

BCS Higher Education Qualification

Diploma

April 2022

EXAMINERS' REPORT

Software Engineering 1

General comments

The standard of written English was once again generally good. However, the performance of candidates varied greatly between questions.

Key messages:

- Candidates need to **take careful notice of mark allocation**, making a clear point for each mark available and providing justification for their answer where a judgement (e.g. advantage/disadvantage) is stated. For example, where 6 marks are available a comprehensive and detailed answer is required for full marks, typically involving explanation and examples to demonstrate correct and thorough understanding.

It is critically important that candidates read the questions carefully and ensure they answer each part fully.

Question number: A1

Examiners' Guidance Notes

- a) Answers for *group composition* focussed on IT skills and most did not mention personality type. *Group cohesiveness* was not usually well defined, but examples of mutual support were more common. Importance of *communications* was understood but answers were often too brief without specific examples.
- b) *Maintenance debt* was very poorly understood, with some candidates simply defining maintenance or identifying it as a real rather than hidden debt. Correct examples of ways to reduce maintenance costs were provided and attracted some credit.
- c) Most candidates correctly described the 3 types of software maintenance but some lost marks for not including the technical terms of *Corrective*, *Adaptive* and *Perfective maintenance*.

Question number: A2**Examiners' Guidance Notes**

- a) Candidates generally showed correct understanding of *unit* and *integration* testing. However, answers were often brief and didn't attract the 6 marks available. Good answers correctly explained verification and related it to their answers. Good answers also provided detail of how unit testing would take place using white box testing and explained the technique.
- b) Test plans were often very briefly described. Good answers clearly explained *validation* and how the test plans were a key part of *product traceability* and provided examples of plan contents.
- c) Unfortunately, many candidates seemed to think Alpha testing did not involve end users, though most candidates explained beta testing correctly.

Question number: A3**Examiners' Guidance Notes**

- a) Candidates usually provided only a brief and quite vague explanation of the cost estimate methods which (while often showing some correct understanding) attracted low marks. The best answers discussed the Delphi technique for expert judgement and included detail such as function point analysis and named algorithms such as Cocomo for part iii.
- b) Most candidates who attempted this question showed a vague understanding of project, product and business risks and did not provide a definition. The best answers also provided examples to explain their answers.
- c) Few candidates answered this part; good answers usually provided an analogy with insurance and provided a concrete example of outsourcing part of the project development.

Question number: B4**Examiners' Guidance Notes**

34% of candidates attempted this question. The mean mark was 6, pass rate: 22%.

Part a) of this question was generally not well answered. Whilst many candidates could form a reasonable representation of the problem specification. Many failed to correctly draw a static class diagram. It was apparent that many candidates understood the basic representation of the problem as a static structural diagram, however many failed to include an appropriate number of classes and did not fully include all of the pertinent attributes. Very few answers gave a convincing representation of relationship between classes

Part b) of this question was poorly answered. Very few candidates were able to draw a correct sequence diagram. In many cases the sequence diagrams did not show consistency with the class diagram. Very few answers correctly showed a sequence which ended with the final destination outcome, which was the successful completion of a journey.

Question number: B5**Examiners' Guidance Notes**

Part a)

i) Many candidates gave a reasonable account of the distinction between iterative and incremental development methods, although some had difficulty in answering the question in terms of the implications for misinterpreting the methods. In a few cases answers simply ignored the question context and described general principles of the methods.

ii) This sub section of the question was well answered by a majority of candidates who showed an appreciation of the disadvantages in using the incremental method. Many candidates provided a fairly extensive answer, showing a range of possible disadvantages, when only TWO were asked for in the question.

iii) In this sub section many candidates were able to provide a good account of the reasons for choosing an appropriate development method. In some cases, candidates did incorrectly suggest that prototyping techniques or an agile approach was appropriate to cope with changing requirements.

Part b)

This part of the question was not so well answered. Many candidates simply referred to a repository as some form of backup for code and did not describe the desired features of a software repository. In many cases answers did not address the question context about software management.

Question number: B6**Examiners' Guidance Notes**

96% of candidates attempted this question. The mean mark was 12.7, pass rate: 76%.

This question was consistently well answered by a majority of candidates.

Part a)

i) Almost all candidates correctly mentioned the prototype as a valuable tool in validation of requirements. Many answers indicated an appreciation of usability, quality and training as reasons for prototyping

ii) Surprisingly, many candidates had difficulty in distinguishing the throwaway prototype from a rapid prototype or prototype development method. Many candidates tended to repeat answers given in sub section i) and did not focus on the use of the throwaway as a cheap/quick way of refining ideas/concepts.

iii) This sub section of the question was less well answered. Many answers seemed to be confused with trying to reconcile the differences between prototypes from sub section i) and sub section ii). Subsequently they tended to answer in terms of the development prototype method rather than the throwaway.

Part b)

This part of the question was well answered by a large majority of candidates. A few answers indicated that there was some confusion between the use of COTS and in-house software development being used in software re-use.