

**BCS THE CHARTERED INSTITUTE FOR IT**  
**BCS HIGHER EDUCATION QUALIFICATIONS**  
**BCS Level 5 Diploma in IT**

**SOFTWARE ENGINEERING 1**

**Examiners Report March 2018**

**Section A**

**A1.**

Testing is an important aspect of software development and maintenance.

a) Explain the purpose of each of the following different types of testing:

- i) Unit testing;
- ii) Link testing;
- iii) System testing;
- iv) Volume testing.

**(12 marks)**

**Answer Pointers**

Unit testing is used to test the functionality of each unit of code (e.g. program, module, sub-routine)

Link testing is used to test the linkages between different units of code (e.g. that parameters are passed correctly).

System testing is used to test the system as whole. That is testing that complete transactions function correctly.

Volume testing is used to test that the system can handle the anticipated transaction and data volumes.

b) Explain how the types of testing in part a) are used together.

**(6 marks)**

**Answer Pointers**

Typically, unit testing should be conducted first, then when all the units function correctly, link testing can be used to test the linkages between the units. If link testing is successful, then system testing can test transactions through the whole system. If system testing is successful, then volume testing can be used to test the capacity of the system.

- c) Explain what is meant by regression testing and why this is important in maintenance activities.

**(7 marks)**

### **Answer Pointers**

Regression testing involves testing the system before any changes, making the changes and then testing the system after the changes. The only differences in the results should be those intended by the change. Regression testing is important since otherwise new errors could be introduced by a maintenance change to a system.

### **Examiner's Guidance Notes**

This was the most popular question in Part A by a narrow margin, with just over three quarters of candidates attempting it.

For part a), many candidates restricted their answers to a brief definition of the four types of testing. The phrase "explain the purpose of..." in the question suggests the answer should give the "why" as well as the "what". Volume testing was the least well understood.

For part b) the majority put the test types in correct sequence, but then went on to summarise the substance of their answer to part a). Whilst there is naturally some overlap between these questions, good answers to part b) aimed to give an explanation of the relationship between the test types (beyond one following another), set in the context of an overall testing strategy.

In part c) there is evidence that most candidates had a reasonable understanding of why regression testing is important, but very few explained adequately the mechanism by which it is achieved.

### **A2.**

- a) Discuss what is meant by Corrective, Adaptive and Perfective maintenance.

**(9 marks)**

### **Answer Pointers**

Corrective maintenance concerns correcting errors in the system (e.g. incorrect totals)

Adaptive maintenance concerns changing the system to cater for new requirements (e.g. new business requirements or new legal requirements)

Perfective maintenance concerns improving the system to enhance operation (e.g. adding a web interface to make a system more widely available).

- b) Discuss the reasons why software systems require maintenance.

**(9 marks)**

### **Answer Pointers**

Software systems may require maintenance due to changing business requirements (e.g. new product lines being introduced in a manufacturing company)

Software systems may require maintenance due to changing legal requirements (e.g. new tax levels, new data protection requirements)

Software systems may require maintenance to systems environment changes (e.g. new versions of operating systems)

- c) Discuss why software systems typically become more difficult to maintain over time.

**(7 marks)**

### **Answer Pointers**

Software systems typically become more difficult to maintain over time due to documentation becoming out of date, the number of changes making the code more difficult to follow, and programs becoming less structured.

### **Examiner's Guidance Notes**

This was the second most popular question in Part A and had the highest pass rate. Part a) asked for a discussion, but there is evidence that some candidates repeatedly restated the question, while others gave definitions. Better answers gave examples of the sorts of changes made to a system as a result of the various maintenance activities.

Most answers to part b) correctly identified changing requirements. There was a lot of text from candidates about maintenance preventing system performance degrading over time, without further explanation.

Answers to part c) were generally quite poor, with most candidates not answering the main point of the question. The evidence shows that many answers discussed a business/financial case for not performing maintenance (mostly citing obsolescence and a lack of skilled developers) but didn't address the fundamental technical issue – that of increasing complexity of a repeatedly modified codebase over time.

