of testing among candidates. These skills were partly knowledge of methods of testing and partly the ability to apply that knowledge.

However, many candidates described methods without displaying ability to use them. Some answers lacked discipline to cover all the situations given in the question.

Question 2

2. *a)* "Software tools have moved the development of software from a profession of craft to a profession of product engineering."

Discuss this statement, giving TWO arguments in favour and TWO arguments against, and make an overall conclusion based on the arguments you provide. **(12 marks)**

b) Describe THREE functions, characteristics or functionalities of a software tool that supports source code control, and includes the sharing of code development among several programmers. **(13 marks)**

Answer Pointers

a) Arguments in favour: two from standard notation for ease of reasoning and understanding, complexity management from computer-based document and version control, automatic production of code and data (tests) from requirements and design specifications, and automatic report generation for QA purposes. Or similar, given as contributions to an engineering ethos or demonstrable correctness. 6 marks

Arguments against: two from; tools provide compliance with method, not application of theory; diagrammatic notations are held to have imprecise semantic meaning; test specifications cannot cover all possible scenarios no matter how automatically they are generated; and requirements traceability is always a matter of interpretation because of the remodelling that is done at each production stage of a software development lifecycle. Or similar, given as contributions to lack of completeness, lack of repeatability and lack of rigorous development. 6 marks

b) Select three from:
Access control, with check-in and check-out locks
Updating changes in the project database
Capability of generating baselined configurations at specific version stages
Generation of reports (when modified, how modified, how configured)
Or similar, emphasising database, locks, versions and reports.
4 marks each, plus 1 if all three well described with sufficient lack of overlap. 13 marks

Examiner's Comments

Part (a) of this question was designed to test candidates' knowledge of the evolution of software engineering. The expected emphasis of answers was on how tools have affected the practice of software engineering. Part (b) asked for specifics of tool functionality and expected features rather than benefits.

However, some answers were about the business benefits of using tools, and others were self-contradicting by claiming both that tools were beneficial because of an enforced discipline and also that an enforced discipline inhibited programmer creativity.

Question 3

3. *a)* What THREE aspects of risk can be identified during software product development? **(12 marks)**

b) The formula for risk exposure is the sum of the products of each impact multiplied by its probability, for each risk. Calculate the risk exposure for the following data about a software development project. Give full reasons for each step of your calculation.

- If the project is late, it will cost £10,000 per week. Recently, 7 out of our last 10 projects were late, with an average delay of 4 weeks.
- If the integration phase is smooth, it will save time. Integration is currently planned to take two weeks, at £5,000 per week. Previously, 5 out of our last 10 projects took longer than planned, the worst being 6 weeks longer because of the amount of redesign and rebuild we had to do.
- If the client has stable finances, we will get paid on time. From our records, 3 out of our last 10 clients delayed payment for an average of 6 weeks because they had financial difficulties. Each month we run an overdraft on this kind of project and non-payment costs us £1000 per month in charges. **(13 marks)**

Answer Pointers

a) Expecting a reasoned answer to identify three concepts or issues with an appropriate rationale for existence:

For example, to do with management of the work-

- Project risks e.g. staff changes or hardware changes
- Product risks e.g. reqs change or new tools (CASE) to learn
- Business risks e.g. product competition.

Another acceptable answer might identify aspects to do with the risk management programme and how it might be applied, again with appropriate rationale. For example -

- Avoidance – can we replace an unreliable build with a reliable purchase?
- Minimisation or mitigation strategies – overlap assignments to reduce damage if we lose staff.
- Contingency plans – what to do if, e.g. we have a fire.

An appropriate answer will fit the context of the work of a software project, and be appropriately explained and exemplified. 12 marks

b) If the project is late, it will cost £10,000 per week. 7 out of our last 10 projects were late **(implies probability of 7/10**), with an average delay of 4 weeks **(implies total cost of delay is 4 times £10K**).
Probability = 0.7, impact = £40,000 3 marks

- If the integration phase is smooth, it will save time. Integration is currently planned to take two weeks, at £5,000 per week. 5 out of our last 10 projects took longer than planned (**implies probability is 5/10**), the worst being 6 weeks longer because of the amount of redesign and rebuild we had to do (**implies average delay of 3 weeks if worst was 6 weeks, at £5K per week**).

Probability = 0.5, impact = (0.5*3) * £5000 4 marks

- If the client has stable finances, we will get paid on time. 3 out of our last 10 clients (**probability 3/10**) delayed payment for an average of 6 weeks because they had financial difficulties. Each month we run an overdraft on this kind of project because of non-payment costs us £1000 in charges (**implies £250 per week for 6 weeks**).

Probability = 0.3 Impact = 6 * £250 3 marks

Risk exposure = (0.7*£40,000) + (0.5 * £7500) + (0.3*£1500)
= £32,200. 3 marks

Method gains marks, arithmetic errors lose few marks.

Examiner's Comments

This question was one of the three most popular. The answer required attention to detail and discipline of approach. Those candidates with detail and discipline gained high marks.

However, many answers were insufficiently distinctive. For example, staff changes, hardware changes and lifecycle imprecision, while all risk factors, are all impacting on one area, that of project execution.

