THE BCS PROFESSIONAL EXAMINATION
The Professional graduate Diploma

April 2000

EXAMINERS' REPORT

Software Engineering

General Comments

The major weaknesses were:

- Handwriting, spellings, grammar, and the organisation of answers made the task of marking very difficult in some cases.
- Some candidates simply set their answers out as a set of bullet points, or a collection of lecture notes. Therefore, they failed to produce answers with a sense of continuity, intellectual engagement, discussion, or debate, and the absence of a strong theme, thread, or focus.
- Some candidates failed to address the question as set. Instead, two or three key words were identified and everything that was known about those topics were written.

Question 1
Answer Pointers

a) Illustrate the V-model software development process and describe each of its phases and the relationship between them.

Diagram of the V-model showing; requirements/specification, high level design, low level design, implementation and unit test, module testing, subsystem testing and system testing : 2 marks.

Showing flows backwards as well as forwards : 1 mark.

Description of each phase : 7 marks (one for each phase).

Relationship between each phase. This should describe the documents that are generated : 2 marks (1/2 mark for each).

Illustration that in addition to the production of design documents and code, test specifications are required to provide the basis of the testing phases on the right hand side of the V: 3 marks

(Total 15 marks)

b) State the strengths and weaknesses of the V-model where the major requirement is to develop software to support rapidly changing business needs.

Strengths; well controlled process so that impact of changes can be analysed, high quality code can be produced, changes to the requirements, design and implementation can be well controlled : 3 marks

Weaknesses; if strictly followed it is very difficult to produce rapid solutions, takes a lot of 'non-technical' management, : 3 marks

(Total 6 marks)

c) Briefly describe and justify an alternative process model that may be better suited to the development of such software.

Possible alternatives include DSDM, V-model with rapid prototyping (although not its real use), evolutionary development and even incremental development.

Brief description of the process : 2 marks

Justification : 2 marks

(Total 4 marks)

Examiner's Guidance Notes:
The purpose of this question is to test the candidate's knowledge of a well-known process model and then to see how they can evaluate that models effectiveness in a practical situation. The candidate then has the opportunity illustrate their knowledge of another process model.

The question was answer quite well by a number of candidates but several candidates provided very weak answers that showed a lack of a basic understanding of the V-model.

Question 2
Answer Pointers

a) Briefly discuss the role of Software Quality Assurance within a software development project.

Expecting a description along the following lines:

" The role of SQA is to ensure that the software development team are following the process and procedures laid down in the software quality plan and it the software development plan. For example if the software quality plan advocates the use of technical inspections on all code, it is the job of the SQA activity to ensure that these take place.

SQA activities are normally carried of by the SQA team who are responsible for the development of the software quality plan and for scheduling and carrying out software quality audits against this plan.

They may also be responsible for ensuring that the software development process being used (and its associated tools and methods) are appropriate for developing the software to a high enough standard.

They are also responsible for ensuring that all standards (international, national, company and project) are conformed to".

(Total 5 marks)

b) Describe each of the following and explain how it contributes to the development of high quality software:

i) configuration management

ii) change control

iii) software tools

iv) standards

v) metrics

Each of the following carried 4 marks and has two parts; a description and a contribution along the lines described below is expected :

configuration management

- to control the versioning of all documents (including code) and the management of builds of the software into executing binary --- not just about version control but managing different configuration of the software and ensuring that there is traceability back to the versions of the various documents, versions of all documents (incl. code) are normally uniquely identified

- enables controls over the release of software to be maintained, ensures that a any version of any document (incl. code) can be traced back to its requirements, and enables fixes to the software to be controlled

change control

- the management of changes to any document (incl. code) within the software development, in large projects this can involve a 'change control board' who are responsible for assessing the impact of all changes, all documents are normally controlled by version numbering (see configuration management)

software tools

- ensures that any required change to the software is properly managed in order to eliminate 'new requirements' being implemented in the code before proper design work has been performed, ensures that all documents are appropriate updated in line with changes in other documents

standards

- the standards used in software development range from international standards such as those defined by ANSI (for example, programming languages) to those defined by the project (e.g. the format of a document), , such standards may define the methods, tools and techniques that must be used by the development team, the software process (e.g. v-model) they must follow, the coding standards that must conform to

- standards are used to provide the basic quality to the software development process and the documents (and code) it produces, they ensure that the development team has a set of basic principles to follow and a set of standards to conform to, they also enhance communication on a project by making documents easier to follow (because everybody understands the structure and the notation being used), the existence of many standards enable tools to be developed to support those standards thus enhancing the quality of the software and associated documentation

metrics

- metrics fall into two categories; those which are collected on a project to provide the foundation stone of estimations for future projects, e.g. LoC per day, bug fixes per day etc. and those used to control the quality of the software, e.g. complexity metrics

- the former can be used to ensure that the correct resources are allocated to a project so that tasks can be completed without the need to 'cut corners' and trade quality for time; the later can be used to provide targets to the development team, e.g. no procedure shall have a complexity of greater than 10, also these metrics can be used to aid both readability of code as well as testability of code

Software tools

Tools exist to support the analysis, design, implementation, and evaluation stages of software development. In the initial stages of a project it is imperative that errors of omission are avoided, and entities are consistently described from one phase to another. Prototyping tools allow the developer and the potential user to agree requirements in terms of the "look-and-feel" of the proposed system. Design tools ensure consistent representation, verification, and documentation of each design refinement. The programmer's workbench (version control, libraries, compilers, debuggers, test generators etc) ensures the right product is delivered on time to customers, using proven modules, with optimal execution efficiency.

(Total 20 marks)

Examiner's Guidance Notes:
The purpose of this question is to test the candidate's knowledge of SQA and the contribution that a variety of activities and tools make to the development of high quality software.

