THE BCS PROFESSIONAL EXAMINATION
Professional Graduate Diploma

April 2005

EXAMINERS' REPORT

System Design Methods

Question 1

1. *a)* Explain the differences between throw away prototyping, evolutionary prototyping and waterfall approaches to systems development. Which approach to systems development would you suggest for the projects characterized by:

i) precise and stable/unchanging requirements
ii) vague and stable requirements, and
iii) vague and unstable/changing requirements?

Justify your answers. **(12 marks)**

b) DSDM (Dynamic Systems Development Method) is one of the most popular RAD (Rapid Application Development) methods. It is based on a number of principles. Some of these principles are:

- active user involvement is imperative i.e. DSDM is a user-centred approach;
- the focus is on frequent delivery of products i.e. DSDM recommends a product-based approach;
- fitness for business purpose is the essential criterion for acceptance of deliverables i.e. the focus is on delivering the business functionality at the required time;
- iterative and incremental development is necessary to converge on an accurate business solution;
- all changes during development are reversible.

Use the above principles to discuss the suitability of DSDM for the systems and projects having the following characteristics:

i) interactive, where the functionality is clearly demonstrable at the user interface,
ii) has a clearly defined user group,
iii) if computationally complex, the complexity can be decomposed or isolated,
iv) time constrained,
v) the requirements are unclear or subject to frequent change.

For each characteristic clearly indicate related principle(s). **(10 marks)**

c) Would you recommend using DSDM for process control/real time applications? Justify your answer. **(3 marks)**

Answer Pointers

a) *Throw-away prototyping* – this approach is primarily used to help identify/elicit/gather user requirements. The prototype is developed from outline requirements/specification. Then it is delivered to users/clients for them to use it and experiment. Any required modifications are implemented etc. until the users/clients are satisfied with its functionality. At this stage, the prototype served its purpose and a full requirements specification document is produced and the development (i.e. design and implementation) of the actual/production quality system begins. (2 marks)

Evolutionary prototyping – it is based on the idea of developing an initial prototype/system, exposing it to user/client comments and improving/refining it, exposing it again to user/client comments, refining etc. until a adequate system has been developed. Suitable for rapid development. (2 marks)

Waterfall approach – it is a 'classical' sequential process with very limited scope for 'backtracking'. Suitable for quality systems. (2 marks)

i) waterfall – requirements are precise, so there is no need for prototyping, requirements are stable, so there is no need for rapid delivery. (2 marks)
ii) throw away prototyping (combined with waterfall) – requirements are vague, so prototyping is needed, requirements are stable, so there is no need for rapid delivery. (2 marks)
iii) evolutionary prototyping - vague requirements, so prototyping is needed, unstable requirements, so rapid delivery is required. (2 marks)

b) *i)* *Interactive.* This characteristic relates to principles 1 and 4. This type of application requires close user involvement. Users must be able to assess the functionality easily through viewing and operating working prototypes (1). This can only be achieved if the approach to systems development is iterative and incremental (4). (2 marks)
ii) *Clearly defined user group.* This characteristic relates also to principle 1. The users group can only be defined if users are actively involved. If the user group is not clearly defined, there may be a danger of e.g. ignoring some important aspects of the project. (2 marks)
iii) *If computationally complex, complexity can be decomposed or isolated.* This characteristic eelates to principle 4. (2 marks)
iv) *Time constrained.* This characteristic relates to principles 2 and 3. There should be a fixed end date by which the project must be completed. This characteristic requires the focus on frequent and timely delivery of products. (2 marks)
v) *The requirements are unclear or subject to frequent change.* This characteristic relates to 4 and 5. It relates to 4 since in this situation iterative approach (prototyping) is necessary (if unclear) or 5 (if subject to change). (2 marks)

c) For process control/real time application I would not recommend DSDM. Much of the processing for such systems is not visible at the user interface. It is usually complex and cannot be decomposed into smaller components. Also, requirements are usually precise and do not change frequently. (3 marks)

Examiner's Guidance Notes
For part (a): Generally no serious problems. Some answers explaining different approaches to systems development were too long.
For part (b): Most candidates properly identified DSDM principles related to various project characteristics and justified their answers.
For part (c) : This part caused some problems. Many candidates recommended DSDM for this project. Some of them confused real-time systems with 'real world' projects.