Question 4

4. Software development, as a relatively new engineering discipline, is beginning to exhibit underlying principles that might be considered to be both enduring and universal.

Present a brief discussion of each of the following principles. You should make clear any distinctions between the terms used, and evaluate the universality of the principle itself.

i) Minimise coupling between, and maximise cohesion of, components; **(6 marks)**
ii) Use formal methods to both define software artefacts rigorously and manage quality throughout the development life cycle; **(8 marks)**
iii) Build systems for reuse and with reusable components; **(6 marks)**
iv) Control system complexity and implement disciplined and flexible process cycles. **(5 marks)**

Answer Pointers

A good answer should present a brief discussion make the following key points:

(i) Coupling is the interdependence of one module to another;
Cohesion is strength of the association of elements within a module;
The minimisation of coupling reduces the possibility of side effects;
The maximisation of cohesion results in simple, single, provable function generation;
As a principle, it is arguably universal. 6 marks

(ii) Formal methods are mathematically based techniques for the specification, development and verification of software and hardware systems;
An opportunity in the profession for rigour, accuracy, and thus quality;
It can be argued to have limited universality as no consensus on the meaning of formality and existing metrics. 8 marks

(iii) Build for reuse requires conformity to standards in interface design, documentation and storage;
Enables new systems to be generated easily from these and other proven components, increasing productivity and improving quality.
The principle is widely accepted, but there remains wide variation in terms of the creation, documentation, and storage of components. 6 marks

(iv) Developing solutions of quality for complex problems is generally tackled by the application of life cycle methods and decomposition techniques (e.g. functional) characterised by successive division of the problem where the meaning of the sub problems are made explicit, and the stages of development are clearly focussed;
The principle is widely accepted, but the implementation exhibit wide variation and is often governed by the nature and context of the problem itself. 5 marks

Examiner's Comments

This question was the most popular amongst candidates with some very good answers framed within the context of the scenario given. That is, not just simply identifying the software engineering principle, but also assessing its universal applicability.

However, there were many answers that focussed only on the principles, ignored the sub-section relating to formal methods, or described specific process models at length and in detail. These answers were often very descriptive in nature.

Question 5

5. A telephone company supplies its domestic customers with communications equipment and services. Subscription customers are charged a monthly fee of £40 for two years. These customers receive the equipment for the service at no extra charge and have no limits imposed on the amount of the service consumed. Other customers are charged for the equipment and billed £2 per unit of service consumed during peak periods (9am to 6pm), and £1 per unit at other times.

A Customer Billing System is required. From the system description given, answer the following questions.

a) Produce a list that gives the names of each possible entity, their attributes and any required processes; **(8 marks)**

b) Use an appropriate modelling technique (state any assumptions made) with supporting notation to demonstrate:

i) the behavioural aspects of the system; **(7 marks)**
ii) the structure and relationships within the system **(6 marks)**

c) Briefly discuss how any future changes to the fee and billing structure might impact on the current design and implementation model. **(4 marks)**

Answer Pointers

A good answer should cover the following areas:

a) Candidate Entities
Telephone Company
Customer
Communications Equipment
Services

Attributes (examples)
Customer:: Subscriber Type
Contract Start and End Dates
Equipment:
PeakUnits:
OffPeak Units:
Usage Limit:

Processes (Examples)::
ComputeBill (subscriberType);
getPeakUnits();
getOffPeakUnits(); 8 marks

b) By using appropriate modelling techniques with supporting notation,

(i) The behavioural aspects of the system can be represented in a variety of ways. These include data flow, state transition, interaction, diagrams. The important behaviour to capture include:

The Subscriber, billed monthly; receives equipment and services free of charge; Subscription renewal at the end of two years; any data collected on usage are for information rather than charging purposes.

The non-subscriber is billed on equipment and service delivery; the service and the period during which it was used are the parameters used in the calculation. 7 marks

(ii) The structure and relationships within the system can be represented using either ERD, Use Case, or Class diagrams.

The relationships to model are:
The Telephone company to the Customer (1 to many)
The Telephone company to the Service (1 to 1)
The Telephone company to the Equipment (1 to many)

The Customer to the Service (* to 1)
The Customer to the Equipment (1 to 1)
The Customer to the bill (1 to many)

The Service to the Bill (1 to 1)
Subscriber & Non subscriber (Customer type selection) 6 marks

c) Changes to the actual rates should have no impact as these can be viewed as parameter values; Changes to the formula, the user group, or method of billing will be more significant requiring the ability to develop and integrate completely new processes with minimal side effects on the other parts of the system. 4 marks

Examiner's Comments
This question type, (problem-solving case study application of technique) was the third most popular choice of candidates compared to its general unpopularity in previous years. However, this may be due to their being many more of this type of question on the paper in comparison to previous years.

The work submitted was very good in some cases, making use of DFDs, ERDs, and OO class diagrams. There were a few candidates that appeared to have either ignored or misunderstood the scenario specification in the formulation of their answers. Other candidates focussed on the data modelling at the expense of the process modelling required in other sections of the question. In future candidates should ensure that they distribute the allotted time equally across the four aspect of design highlighted in the question. Thus, many marks were lost by only addressing one or two deliverables.