

THE BCS PROFESSIONAL EXAMINATION
Professional Graduate Diploma

April 2002

EXAMINERS' REPORT

System Design Methods

Question 1

1. a) **With the aid of a diagram illustrate the classic waterfall software process model and explain briefly, for each stage, what activities are carried out and what product results. (10 marks)**
- b) **With the aid of a diagram explain the fundamental principles behind the prototyping approach. Compare and contrast this with the traditional waterfall approach to software development. (10 marks)**
- c) **Explain when it is most appropriate to adopt a waterfall approach, when it is most appropriate to adopt a prototyping approach and when it is most appropriate to adopt a hybrid of the two approaches. (5 marks)**

Answer Pointers

- (a) Expect classic diagram with phases to be identified: requirements definition, design, impl and unit testing, integration and system testing, operation and maintenance. Indication of how phases link together in separate linear manner with consideration of iteration.

For each phase consider:

- Requirements Definition: elicitation and analysis of requirements, building of requs model – output is the requirements specification of functional and none-functional requirements
- Design: Establish and model overall system architecture – product is an architectural model of the software system
- Implementation and Unit Testing: Realise design as a set of programmed units, and test each unit for conformance to specification – product is a set of software components, plus lower level design documentation.
- Integration and System testing: integrate separate modules and test the system as a whole to ensure requirements are met – product is complete software system (to be delivered to customer)

Operation and Maintenance: system is installed for use in practice, and ongoing improvements and enhancements take place – product is the installed system and continued maintenance support.

- (b)

Prototyping Idea:

- Develop system prototype quickly
- Expose user to it early for feedback
- Continue with cycles of improvement of the prototype and feedback until user satisfied

- Abandon prototype and develop the actual system based on the aspects now well understood

Contrast issues:

- Rather than WFs separate specification, development and validation phases all these are performed concurrently – with rapid and regular user feedback
- Process not as visible, explicit or manageable
- Prototype may end up being poorly structured, and poorly engineered
- Must be prepared to throw away the prototype – it is not the system, although clients often want it to be evolved into the system

Expect diagram to make pertinent issues explicit.

(c)

Waterfall: Better for large systems that require clear management control; when need to audit process; when system will require long term maintenance.

Prototyping: Useful for RAD projects; requirements poorly understood initially; when development involves significant investigation/research aspects; when objective of project is to better understand user requirements; can be adopted as development approach for small systems

Hybrid: Can use prototyping approach within a waterfall lifecycle either during the early stage (to better understand requirements) or for elements of interface and or technical (proof of concept) prototyping.

Examiner's Comments

Part (a): This part was done quite well in general. Most candidates managed to produce a reasonable diagram. However, some provided long descriptions of the activities and products associated with the key stages which lacked clear focus and demonstrated a lack of understanding. The best answers tended to be reasonably brief, but very well focussed, clearly indicating that the fundamental points were well understood.

Part (b) : This part wasn't done quite so well. Some candidates did not clearly explain the main principles that were asked for, even though they managed verbose answers. Also the quality of some of the diagrams was poor. Key comparison points were included in many cases, but again the best answers provided a much clearer structure for the issues presented. The less well-structured answers included a lack of focus and repetition of the same points.

Part (c) : A number of candidates got a little mixed up here, and struggled to find the main reasons for adopting the different approaches. Most candidates only provided minimal consideration of the hybrid approach, and failed to acknowledge any uses for prototyping after the initial requirements gathering phase.

Question 2

2. a) Discuss the differences in purpose and use of:
- i) entity relationship diagrams
 - ii) normalisation (the process of revising data structures into 3rd normal form) for database design.
- (4 marks)
- b) Compare and contrast the formal methods approach (the use of formal mathematics for systems design) and the socio-technical approach (the analysis of social, organisational and technical aspects of systems design).
- (9 marks)
- c) You are to develop a website for your company. Outline which system design techniques you would use for this task, justifying your choice.
- (12 marks)

Answer Pointers

- a) ER diagrams identify entities and the relationships between entities, which is useful for determining relationships between physical DB records, whereas normalisation reduces duplication and identifies the most efficient keys for data storage and retrieval.
- b) Socio-technical methods address social and organisational issues whereas formal methods address technical issues in a formal mathematical way. Socio-technical methods tend not to provide much of a framework for actual design, whereas formal methods do. Verification of specifications by users is typically easier with socio-technical than formal methods.
- c) Different approaches can be used for designing a website, however any approach should include the modelling of the structure of the website, the entities required on the website and the layout of website pages as a minimum.

Examiner's Comments

This question was answered in a reasonable manner. Most candidates were able to explain how entity relationship diagrams and normalisation relate to each other. However, the comparison of formal and socio-technical methods was generally more difficult for candidates. Discussion of website design approaches lacked detail.

