

**BCS Professional Examination 2015
Professional Graduate Diploma**

April 2015

Examiners' Report

System Design Methods

Question 1

- 1.a) Discuss why prototyping and agile approaches to systems design are increasingly being used instead of traditional waterfall approaches. **12 marks**
- 1.b) Give a brief explanation of the following terms: software reusability, software reuse.
Component Based Systems Development (CBSD) methods place a lot of emphasis on component reuse when developing a new system and on developing ('fabrication') of new reusable components. Identify the main stages which should be provided by a typical CBSD method. **13 marks**

Answer pointers

- 1.a) The following (or suitable alternatives) would be expected:
- The availability of graphical based software development tools supports prototyping and agile development approaches and allows systems design models to be quickly and easily developed for user feedback. This allows for more user involvement than the more limited user role in specification in a waterfall approach. **4 marks**
- Businesses are increasingly looking to reduce systems development project lead times and using a prototyping or agile development approach can be one means of achieving such a time reduction. **4 marks**
- Prototyping and agile methods can cater for uncertain requirements and can be used to develop and refine the requirements for the system. **4 marks**
- 1.b) Explanation of software reusability e.g. "Reusability – the extent to which a program (or parts of a program) can be reused in other applications." **2 marks**
- Explanation of software reuse e.g. "Software reuse – an approach to software development that tries to maximise the reuse of existing software". **2 marks**

A typical CBSD method is likely to have 2 major processes: the solution process and the component process.

The solution process is a typical systems development process (with 'traditional' stages of Feasibility and Analysis). These stages are followed by Design and Build stage which is less 'traditional' as it seeks to 'harvest' reuse by looking for existing components which can be used to solve the problem.

The component process is responsible for production of components. The main stages within this process are 'traditional' Design and Build followed by Acceptance (to ensure acceptance and certification of a set of components) and

Roll out (to install a set of components in the repository supporting the live user environment) stages. There should also be a stage (prior to Design and Build) to assess needs for reusable services (these may arise from the solution process(es), legacy assets, etc).

9 marks

Examiner's Guidance Notes

1.a) Many candidates answered this part reasonably well by providing reasons why prototyping and agile approaches are becoming popular. A substantial number of candidates however just discussed/described the waterfall, prototyping and agile approaches.

1.b) Many candidates did not adequately provide proper definitions of reusability and reuse. Only a small number of candidates identified the main stages of CBSD methods.

Question 2

2. a) Discuss why systems design methods commonly include techniques to model the structural, procedural/functional, and temporal/dynamic aspects of an IT system. Give an example of the techniques that model these systems aspects in a systems design method of your choice.

15 marks

b) Explain how you would cross-check the three models discussed in question 2.a).

10 marks

Answer pointers

2.a) The following (or suitable alternatives) would be expected:

Systems design methods commonly include techniques to model the structural aspects of IT systems since even the simplest of IT systems will require some kind of structure, for example data structures for holding system data.

3 marks

Examples:

Entity relationship diagrams, class diagrams

2 marks

Systems design methods commonly include techniques to model the procedural aspects of IT systems, since these describe the actual activity to be performed by the system, for example calculations and data input and output.

3 marks

Examples:

Data flow diagrams, Use cases

2 marks

Systems design methods commonly include techniques to model the temporal aspects of IT systems since data will change over time and therefore the state of the data needs to be specified in order to ensure that it is processed correctly.

3 marks

Examples:

Entity life histories, State transition diagrams

2 marks

2.b) Structural vs Procedural/functional (SSADM example):

Each datastore should hold/represent a whole number of entities i.e. a data store is related to one or more entities and an entity may not appear in more than one data store. **4 marks**

Structural vs Temporal/dynamic (SSADM example): Each ELH corresponds to an entity. **3 marks**

Procedural/functional vs Temporal (SSADM example):
Each bottom-level process that updates a main data store will be triggered by one or more events that appear on the ELH. **3 marks**

N.B. UML example can be used instead of the above SSADM example. The UML example should explain how you would cross-check e.g. classes, state transition diagrams and use case diagrams.

Examiner's Guidance Notes

2.a) Many candidates did not provide a reasonable explanation of why system design methods include techniques to model all three aspects of a typical information system. Most candidates gave proper examples of modelling techniques.

2.b) Many candidates did not address the question and discussed modelling techniques/diagrams instead. A few candidates discussed software development models (e.g. prototyping) instead of systems models.

Question 3

3. a) You are in charge of five software development projects. The 'characteristics' of each of your projects are as follows:
- Project 1. Website for a local company. Relatively small system. Requirements are vague and likely to change in the near future.
 - Project 2. A very large embedded system whose requirements can be easily identified and are relatively stable.
 - Project 3. A 'standard' business application. You have developed similar systems in the past.
 - Project 4. A relatively complex administrative system for one of the local hospitals. Some of the requirements seem to be pretty vague, but all the requirements are stable.
 - Project 5. A small real-time control system to be used for monitoring patients in a local hospital.

Consider the following software development approaches/models: waterfall, throw-away prototyping, evolutionary prototyping, component-based development, formal development.

Which of the above approaches/models would you choose for each of your projects? Briefly justify your choices. **10 marks**

- b) Agile methods (RAD methods) are not suitable for all types of systems and systems development projects.