In general this question was not well answered. Many candidates did not seem to understand the basic principle of Software Quality Assurance and a number did not seem to understand the role of configuration management, change control, standards and metrics in a software development project.

Question 3
Answer Pointers

After several years the software house you work for is contemplating changing from using a structured development method and a structured programming language to using an object-oriented development method and an object-oriented programming language.

As a Software Team Leader you have been asked by your Technical Director to produce a risk assessment and an implementation plan for the proposed change.

Provide a report to your Technical Director including the following:

i) your risk assessment

Risks:

new paradigm; a different way of developing software
new development method; time spend training
new programming language; time spent training
new tools; time spent training and becoming familiar with tools, new suppliers
lack of experts in the new approaches
previous metrics of limited use
a new development process, standards and documentation will be required
may need to 'port' existing code or support two development processes
lack of experience may result in early projects; failing/overrunning/exceeding their budget
transition will divert resources away from current projects
may be difficult to gain business because of perceived lack of experience with new approach
loss of staff once they are trained up

(Total 8 marks)

ii) a plan of the activities required to reduce the risk

Plan should include:

training of staff
use of consultants and experienced subcontractors
employ experienced staff
purchase new tools (& associated training and consultancy)
obtain published metrics
develop new processes, documents and standards or re-use published ones
produce detailed risk assessments for early projects to ensure that all contingencies are covered
employ subcontractors to reduce impact of change on current project
produce a detailed 'cost-benefit' analysis for the company
evaluate the Company's perceived strengths and weaknesses in order to assess impact of change on business

(Total 12 marks)

iii) the potential benefits of change

Benefits could include:

potential for more business by being seen to be using state of the art technologies
retention of staff
better able to recruit staff
possibility to develop a good reuse strategy; thus reducing project costs and time scales
opportunity to re-vitalise staff, processes and standards

(Total 5 marks)

Your report must cover both technical and managerial issues.

Examiner's Guidance Notes:
The purpose of this question is to test the candidate's awareness of the impact of such a change on a development team and the company concerned. Candidates are expected to produce a wide range of risks depending on their experience and only some of the potential risks are outlined.

The question was not answered very well. Too few candidates seemed to understand the concept of risks and the problems involved in changing to a different technology.

Question 4
Answer Pointers

"Compared to structured design, object oriented design is well suited to managing the complexity inherent in developing large programs".

Discuss the success and failures of such methods in the design and implementation phases of software development. Illustrate your answer with examples.

This question assesses the candidate's

1) Knowledge of the strength and weaknesses of different software development methods;

2) Ability to apply that knowledge with illustrations of design and implementation scenarios.

A good answer will consist of:

- An overview of software complexity in terms of system size and the need for performance, flexibility, and functionality.

(6 Marks)

- The recognition of decomposition as an essential aspect of both systems for handling size complexity. However, this should be followed by a discussion of the difficulties faced by structured design at the design and implementation stage such as the mismatch between real world entities and their abstract representation; and the difficulties of the efficient integration of exceptional cases. These are instances where the OO approach is strongest.

(6 Marks)

- Instances of Structured Design success is in the communication of technical requirements to programmers; and the production of efficient applications in terms of their use of memory and processor resources. However, such systems are inflexible, and often fail to meet the current end-user requirements.

  Instances of OO success are in the communication of requirements between user and engineer; and the ability to accommodate changes easily. However, resulting systems at the implementation stage have a tendency to be inefficient and impractical resulting re-engineering of the implementation using structured methods.

(8 Marks)

- Brief discussion of the "reality" of current object models and the attainment of engineering quality against a background of problems associated with reuse, and inheritance.

(5 Marks)

Examiner's Guidance Notes:
This question attempted to assess the candidate's: knowledge of the strengths and

weaknesses of different software development methods; and ability to apply that knowledge with appropriate illustrations of design and implementation scenarios.

This was not a popular question, probably because it required breadth of knowledge of two quite distinct design approaches. It was reasonably well answered with most of the answers focussed on object orientation and its benefits alone. Thus, there were not many answers that gave illustrations relating to the life cycle development of software using both methods, supported by a discussion of their relative strengths and weaknesses.

Question 5

Answer Pointers

You have recently joined a large IT department as a Software Requirements Analyst. However, key members of staff have already expressed the opinion that software requirements analysis is only about building prototypes that are merely refined until they become final production software.

Write a report that presents an overview of:
i) the role of requirements analysis;
ii) the tools and techniques available to support this task.

Your report should address some of the misconceptions that your colleagues may hold regarding your role as an analyst.

This question assesses the candidate's: knowledge of requirement analysis; and appreciation of the common misconceptions about analysis and prototyping.

A good answer will consist of:

- A properly structured report whose contents include: background information on the importance of requirements analysis in software engineering as a bridge between system engineering and software design;
(5 Marks)
- A discussion of the various stages, namely: problem recognition; problem evaluation and solution synthesis; modelling; specification; and review;
(8 Marks)
- A brief discussion of tools that might be deployed to support stages of analysis;
(4 Marks)
- A discussion of the role of prototypes in requirements analysis; and the need for such prototypes to be discarded once the requirements have been established.
(8 Marks)

Examiner's Guidance Notes:

This question assessed the candidates' knowledge of requirement analysis, and their appreciation of the common misconceptions about analysis and prototyping.

This was a relatively popular question and many candidates clearly knew something about requirement analysis and prototyping. The question was reasonably well answered, however many answers were let down by a lack of an overall report structure. Not many answers attempted to engage in a discussion of the "grey area" (in terms of definition) of the requirements analysis role and personnel, and develop a common understanding by which misconceptions about it can be alleviated. Instead, the content of some answers was presented as merely a collection of lecture notes and bullet points that had been committed to, and subsequently reproduced from, memory.