THE BCS PROFESSIONAL EXAMINATIONS
BCS Level 5 Diploma in IT

April 2008

EXAMINERS' REPORT

Systems Analysis

Question 1

Cycle rickshaws are now a common sight on the streets of London. Imported from Asia, this form of transport is quickly becoming a lucrative and highly competitive business. Bangsar Cycle Rickshaws is a new company determined to be successful through the use of information and communications technology and they want a system that supports the following functions:

A. Process a new booking by getting from the potential customer their requirements, checking that a rickshaw operator (RO) is available and that they are in the right area using the GPS function built into the handheld computer that RO's use. If we have someone available and in the right area, we accept the booking, record it, allocate it to an RO and advise the customer of the booking number and the estimated time of arrival of the rickshaw. Booking are communicated to our RO's via their hand-held computers.

B. Using hand-held computers, RO's record details of all 'hires' and bookings they undertake during a shift. Details recorded include start and end time, start location, destination and total amount charged for the hire or booking. This information is automatically transmitted back to base in real time.

C. The system will record details of a shift once it's over. This includes details of the RO, the rickshaw used, total number of hires and bookings, the total time 'hired' and the time spent waiting during the shift.

D. Each Thursday night (or as requested by the Manager) the system should run a process that calculates RO wages. Currently we do this by adding up the total number of hours worked, multiplying it by the RO hourly pay rate and then calculating 10% of the overall income earned by the particular RO in the given week. We then store these details in the 'pay file' and produce a print out for the Manager.

In supporting these processes we need to store details of:

- The thirty eight licensed RO's the company employs. We record their name, address, email address, telephone number, hourly pay rate, RO's license number and its expiry date.

- The fifteen rickshaws that the company own, each of which is identified by a given number between 1 and 15. We need to store details of when it was purchased, details of any visible damage and when it was last serviced.
- Which RO the Manager allocates to which rickshaw for their 10 hour shift (i.e. a period of work) and the details of which hand-held computer they are issued with.
- All bookings made and to which RO they are allocated.

a) Draw a Top Level Current Logical Data Flow Diagram for the above scenario.

(15 marks)