**A3.**

- a) Discuss which UML techniques would be best suited to modelling the user interaction and data storage aspects of website design. **(10 marks)**

**Answer Pointers**

The user interaction aspects of website design could be modelled using use case diagrams which would show the different functionality to be provided by the website for different types of website user.

The data aspects of website design could be modelled using class diagrams which would show the data entities required for the website, the different data attributes required and the methods (functions) that would be applied to the different data entities.

- b) Discuss which UML technique could be used to model the transactions conducted through a website. **(5 marks)**

**Answer Pointers**

Activity diagrams could be used to model the transactions conducted through a website which would show all the activities involved in a given website transaction and their sequence, iteration, and decision points.

- c) Discuss whether a prototyping approach or a waterfall approach would be best suited to website development. **(6 marks)**

**Answer Pointers**

Typically, a prototyping approach would be better suited to website development due to the difficulty of completely specifying the visual layout and navigation of a website that would be required if a waterfall life cycle model were to be used.

- d) Discuss why testing a website using different versions of different web browsers is important. **(4 marks)**

**Answer Pointers**

Testing a website using different versions of different web browsers is important since otherwise different features of the website may not function correctly or display correctly in some versions of some browsers.

## **Examiner's Guidance Notes**

This was by some margin the least popular question in Part A and attracted the fewest marks.

Several answers to part a) referenced Entity-Relationship diagrams and Data Flow Diagrams, neither of which are part of UML. There is evidence that some answers set out all the candidate was able to explain about the diagram they had selected, but rather fewer related their discussion to the problem domain of website design.

For part b) a large number of candidates put forward sequence diagrams. Unfortunately, these diagrams show internal communication between objects within the system, whereas the transaction the question is asking about embodies the external communication between the user and the system. Activity diagrams are much more appropriate to capture the control/flow of this communication. Alternatively, detailed use case descriptions (not diagrams) with pre/post conditions and alternative flows might capture similar design aspects.

Parts c) and d) were generally much better answered.

## Section B

**B4.**

- a) The software development process can be represented by a waterfall model. Describe the FIVE principal stages of this model.

**(15 marks)**

### Answer Pointers

The classical theoretical model.

The five phased stage model. Some authors provide liberal interpretation of the original Royce model and have implementation as the final stage/phase such models are equally Valid if the stages are explained

Requirements – goals and constraints user's requirements gathered

Design - Software and hardware abstractions are arrived at system architecture defined

Implementation – programs produced testing begins of system functions

Operation & maintenance – system install and working life maintained.

Waterfall model has each phase signed off and feedback to previous stages.

Essential points are feedback as a result of in phase sign off and feedback to earlier phases as a result of test and rework. 3 marks each described phase/stage.

- b) Explain any TWO benefits that an incremental software development process model might have compared to the waterfall model.

**(10 marks)**

### Answer Pointers

This non-exhaustive list or similar points would be acceptable explanations of incremental development.

Incremental development is an evolutionary approach to production. The evolutionary method allows changes to be immediately refined through iterations before moving forward it is thus thought superior to the waterfall model which goes through a process of forward separate stages. Cost is therefore a considerable benefit over waterfall methods – Costs of changes to user requirements is reduced the volume of analysis, documentation and overall testing is reduced compared to waterfall methods. Clients see parts of the system actually working allowing quick feedback/changes. The early increments allow easier buy-in from clients on timescales for the complete system. It is simpler to view progress from partial complete programs rather than design documents. It is also easier for clients to undergo staff training/familiarization as parts of the system are produced and evolve.

5 marks for each explanation that makes a valid comparison.

### Examiner's Guidance Notes.

This question was attempted by almost all candidates and had the highest pass rate of this section.

Part a) was almost universally well answered with candidates demonstrating a good understanding of a fundamental model of software engineering.

Part b) was less well answered with many candidates able to provide a good list of benefits but no reasonable comparison with the waterfall model.

