Answers

1 (a) (i) There are two related problems here. The first is that it is unclear that entering data into these fields is mandatory and secondly the presence of data values in the fields is only detected late in the whole data entry process. The first problem can be addressed by either:

- Not allowing the data entry operator to skip that field or indeed to leave the field until a data value has been entered into it.
- Clearly marking in a standardised way that data entry into that data field is required. This is often denoted by showing an asterisk (*) next to the field heading.

The second problem may be addressed by validating field entries as they are entered rather than waiting for the whole of the screen to be completed. However, this may slow down data entry and so this would have to be taken into consideration.

(ii) The problem here is that the designer has not taken into consideration that a lot of data entry is either repeated or taken from a standard list of valid values. The repetition of data values can be addressed in two ways. The first approach is to have default values, which are then overtyped if they are incorrect. For example, the delivery address of a customer may default from their account address. The second approach is to fill fields with suggested values which are taken from the first unique sequence of letters typed in by the user. For example, for the customer Christine Cheepen, typing in the surname Che might automatically result in Cheepen being inserted in the second name field and Christine in the first name field. In certain circumstances, where data entry is very limited, a pull-down list of acceptable values may be provided. These values may not be overtyped, hence ensuring the accuracy of data entry.

(iii) The problem here is that the designer is offering inappropriate transitions. This is probably because the designer is re-using code or whole screens from elsewhere in the system where such transitions are appropriate. The problem may be resolved by either:

- Greying out transitions to make it clear that it is not possible to take this option from this part of the system. This is extensively used in Windows based software where the availability of the menu option depends on where the user currently is in the system and/or what they are doing in the system.
- Alternatively, the option can be removed altogether, although this may lead to a lack of consistency in the presentation of screens and dialogue in the complete system.

(b) (i) Colour has to be taken into consideration in at least two ways. The first is that the colour contrast between the foreground and background is easy on the eye and that any foreground text is easy to distinguish. The second is that the agreed colour combination must be checked for possible problems with users who have colour blindness.

(ii) The font and the font size must also be considered in at least two ways. The first is that the font is easily readable on the screen. Research has shown that certain fonts (such as Verdana) are more appropriate for screen presentation than those usually used for text printing (such as Arial and Times New Roman). Secondly, the font should be scaleable so that visually impaired users can increase the text size to fit their requirements.

(ii) Error messages must be meaningful and presented using the vocabulary of the intended recipient of the message. The message should also suggest a course of action, rather than just reporting the error. Finally, error messages should be placed so that the location of the error is obvious.

2 (a) Prototyping is the term given to an approach to systems development that stresses the delivery of partial software solutions to allow users to evaluate the functionality and usability of the proposed solution. It can be associated with an iterative approach, where user requirements are increasingly fulfilled by a series of prototypes culminating in a system which can be delivered into the live environment. This is often called the **incremental** approach to prototyping because the prototypes are successively refined until they become the final product. It is also possible to develop **throwaway or proof of concept** prototypes where the prototype is developed to, for example, agreed user requirements, but is then discarded and the final product is coded from scratch. Prototypes are often used to explore user requirements, agree the structure and conventions of the user interface and to investigate the feasibility of technical challenges facing the project.

(b) Prototyping could have been used at CAETFONE to:

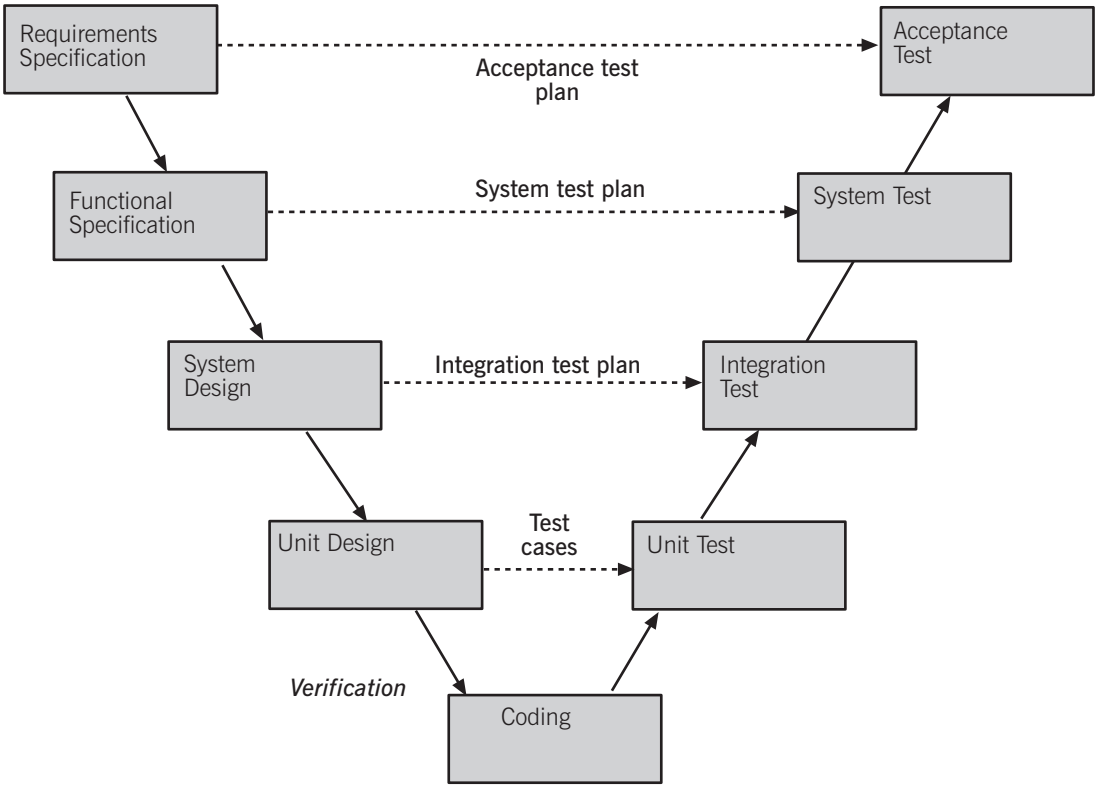
- Reduce the 'us and them' approach that appears to have been present during the analysis of requirements. Prototyping would have encouraged much greater user involvement and 'buy in' to the project. It would have shown users the analysts' thinking and allowed them to input their own ideas.
- Define the content, structure and use of the input data screens, so potentially avoiding the usability issues that arose when the system went live.
- Define the functionality of the system in such detail that later testability issues around system testing were either clarified or eradicated altogether.
- Allow an early discussion about the viability of an in-house, bespoke build and perhaps come to an agreement that the package solution would have been worth investigating.