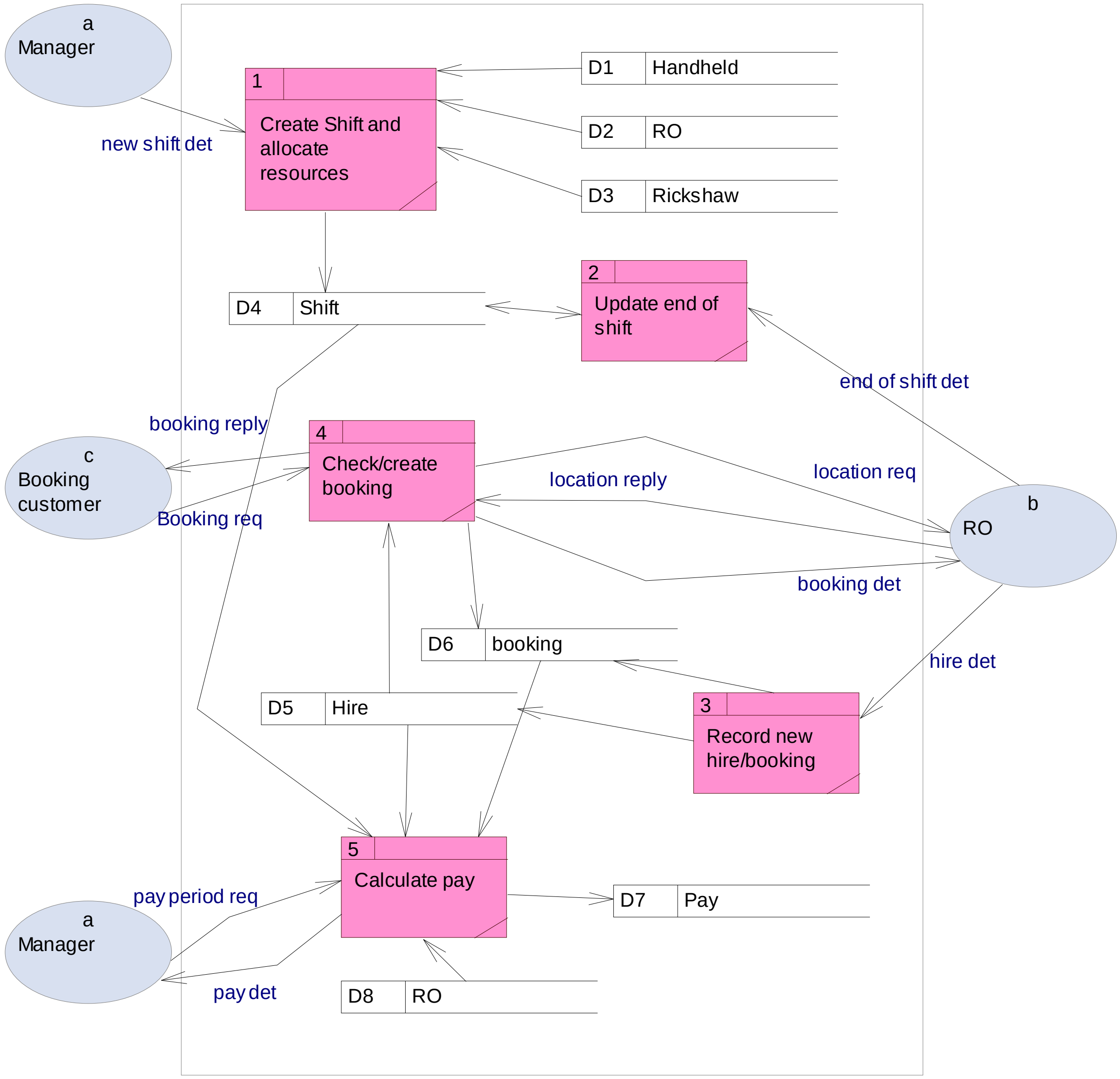
b) Produce an Entity Relationship Diagram (Logical Data Structure) and a set of normalised tables for the above scenario. DO NOT show evidence of the normalisation process.

(20 marks)

c) The third view of systems in structured systems analysis is often the Entity Life History (ELH). Using an entity from the scenario outlined above produce an ELH diagram that shows the main constructs used in this form of systems modelling and explain the role of ELH in checking the process and dat models

(15 marks)

Bangsar Cycle Rickshaws (Level 1 Diagram)

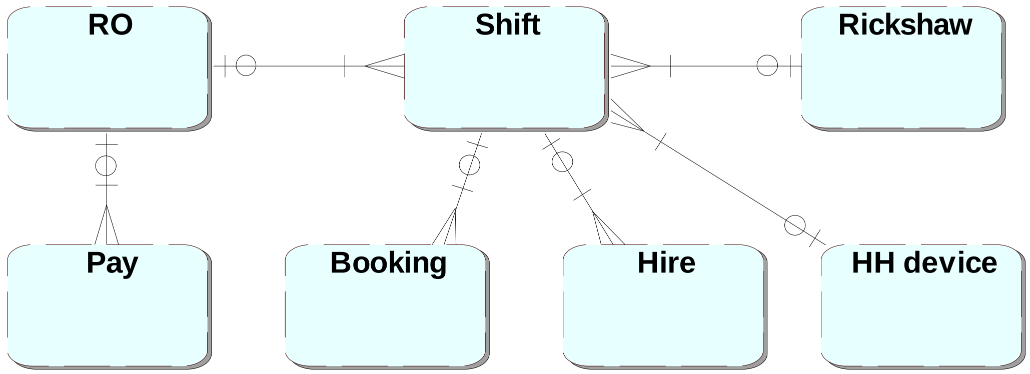


Marking Scheme:	**(Note that other suitable notation will be accepted)**
Processes:	1 each up to 5 (good process names required for max' marks)
Data Stores	0.5 each up to 2
Data Flows	0.5 each up to 5 (logical and passive)
External Entities	1 each up to 3

Total: 15 marks

Q1b.