#### **B5.**

- a) Computer Aided Software Engineering (CASE) tools are used to automate aspects of the software development life cycle.
- i) Give TWO examples of an upper CASE tool that might be used in a software project; **(8 marks)**

#### **Answer Pointers**

Answers here can include an example of a tool that the candidate has actually used or for a general description that notes an understanding of an upper case tool that deals specifically with that part of the software life cycle. Upper tools are those used at the beginning phases (by reference to the waterfall model) upper tools would be used by the requirements / analysis team. An upper tool would be Business process tool giving global meta data for an organization, process modelling tools to provide a detailed analysis of process flows in a business. A range of management tools (Microsoft Project) for tracking monitoring time/risk. System analysis tools. An upper tool might also include documentation (Word/Excel) tools that allow higher level analysis and reporting. 4 marks each.

- ii) Describe any THREE lower CASE tools that might be used in a software development project. **(9 marks)**

#### **Answer Pointers**

There are many hundreds of lower case tools that can be cited the essential point is identifying the phase of the life cycle that 'Lower' refers to. Lower case refers to end points of the software life cycle. Tools would include compilers, configuration management tools, design tools, IDE's and automated test suites. Candidates might give specific examples of such tools (NetBeans, Git, Eclipse etc.) 3 marks for each.

- b) Explain any TWO functions you would expect to find in an integrated CASE tool repository. **(8 marks)**

#### **Answer Pointers**

The iCase tools have to provide functions that enable the whole of the life cycle to be accommodated. Thus, functions needed include a sharing mechanism that communicates with all of the phases of the life cycle. Change tracking to inform all other phases. Version control a central repository, communications support for software engineers and managers. The repository should therefore perform base functions such as data integrity checks, information sharing, tool sharing and

integration, a database system, documentation standards. An explanation of any two such functions 4 marks each 8 marks total

### **Examiner's Guidance Notes.**

This question was the least popular of section B, with only about a quarter of candidates attempting it. Half of those candidates attempting this question failed to achieve a pass mark. Answers were in general poor with half of the candidates not able to place the tools in the appropriate category of upper or lower. The evidence shows that although most candidates had a grasp of some CASE tools, many had a vague idea simply citing a single tool being used throughout the lifecycle.

### **B6.**

- a) Describe any FOUR types of risk that might be identified in a software project checklist.

**(12 marks)**

### **Answer Pointers**

Typical risk checklists

Technical -coming from software and hardware issues

Personnel – arising from staff skills team adhesion personalities

Organizational- type of development organization management styles, funding models

Development tools the software and hardware platforms reliability familiarity

Requirements- types of development model ability to cope with changing requirements, customer expectations

Cost estimators – cost/time risks from errors in estimating bias in deriving estimates  
3 marks each

- b) Outline the difference between an avoidance strategy and a minimization strategy in project risk management.

**(8 marks)**

### **Answer Pointers**

Avoidance is to reduce the probability of the risk event e.g. Poor time/cost estimate should investigate costs of buying off the shelf components, accessing API's from outsourcing so alternative are available. Any similar examples from the particular risk  
Minimization is to reduce the impact of the risk once it has happened e.g. Staff illness mitigated by having redundancy in skill sets to allow substitution of skills to cover absence

4 marks each

- c) Describe any TWO business risks that might be identified in a software development project.

**(5 marks)**

### **Answer Pointers**

Correctly identify business risk - then typical are making wrong product (no market)  
building redundant/dated or project (about to be superseded) giving a business



strategic risk. Complicated/difficult product to sell (understand). Loss of internal sponsorship/support – loss of funding.

**Examiner's Guidance Notes.**

This question was the second most popular in part B. The question was well answered by most candidates with Part a) being very well answered by almost all.

There is evidence that candidates struggled with part b) with many being unable to distinguish between avoidance and minimization of risk, and just repeating the same risk across their answer to each category. Most candidates listed many types of risk associated with development projects but did not demonstrate an understanding of how that risk can be managed through avoidance or minimization.

Part c) was poorly answered by most with candidates unable to correctly identify a purely business risk, most answers introduced other categories of risk and perhaps mentioned a business risk as part of a general list.