- (i) Suggest at least five system/project characteristics that would be appropriate for Agile methods. Give a brief justification for each characteristic. **10 marks**
- (ii) Suggest also at least two system/project characteristics that would not be appropriate for Agile methods. Give a brief explanation. **5 marks**

Answer pointers

3.a) Project 1: Evolutionary prototyping (explain)

Project 2: Waterfall (explain)

Project 3: Component-based development (explain)

Project 4: Throw-away prototyping (during the Requirements Analysis stage) followed by Waterfall (explain)

Project 5: Waterfall with some 'critical parts to be developed using formal approach. (explain)

2 marks for each

- 3.b) (i) Possible system and project characteristics are:
- the system is interactive i.e. users interact with 'system functions'
 - the system is not computationally complex
 - it has clearly defined user groups
 - the project is time-constrained
 - it has requirements that can be prioritised
 - it has requirements that are unclear or subject to frequent change
- 10 marks**

(ii) Possible system and project characteristics:

- safety critical systems
- systems with minimal user-system interactions (e.g. scientific)
- complex systems whose complexity cannot be easily broken down
- management requires detailed long-term plans as a deliverable etc.

Two characteristics with appropriate explanation.

5 marks

Examiner's Guidance Notes

3.a) This part was answered reasonably well, but many candidates did not provide sufficient explanation of their choices.

3.b) Both parts (i) and (ii) were answered reasonably well. Some candidates did not discuss system/project characteristics but gave examples of specific systems (e.g. hotel reservation, school, hospital, etc.) instead.

Question 4

4. a) You are an IT manager and have decided to introduce object oriented development techniques into your IT department. Discuss which approach to implementing object oriented development you would use:

- training staff in an object oriented design method first and then an object oriented programming language,
- training staff in an object oriented programming language first and then an object oriented design method,
- or training in both at the same time.

Discuss the issues to be taken into account when considering each option.

10 marks

4. b) There are many reasons why a systems design method introduced in an organization might fail or does not meet expectations, such as:

- Productivity
- Complexity
- Skills
- Tools
- Social and organizational issues.

Provide an explanation of each of these reasons.

15 marks

Answer pointers

4. a) Would it be more appropriate to learn object oriented concepts first (for example, encapsulation, polymorphism, and inheritance) or to learn practical examples first (for example classes, method overloading and method overriding) or can the two be done together.

4 marks

Would it be more appropriate to learn how to code first or how to design first or can both be done together. For example, does learning how to code first provide insights into the structure and therefore design of programs, or does learning how to design first provide insights into the structure of programs and therefore how to code programs, or can the two be done together?

4 marks

Would it prove confusing or overly complex, or too demanding to attempt to learn both object-oriented coding and design at the same time for the IT staff concerned? In particular, intensive training may interfere with the workload of the IT staff concerned.

2 marks

4. b) Productivity: It happens frequently that the method introduced fails to deliver the suggested productivity benefits e.g. it does not reduce the time taken to develop a project, it is resource- intensive, etc.

3 marks

Complexity: Some methods have been criticised for being over-complex. They are designed to be applied to the largest projects, therefore they are 'too complex' for small projects.

3 marks

Skills: Methods require significant skills in their use. These skills may be difficult for method users.

3 marks

Tools: The tools supporting the method may be expensive and difficult to use.

3 marks

Social and organisational issues: Many methods do not take into account these issues (they focus only on technical issues). **3 marks**

Examiner's Guidance Notes

Most candidates answered part (a) reasonably well and discussed the different considerations regarding the order of learning of object oriented design techniques and object oriented programming.

Most candidates answered part (b) reasonably well and appropriately discussed the reasons why a systems design method introduced into an organisation might fail or might not meet expectations.

Question 5

5. a) There are various criteria that might be considered when assessing the suitability of a systems development method including: life cycle coverage; separation of analysis and design; visibility of product; designing for change; and 'extendibility' of the method. Discuss the importance of these criteria when assessing a method. **10 marks**

5. b) Outline the types of software tools that could be used to support systems development activities, explaining the potential benefits that they could provide. **15 marks**

Answer pointers

5. a) A method should ideally cover the entire systems development process (including maintenance) **2 marks**

The analysis of the user requirements should not be influenced by design considerations **2 marks**

The product should be specified at different stages of its development (analysis, design, etc.) **2 marks**

The logical and physical designs should be easily modified **2 marks**

It should be possible to add new techniques and tools to the method, but still maintain the overall consistency and framework of the method **2 marks**

5. b) Project management tools that can produce project plans and schedules (e.g. gantt and pert charts) for design activities. Project management tools allow monitoring and tracking of design activities.

Documentation tools can allow easy development, storage, retrieval and maintenance of system design documentation. Such tools may include a data dictionary facility (that is a definition of all the data elements and data structures used in a system, and their relationships). Documentation tools are useful for keeping systems design documentation up to date for future software maintenance activities.

Reverse engineering tools take source code as input and generate graphical design models, where-used lists etc. In systems design work such tools are useful for understanding the design of badly documented existing systems.

CASE tools can automate aspects of the development of design documentation and can in some cases generate code from the designs produced. CASE tools can often check for consistency between different design diagrams and documents, thus improving quality control, and helping to enforcing design standards in the design process.

Marks - 5 per point up to 15
15 marks

Examiner's Guidance Notes

Part (a) was generally answered reasonably well with most candidates appropriately discussing the means of assessing the suitability of a systems design method. Part (b) was generally answered reasonably well, however there is evidence that a small number of candidates confused software tools with development techniques.