Entity Relationship Diagram for cycle rickshaw



Tables:

RO	(**ro#**, name, address, email, tel no, hourly pay rate, RO licence#, licence exp date)
Shift	(**shift#,** date, shift start time, shift end time, time hired, time waiting, *ro#, rickshaw#, hhd#*)
Rickshaw	(**rickshaw#,** purchase date, visible damage, last service date)
Pay	(**ro#, date**, total hours, commission, total pay)
Booking	(**booking#,** date, time taken, ETA, *shift#*)
Hire	(**hire#,** start time, end time, from, destination, total charge, *shift#*)
HH Device	(**hhd#,** make, date purchase)

Marking Scheme: (other suitable notation accepted)

Entity types:	1 mark each to max of 7
Relationships with appropriate degrees & membership classes:	1 mark each to max of 5
Table types, consistency with ERD + appropriate PK:	0.5 each to max of 2
FK postings:	0.5 each to max of 3
Other attributes placed to result in well-normalised tables:	to a max of 3
	Total: 20 marks

Q1c Answer pointers:

The ELH Diagram:
As students may select their own entity to model a suggested ELH is not provided. However, whichever entity they select it should:

1. Show the main structure, i.e. JSP
2. Be top-down with a 'boss' module and 'birth', 'life' and 'death' sections.
3. Iteration, selection and sequence are also expected to be shown.
4. Status indicators will be shown by the better candidates.

Checking role:

1. It can be used to check that there are enough processes to support all events that affect entities.
2. It can also be used to check that there are enough entities and that they contain sufficient attributes.
3. Provides the analyst with a tool to model all the states an entity can be in during its life in the system and the events that change an entity from one state to another.

Q1c. Marking Scheme:

8 marks for an ELH correctly drawn from the **students own models** 8 marks
7 marks for **explaining** its role in checking other models 7 marks
Total 1c: 15 marks

Examiners Comments

DFD: The quality of DFDs submitted in response to this question was generally of an acceptable to good standard. Nearly all candidates correctly identified the main processes, stores and external entities, but some persistent errors were noted:

1. Embellishing the processing beyond that suggested in the scenario. Some candidates had processes such as "Calculate housing cost" and "Order spare parts for rickshaw". Candidate should not do this; it is outside the systems responsibility as defined in the scenario.
2. Poor process and data flow names were noted. Process names should include a verb, i.e., a 'doing' word such as 'record', 'create', calculate', etc. Process names should also clearly convey to the reader what the process actually does. 'Record det' and 'Payments' frequently appeared as suggested process names and in both cases the names could be improved.
3. Data flow names were also poor on some scripts, though many candidates are appearing to get better at this. Candidates should remember that all data flows from process to process, or from process to or from a source/sink, should be labelled with a logical and passive name. The following are examples of poor data flow names that were taken from this year's scripts, 'sent to manager' and 'give details'. These names do not describe the data that is flowing round the system.

ERM: These were generally well answered with high marks frequently awarded. Candidates are to be applauded for their increasing skills in this important area. It was impressive to see the degree of precision that was evidenced in the mapping of data requirements from the scenario to the well-normalised tables that most candidates produced.

ELH: Overall, most candidates knew a little about this modelling technique. The use of a structure diagram with a 'Boss' module and Birth, Life and Death sub-modules was well understood and presented. Many candidates were also familiar with the concepts of Sequence, Iteration and Selection and modelled them appropriately. What was less well understood and presented was the cross-checking role of ELHs. Some candidates did know this and explained it well, but too many simply did not attempt this part of the question.

Question 2

Agile and Rapid Application Development (RAD) approaches aim to deliver business solutions within a shorter timeframe than more traditional approaches.

a) Identify the advantages and disadvantages of adopting an Agile or RAD approach

(9 marks)

b) Describe TWO techniques that are used in Agile or RAD approaches to help achieve the rapid delivery of an information system. Explain how each technique helps to achieve the aim of rapid delivery.