(c) A Fourth Generation Language (4GL) is likely to provide:

- An underlying database and data dictionary.
- A query language, often built around standard Structured Query Language (SQL) constructs.
- Automatic code generation from a high level (perhaps graphical) specification.

(d) The 4GL contributes to prototyping by providing a rapid development environment where demonstration systems can be quickly constructed and shown to users. Users are often particularly interested in inputs and outputs of systems and so it is important that the 4GL has a screen design tool for data entry processes and a report generator for the easy production of output reports. In many instances it is possible to use the 4GL in a facilitated user workshop to co-operatively develop significant parts of the system during the workshop session. If incremental prototyping is being used then the 4GL must be scaleable (that is, it must offer acceptable performance in the target environment) and it must also have integrated testing and debugging facilities to help ensure that the delivered system is fault free.

3 (a) An example of the 'V' model is shown below. On the diagram, the left, downward, leg of the V shows the progress from analysis through design to programming and the increasing breakdown of the system components. The right, upward leg shows the progressive assembly and testing, culminating in the delivered product. The important feature of this model is that it shows correspondence between the different stages in the project. For instance, the individual programs or modules are tested against the individual module designs, the integrated set of software is system tested against the system design and the final system is user acceptance tested against the requirements specification. Within each stage there will be quality standards defined which will help determine the exit criteria of the stage.



(b) (i) The problem of coders releasing code to different levels of robustness would be addressed in the **Unit Design** and **Unit Test** stage. The unit design stage requires programmers to specify their intended test cases before they write any program code. These test cases will be agreed with their manager. These will then be executed in test design and if the software successfully passes the tests then it will be released to integration testing. Elements of robustness may also be encouraged by code inspections in the **coding stage** of the model.

(ii) The problem of unclear specifications for system testers would be addressed by ensuring that specified requirements are testable. Hence the relevant stage of the V model is the **Functional Specification** stage. It must be possible to produce test cases for system testing before the specification is released for physical design. Walkthroughs and inspections in this stage will help ensure the testability of requirements.

(iii) The problem of the unclear responsibilities for testing would be essentially solved by the V model itself. So, **all stages** are relevant. Part of the definition of each stage is WHO is responsible for performing the testing and quality assuring the outputs. For example, programmers are usually responsible for unit testing. Each stage of the V will define the standards for testing and these will be applied before work is passed to the next stage.

4 **(a)** The reasons why a project may fail to meet its deadline fall into three broad categories.

(1) Over-optimistic estimates developed at the start of the project. These early estimates lead to the agreement of a deadline date that could never realistically be achieved.
(2) Failure to provide promised resources during the project. The absence of both user and development staff may lead to activities slipping, so contributing to the overall delay of the project. Staff sickness would also fall into this category.
(3) Changes in requirements during the project without an associated change to the project deadline. Some of these changes will be presented 'as what we said we wanted in the first place', so making it difficult for the project manager to negotiate an extension to the project deadline date.