Question 2

2. *a)* DFDs (Data Flow Diagrams) and ERDs (Entity Relationship Diagrams) are popular systems modeling techniques. What aspects of information systems do they model? Justify your answer. Give a detailed explanation of how you would cross-check DFDs and ERDs. **(10 marks)**

b) Suggest a 'high level' systems complexity metric which is based on DFDs and ERDs. **(6 marks)**

c) What are State Transition Diagrams (STDs) and Entity Life Histories (ELHs) and what aspect of information systems do they model?

Illustrate the difference between STDs and ELHs by producing a STD and ELH for entity X whose instances are created by event *Ev1*, updated by event *Ev2*, and deleted by event *Ev3*. Assume that *Ev2* can affect an instance of X several times. **(9 marks)**

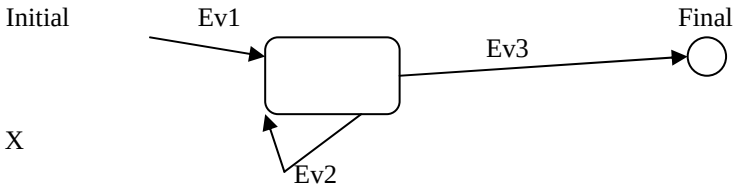
Answer Pointers

a) DFDs primarily model/specify the functional aspect of information systems (but they also address the data aspect). The functionality of a system is represented by processes. The data aspect is 'represented' by data stores and flows. (2 marks)
ERDs model data/structural aspect. Entities consist of data attributes. (2 marks)
Cross-checking:

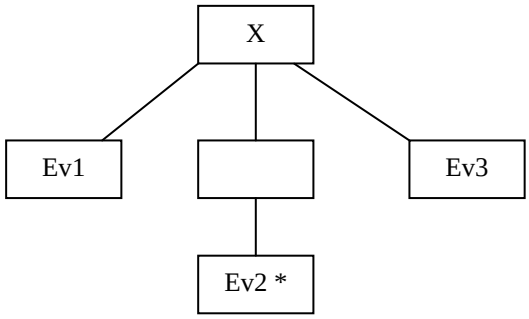
- Each data store should represent a whole number of entities i.e. a data store is related to one or more entities and an entity may not appear in more than one data store. (2 marks)
- Attributes on data flows should belong to entities (related to data stores) (2 marks)
- Each entity should 'have' a process responsible for creation and modification (and possibly deletion) of instances of this entity. (2 marks)

b) The following elements of DFDs can be taken into account when 'calculating' complexity: No of 'elementary'/atomic processes. No of data stores, No of data flows, etc. And the following elements of ERDs can be taken into account: No of entities, No of relationships, No of attributes.. (6 marks)

c) STD – a diagram which shows the various states (of a system, object, entity) and transitions between states. Usually events that cause transitions are also shown (and initial and final states).
ELH – a diagram showing events that may cause a particular entity to be changed in any way. It shows the valid structure/combination of events.
Both model temporal/dynamic aspects of information systems (5 marks)



STD for X (2 marks)



ELH for X (2 marks)

Examiner's Guidance Notes

For part (a): No serious problems with the modelling 'aspect' of DFDs and ERDs , but only a few candidates managed to identify more than one 'rule' for cross-checking. Some answers were too long as candidates provided quite complex examples of DFDs and ERDs.
For part (b): This part caused many problems. Only a few candidates provided satisfactory answers. Many candidates discussed various (irrelevant in this case) complexity metrics e.g. cyclomatic complexity.

For part (c): Explanations of STDs and ELHs were generally satisfactory, but diagrams caused some problems.

Question 3

3. *a)* Discuss how the implementation of a systems design method within an organisation might fail, and outline the measures that could be adopted in order to attempt to avoid such failure. **(15 marks)**

b) Outline and evaluate the process of reverse engineering. **(10 marks)**

Answer Pointers

a) The implementation of a systems design method might fail because the staff using the systems design method might not have been trained sufficiently in the method. In order to attempt to avoid this situation appropriate training methods could be used such as short courses, computer based training packages (where available) or a train the trainers approach whereby a small group of staff are trained and then train the other staff. (5 marks)

The implementation might also fail because there is insufficient on-going support for the use of the systems design method. This can potentially be avoided by having readily available documentation and training materials, and / or designated members of staff who will answer any queries regarding the use of the systems design method. (5 marks)

The implementation of the systems design method might fail due to staff being unwilling to use the design method or using it inappropriately. This can potentially be avoided by encouraging the staff to use the method, and incorporating the use of the method into quality control measures used within the organisation such as inspections and reviews. (5 marks)

b) Reverse engineering involves:

Identifying inputs to a system such as data entry fields on screens and possibly other sources such as bar code readers. This is useful because it helps to confirm a match between actual operation and the coding that handles inputs.

