

BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 6 Professional Graduate Diploma in IT

SOFTWARE ENGINEERING 2

Friday 30th September 2016 - Morning

Answer **any** THREE questions out of FIVE. All questions carry equal marks.
Time: THREE hours

Answer any Section A questions you attempt in Answer Book A
Answer any Section B questions you attempt in Answer Book B

The marks given in brackets are **indicative** of the weight given to each part of the question.

Calculators are **NOT** allowed in this examination.

SECTION A

Answer Section A questions in Answer Book A

A1.

- a) Reliability and Usability are important software quality attributes. Give a brief explanation of both attributes and discuss which of these attributes is easier to quantify and measure. **(8 marks)**
- b) Object Points and Function Points are general, high level system size metrics. Which aspects of the software system are taken into account in the Object Point metric, and which in the Function Point metric? **(8 marks)**
- c) ISO standards define a set of software quality characteristics and sub-characteristics. They specify which quality characteristics are influenced by which sub-characteristics.
 - i. Discuss briefly quality sub-characteristics affecting software reliability and software usability. **(5 marks)**
 - ii. Explain why highly reliable systems tend to be less efficient. **(4 marks)**

A2.

- a) With respect to Lehman's laws of software evolution, state the two most fundamental laws and explain their implication for software lifecycle management.

(5 marks)

- b) When you are assessing a legacy system, you have to look at it from a business perspective and a technical perspective. From a business perspective, you have to decide whether the business really needs the system. From a technical perspective, you have to assess the quality of the system and its related support software and hardware. You then use a combination of the business value and the system quality to take one of the following informed decisions: scrap the system, re-engineer the system, replace the system, continue the system's maintenance.

Your task is to assess legacy systems in your organization and decide what would be the most appropriate strategy for maintaining these systems.

- i. Discuss possible factors you would use when assessing the technical quality of the legacy system.

(10 marks)

- ii. Assume that you assessed four systems and the results of the assessment are as follows:

System A: high quality, low business value

System B: high quality, high business value

System C: low quality, low business value

System D: low quality, high business value

What would be your recommendations for each of these systems? Justify your decisions.

(10 marks)

SECTION B
Answer Section B questions in Answer Book B

B3.

- a) Compare and contrast the main features and practices of the agile approach and more traditional approaches at each of the key phases of the software development life cycle. **(16 marks)**
- b) Discuss how the clearly identifiable good practices in agile methodologies can be effectively incorporated into any software life cycle environment. **(9 marks)**

B4.

- a) Compare and contrast the following pairs of software lifecycle models, giving particular attention to the application of tools, techniques, and project life cycle phases as progress is made towards a complete system:
 - i. The V-Model and Evolutionary development **(9 marks)**
 - ii. Extreme programming and Incremental development **(9 marks)**
- b) Discuss the extent to which the choice of lifecycle models impacts, influences, and determines project test planning and testing techniques. **(7 marks)**

B5.

- a) Discuss how the use of Object Constraint Language (OCL) in UML makes designs more logically robust and easy to understand. Illustrate your answers with OCL statements. **(10 marks)**
- b) Explain what is meant by the static and dynamic semantics of behavioural diagrams. Give some examples. **(5 marks)**
- c) Discuss whether UML models can become the primary means of communication within development teams and the contract between developers and customers. **(10 marks)**