Examiners' commentary 2018–2019

CO3353 Software engineering project management – Zone A

General remarks

Overall, candidates performed well on this examination. Answers to questions were mainly well structured and candidates had taken care to read the questions and provide appropriate answers.

However, there is still room for improvement on this paper in that there is a tendency for some candidates not to take sufficient care in the presentation of their answers. For example, each question or sub-question should always be started on a new page.

Another drawback in the performance of this paper was that candidates failed to pay sufficient attention to their handwriting. In many instances the examination script proved very difficult for the examiners to read. Candidates are reminded that it is very important that they present their work in a clear and legible way.

Comments on specific questions

Question 1

a. There were a number of different ways to go about answering this question, most of which were acceptable. Good answers focussed on a wide range of issues that may result from adding functionality after a system is in operation. These included discussions around the technical issues, but also considered the impact of undertaking these updates when the team had already disbanded. Other good answers considered the financial implications. Excellent answers placed the various implications into categories.

b. This was a straightforward question. Candidates were required to think about and set out the differences between agile and conventional project management methods which most of them did well. Where candidates linked their answer very clearly to the scenario of UOS, they scored additional marks. The question required candidates to give a clear indication at the end of the question as to which method they felt most appropriate. Simply comparing the advantages and disadvantages of each method was insufficient without coming to a firm conclusion on the basis of the scenario given.

c. Again, this was a question that allowed for a range of different appropriate answers. The key to getting high marks was to directly address what was being asked, that is, to what **extent** do you believe that good project management is important in the given scenario. Candidates therefore needed to put forward strong supporting statements for their point of view. Where they added examples or evidence to support their assertions they scored well.

d. Some good answers put the key issues that go wrong in software engineering project management into categories. For example, a category for technical, social or external or internal forces.

Question 2

a. This question required a definition of requirements validation. Where candidates were able to provide this, they scored the full three marks available.

b. If a clear definition had been provided for 2a), it was easier for candidates to discuss why this process is so important. Good answers pointed out that requirements validation is not only necessary at the outset of the project to ensure that the system meets the needs of the client, but it is also appropriate at various intervals throughout the project, as a result of the fact that requirements change from time to time.

c. In general, answers to this part of the question were extremely vague. Candidates scored high marks where they were specific and focussed in their answers, highlighting the importance of validity checks, consistency checks, completeness checks, realism checks and verifiability.

d. This was a straightforward question requiring knowledge of what is meant by validation and what is meant by verification. Many candidates were confused as to the distinction between these two terms.

Question 3

a. The key to earning high marks here was to read the question carefully. It asked for a critical assessment of the Use Case as model for requirements gathering. Candidates were therefore expected to discuss both the advantages and disadvantages of this method. It was insufficient to merely describe use cases or only to consider their benefits.

b. Disappointingly, many candidates were unable to provide a clear definition of what is meant by refactoring. This topic is fully discussed in the recommended textbooks for the course.

c. Where candidates had clearly defined the meaning of the term refactoring in 3b), they were able to discuss the benefits of refactoring such as the elimination of duplicate code and its impact on the structure of the data.

d. There are a wide range of issues that affect the maintainability of software. Many of these issues were discussed and there was no single right answer to this question. Such issues include references to the documentation, the availability of the skills base, the degree of modularity of the code, etc. All of these were acceptable.

Question 4

a. There are different ways in which a software requirements specification (SRS) document may be constructed, and many answers pointed out that this is the case. They then suggested the typical headings that might be included in the SRS such as the purpose and scope of the document, the project overview and system features, the external interfaces and the supplementary requirements.

b. This question was well answered. These issues are outlined in the CO3353 subject guide. Candidates who had studied the subject guide were able to identify questions the customers should be asking. These were questions concerning the project outcomes, for example, does the supplier know what we are trying to achieve, do the benefits outweigh the costs, etc. Whereas the supplier is concerned with questions such as: can all

parties agree on the overall architecture, and are we listening to the right stakeholders?

c. Most answers discussed the fact that business requirements define the scope of the project, user requirements define the system to be produced in sufficient detail to be able to plan the project, and system requirements define the system in sufficient detail to allow it to be implemented.

Question 5

a. Answers to this question were disappointing. All that was required were descriptions of the three types of testing, such as development testing, which is carried out in order to discover bugs and defects in the system. Many candidates failed to discuss who is involved in the testing process, such as system designers, programmers, etc.

b. What was required here was a clear description of the concept of software maintenance.

c. Good answers to this question included discussions on types of software maintenance, such as fault repairs, environment adaptation and functionality addition.

d. As this part of the question was only worth 3 marks it did not require a great deal of elaboration. It did, however, require the clear definition of issues such as:

- The number and complexity of system interfaces.
- The number of inherently volatile system requirements.
- The business processes in which the system is used.