(16 marks)

Answer Pointers

a) Advantages include:

- More likely to deliver what the client wants
- Client receives a usable system within a shorter period of time
- Helps to deliver within timescales/budgets
- Lessons learned from earlier iterations can feed into better design in later ones
- Development team gather more business intelligence as the project proceeds
- Focus is on producing a working system rather than on extensive documentation.

Disadvantages include:

- Lack of documentation could mean problems later with maintenance
- Close collaboration with client is required; this is not always possible
- Not suitable for all types of application
- Requires specific developer skills, including ability to work closely with clients.

1 mark for each suitable advantage/disadvantage identified, to a maximum of 9.
Maximum of 5 marks if only advantages (or only disadvantages) are identified.

b) Any two from:

Timeboxing: specifies deadline for project, for project phases and for iterations. Deadline may not be exceeded. Helps project deliver within specified timescales.

MoSCoW prioritization: requirements are classified as 'Must', 'Should', 'Could' or 'Wish list/Won't have this time'. In order to meet deadlines those with a lower priority can be dropped for the current iteration. Works with timeboxing to deliver priority requirements within specified timescales.

Iterative & incremental development: an alternative to a 'waterfall' development lifecycle, systems are built in increments, with iteration possible within each phase of development. Enables more rapid delivery of key functionality, where this is appropriate. Client feedback is obtained during the course of incremental development, allowing later increments to benefit.

Prototyping: by using early working versions of the eventual system, requirements can be better elicited and feedback can be obtained, thus helping to achieve the delivery of an appropriate system.

Up to 4 marks for accurate description of each technique x 2 = 8
Up to 4 marks for explanation of how the technique helps achieve the aim of rapid delivery x 2 = 8

Examiners Comments:

Poor examination technique led some candidates to spend more time on part (a) of the question – worth 9 marks – than on part (b) – worth 16 marks. There was also a tendency for some candidates to *describe* RAD instead of stating its advantages and disadvantages.
For part (b), credit was given for valid alternatives to the techniques given in the marking scheme above, e.g. simulation tools. Surprisingly few candidates mentioned 'MoSCoW prioritisation' as a technique that, alongside timeboxing, helps achieve rapid delivery. Credit could not be given for merely describing the stages of DSDM.

Question 3

a) Explain, using examples, the difference between 'functional' and 'non-functional' requirements.

(12 marks)

b) Describe what you would expect to find in a User Requirements Specification.

(13 marks)

Answer Pointers

a)

- Functional: the business processes that the system is intended to perform, e.g. to record customer details, to record business transactions such as sales of a product or the fulfilment of customer orders, or to produce management reports.
- Non-functional: requirements that are more concerned with the operational environment of the system, e.g. that a certain level of security is essential, that a particular interface style must be used, that the system must perform to certain standards or that it be portable between different hardware/software platforms.

Up to 2 marks for accurate explanation; up to 4 marks for suitable examples x 2 = 12

b) Contents might be expected to include:

- Background to the business problem
- Description of the current system
- Advantages/disadvantages of the current system
- Objectives of the proposed new system development project
- Details of all functional requirements
- Details of all non-functional requirements
- Models produced during systems analysis, e.g. use case diagrams/descriptions or data flow diagrams
- Resources and constraints

Up to 2 marks for each appropriate feature identified, to a maximum of 13

Examiners Comments

To state that 'functional requirements' are requirements for functions, or that 'non-functional requirements' are requirements that are not functions, were not sufficient as answers to part (a)! It is necessary for candidates to explain the concepts behind the words, not just repeat the words used in the question in such a way that it is not clear to the examiner whether or not they are understood. Providing some examples is a good way to demonstrate understanding. If candidates find it difficult to think of examples on the spur of the moment, they can always refer to the case study used in Question 1.