Question 3

3. a) Define what is meant by the terms “metric” and “indicator” when used in a software engineering context and explain the role that they play in software product and process improvement. (5 marks)
- b) Correctness, maintainability, and integrity are useful indicators often associated with software quality. Provide a clear definition of each of these indicators and suggest appropriate metrics to be associated with them. (10 marks)
- c) Suppose that you are the manager of a software development team and have decided to introduce a number of software metrics into your organisation. Describe in detail the approach that you would take to identify the most appropriate metrics to be introduced and indicate what you would do with the results collected from the metrics. (10 marks)

Answer Pointers

(a)

Metric

- A metric is a quantitative measure of the degree to which a system, component or process possesses a given attribute.

Indicator

- An indicator is a metric or combination of metrics that provide insight into the software process, project or product itself.
- Such insight allows the project manager or software engineer to adjust the process, project or product to improve things

(b)

Correctness

- Def: Degree to which a program performs its required function
- Metric: Defects per KLOC (where defect is a verified lack of conformance to spec)

Maintainability

- Def: Ease with which a program can be corrected if an error is encountered, adapted if the environment changes, or enhanced if the requirements change
- Metric: Mean time to change (MTTC) – i.e. time taken to analyse change requests, design appropriate modifications, implement those modifications, test, and distribute to all users

Integrity

- Def: Measure of a systems ability to withstand security attacks (accidental or intentional)
- Metrics:
- *threat* - probability of particular type of attach
- *security* - probability that a specific type of attach will be repelled

Integrity is sum of relative threat and security for each type of attack (i.e. integrity = SUMOF((1-threat) * (1-security)))

(c)

Approach taken to introduce metrics

Expect an organised approach for example something based on GQM which identifies the following:

- Goals : What organisation is trying to achieve – e.g. shorter development time
- Questions: Refinement of goals. Highlights specific questions to be answered associated with the goal
- Metrics: Measurements to be collected to allow the questions to be answered

Specific example:

G: achieve shorter development times

Q: how can requirements capture time be reduced?

M: measure number of communications between analyst and client

Processing of Results

- Analyse data focussing on specific questions posed
 - Make sure feed into process improvement
 - Re-measure when process updated
- De-focus personal issues – no link to appraisal

Examiner's Comments

Although there were a number of very good and clear answers, in general this question wasn't particularly well attempted. Several candidates showed a general lack of understanding of measurement and its importance to process and product improvement. There was also some lack of clarity in the answers, with candidates providing information that wasn't asked for and neglecting to answer the core question. I would advise that future candidates focus on addressing exactly what the question is asking.

Part (a) : The metric/indicator distinction was missed by some candidates.

Part (b) : The answers to this part suffered from a lack of clear definitions. Many candidates provided lengthy explanations of the reasons why maintenance was important (for example) but failed to provide a clear definition of what "maintainability" was. Consequently credit couldn't be given. Also some of those that attempted to suggest appropriate metrics did not manage to focus on clear measures.

Part (c) : There were some good answers to this part, but some of the poorer answers described a very unstructured approach to the identification of metrics. The question asked for detail on the approach taken to identify the metrics, not for explanation of what the metrics would be. Several candidates made this mistake and some also indicated uncertainty of what to do with the results collected.

Question 4

4. You are the IT director of a small software house that specialises in packaged software for the shipping industry. Currently your packaged software runs under MS DOS. You intend to re-engineer the packaged software to run under MS Windows. There is no documentation for the packaged software other than a few pages of sales literature, and the code contains few comments. Explain how you would re-engineer the packaged software to fit in with the structured design method that is now used with your software house. (25 marks)

Answer Pointers

Re-engineering involves extracting the design of the existing system from the coding, documenting the existing system and then making the necessary changes required to coding and documentation. This would typically involve using all sources of possible information to determine the structure and operation of the existing system in terms of inputs, outputs, programs, modules, sub-routines, functions, data structures etc. The existing system then needs to be documented to the appropriate standards. Then the new requirements need to be incorporated into the system in an appropriate manner, and the documentation changed accordingly.

Examiner's Comments

This question was generally answered quite poorly. Candidates seemed unsure of how re-engineering is actually undertaken.

Question 5

5. a) **Explain how dataflow diagrams, entity relationship diagrams and entity life histories provide three different views of a system, and how they can be used to check the quality of each other.** (6 marks)
- b) **You are the IT manager of a large IT department that has recently introduced a new systems design method for use in IT projects. Briefly outline three ways in which you could assess the benefits obtained through the introduction of the new systems design method.** (6 marks)
- c) **Analysis and design methods are often supported by integrated collections of tools called CASE workbenches. With the aid of a diagram explain the logical structure of a typical CASE workbench, and briefly indicate the role of each component of the workbench.** (13 marks)

Answer Pointers

- a) DFDs show dataflows, ER diagrams show entities and relationships, ELHs show entity lifecycles. Datastores on DFDs should match with entities in ER diagrams. ELH for each entity on ER diagrams. DFD processes and ELH events should match.

- b) Benefits could be assessed via error correction workload differences, post implementation review differences and development timescale differences.
- c) Typical structure would be central repository with tool components such as diagram editors, data dictionary, report generators, code generators etc.

Examiner's Comments

This question was generally answered in a reasonable manner. However, some candidates were not fully aware of how DFDs, ER diagrams, and ELHs were used together. Measuring the benefits of introducing a method was generally well answered, as was the structure of a typical CASE workbench.