Identifying outputs from a system such as report screens, printed reports and possibly other forms of output such as graphics and charts. This is useful because it helps to confirm a match between actual operation and the coding that handles outputs.

Examination of the source code for the system to determine the structure and detail of the coding. This is required because it has to be determined how and where the coding should be changed.

Examination of the documentation for the system such as entity relationship diagrams or use case diagrams. This is useful because it can hopefully help to confirm the view of the systems operation gained from the code.

Establishment and documentation of the modules within the system, the connections between modules, the main functions of individual modules and the data stores used based upon the above. This will hopefully assist in making future maintenance easier. (10 marks)

Examiner's Guidance Notes
For part (a) most candidates gave reasonable answers, and outlined appropriate mechanisms for reducing the potential for failure e.g. staff training and on-going support in the form of mentors.

For part (b) most candidates could explain how reverse engineering was undertaken.

Question 4

4. *a)* You are an IT quality manager in a large retail company. You have been asked to create a systems design method for the company's intranet. Discuss what you would include in such an intranet design method. **(16 marks)**

b) Discuss the benefits and drawbacks of formal methods for the design of safety critical applications such as medical and aerospace applications. **(9 marks)**

Answer Pointers

a) A discussion of the design techniques that could be used to model the structural aspects of an intranet (e.g. hierarchy charts, storyboards). (4 marks)

How the user interface could be modelled, for example how human computer interface design techniques could be used to design intranet web page layouts. (4 marks)

What techniques could be used to model the functionality of web pages within the intranet, for example using object diagrams to model client and server processes. (4 marks)

How database design for the database(s) underlying the company intranet would be carried out. For example using entity relationship diagrams to model the data structures required for the intranet. (4 marks)

b) The benefits of using formal methods for the design of safety critical applications such as medical and aerospace applications include:

Formal methods can provide a validation of the designs produced for such applications, since the mathematical logic of the designs can be examined. (3 marks)

Formal methods can allow verification of the coding produced from designs created using formal methods by enabling the checking of the code against the requirements expressed in the design. (3 marks)

The major drawback of formal methods is that they require skilled IT practitioners and users with good mathematical abilities to be able to use them effectively. (3 marks)

Examiner's Guidance Notes
For part (a) most candidates discussed relevant aspects of a practical intranet design approach

For part (b) most candidates were aware of the benefits and drawbacks of formal methods

Question 5

5. *a)* Discuss the relevance of computer aided software engineering (CASE) tools to systems development activities. **(13 marks)**

b) Outline and critically assess three quality assurance approaches that could be used to improve systems design activities. **(12 marks)**

Answer Pointers

a) Create and modify design diagrams such as dataflow diagrams, entity relationship diagrams or object diagrams quickly and easily.

CASE tools can often check for consistency between different design diagrams and documents, thus improving quality control.

CASE tools can assist in enforcing design standards in the design process.

CASE tools can often generate code directly from the designs produced, thus reducing the possible misinterpretation of designs. (13 marks)

b) Reviews involve supervisors or managers formally assessing the quality of design artefacts produced by a systems designer in order to identify errors in the designs. However, staff may feel intimidated by this since the results of the reviews might be used for deciding pay rises and promotions. Staff may therefore be reluctant to admit to errors and make corrections.

Walkthroughs involve a group of peers informally examining design documents in order to identify and discuss possible improvements. Hopefully this should be an egoless environment since all the staff involved will be at the same level within the organisation, and staff may be more willing to admit errors and make corrections.

Quality circles involve a group of peers discussing mechanisms for improving the quality of designs, for example developing more rigorous design standards or adjusting the design techniques used. This can be used for on-going improvement of the systems design process, rather than just attempting to catch errors as in the two mechanisms above. (12 marks)

Examiner's Guidance Notes

For part (a) most candidates discussed appropriate benefits that CASE tools can provide for systems development

For part (b) some candidates discussed the differences between quality assurance and quality control rather than providing examples of quality assurance approaches