Question 4

Avison and Fitzgerald (2006) define ‘tools’ as ‘software packages that support the analysts and users in particular aspects of the application development process’.

a) Describe THREE different types of software tool that can be used in the Systems Analyis phase of a project.

(9 marks)

b) For EACH of the three types of tool you have identified in your answer to 4 a), explain the benefits they can bring to the Systems Analysis phase of a project.

(16 marks)

Answer Pointers

a) Examples include:

- Project Management Tools: allow the project manager to create a schedule of tasks to be completed (e.g. by means of a Gantt Chart), and to indicate the estimated amount of time and resource for each.
- CASE Tools: a central repository for system data; can have ‘upper’, ‘lower’, ‘integrated’ CASE tools to support different development activities.
- Prototyping Tools: a means to generate early working versions of the system, but without the full functionality of the final system; enables the rapid production of screens and draft report layouts.

Up to 3 marks for each brief description x 3 = 9
If ‘upper’, ‘lower’, ‘integrated’ CASE tools are presented as 3 different types of tool: a max of 5 marks to be awarded

b) Benefits include:

Project Management Tools:

- Aids the planning and allocation of project resources
- Enables progress to be tracked, as whatever level of detail is considered appropriate by the project manager
- Thus, can help improve the management and control of projects.

CASE Tools**:**

- Provide automated checking of model completeness and consistency, thus contributing to model quality
- Can help improve productivity, can encourage re-use.

Prototyping Tools:

- Ability to produce rapid prototypes can help elicit requirements
- Alternative designs can be tried out
- Can help users to feel engaged, helps improve their commitment
- A more effective means of communication than documentation of requirements.

Up to 2 marks for each explanation of a benefit, to a maximum of 16
No more than 6 marks for any one of the types of tool.

Examiners Comments

This question was less popular than the others, being chosen by fewer candidates than any other question, and was generally not very well answered. Part (a) asked candidates to *describe* three different types of software tool, not just *identify* them – the allocation of 9 marks should be a clue that more than just naming a type of tool was required. Most candidates who answered the question identified CASE tools, but often not another other type of software tool. Some candidates erroneously described lifecycle phases, or output such as the Feasibility Study, or practices such as walkthroughs, instead of *software tools*.

Question 5

Identify TWO models built during the Systems Analysis phase of a project using an Object-Oriented approach to systems development, and for EACH one:

a) Explain its purpose;

(10 marks)

b) Sketch an example to demonstrate the notation.

(15 marks)

Answer Pointers

Models could include: Activity Diagram, Use Case Diagram/Descriptions, Analysis Class Diagram.

a) Purpose:
Activity Diagram: To model business activities; they show the business logic, the events that cause decisions/actions to take place, and the order (flow) in which this happens.

Use Case Diagram/Descriptions: To identify each function from a user perspective (= 'use case'), and the actor(s) who perform(s) it. Can also show interfaces to other systems (an actor can be a user, a device or another system). Descriptions provide the detail of each use case.

Analysis Class Diagram: To identify the key analysis classes, their attributes, operations (methods) and the associations between them.

Up to 5 marks for accurate explanation of each model x 2 = 10

b) Complete diagram not necessary, but sufficient to demonstrate an understanding of the notation.
Activity Diagram: Should include start and end point; example of activities, decision point, concurrent activities, with appropriate labelling/naming.

Use Case Diagram: Should include Actor, Use Case, association linking the latter; examples of <includes> and <extends>, with appropriate labelling/naming.

Analysis Class Diagram: Should include Class with examples of attributes and operations; association between 2 classes; example of generalisation/specialisation and of aggregation and/or composition; appropriate labelling/naming.

1 mark for each correct feature, to a maximum of 8 per diagram x 2 to max of 15

Examiners Comments

Most candidates correctly identified an OO analysis model, and the knowledge demonstrated about Activity Diagrams, Use Case Diagrams and Class Diagrams was generally good. Some candidates mentioned design models such as State Charts or Interaction Diagrams, for which credit was not given.