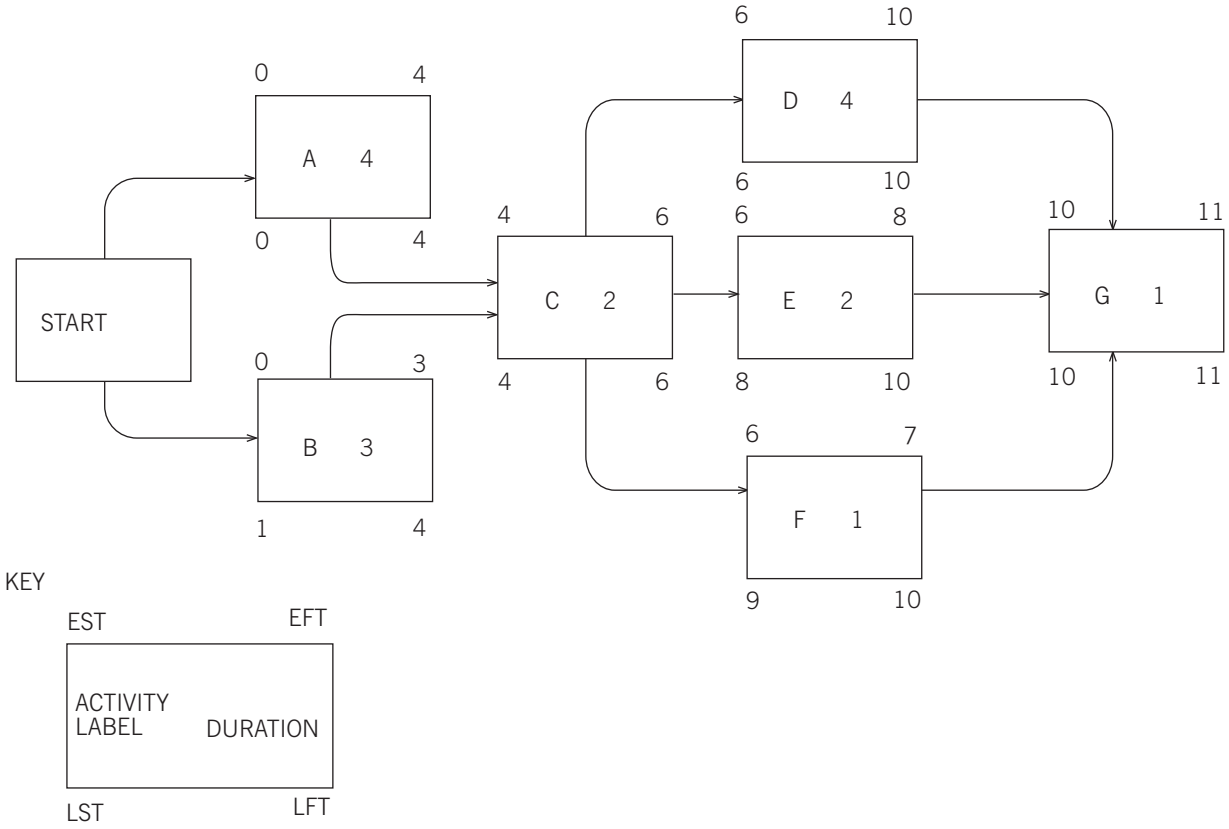
(b) A project manager may have addressed the problem by:

(1) Requesting an extension to the project deadline.
(2) Requesting that the scope of the project is reduced so that only part of the functionality is delivered on the project deadline date and the rest is delivered later.
(3) Requesting extra resources so that the deadline date may be achieved. This may be a successful strategy if applied early in the project but is unlikely to be successful if these resources are applied later. Adding more resources to a late running project often makes it even later!
(4) Reducing the quality of the product that the project is delivering. In terms of software, this might mean reducing the testing and releasing systems that still contain known faults.

Only three suggestions are required.

(c) **(i)**

Figure 1



(ii) A, C, D, G

(iii) 11 days

5 Interviews

Interviews are face to face meetings between the person trying to find the requirements (the analyst) and the person who has, or knows, or can describe the requirements (often termed the user). The interview is formal with its content defined by an agenda sent in advance to interviewees. The analyst usually takes notes during the interview and writes these up after the meeting. These interview notes are then returned to the user who confirms them (or not) as a true reflection of the interview. One advantage of interviews is that it provides an opportunity to build rapport with the user so that it is easier to overcome any future problems and difficulties. One disadvantage is that it is a relatively time-consuming and expensive way of collecting requirements and may not be viable where there are many users or where these users are spread over a wide geographical area.

Questionnaires
Questionnaires are essentially lists of questions which can be sent (usually e-mailed) to participants who are asked to respond by a certain deadline. Programs are available to automatically analyse the responses, providing a statistical summary of requirements. One advantage of questionnaires is that they are relatively cheap to distribute to many users or to users spread over a wide geographical area. One disadvantage is that their response rate can be quite low, with many users choosing not to respond at all. This may lead to erroneous conclusions being drawn on responses from an unrepresentative set of users.

Observation
Observation is concerned with the analyst watching the user go about their job and noting down requirements, problems and issues as they emerge. Observation may also include the measurement of tasks (for example, the time taken to deal with a customer query) so providing information that is not routinely available in the organisation. An advantage of observation is that it allows the analyst to see the user in their work environment and this may raise issues (such as noise, dust, physical space) which may not have emerged from an interview. A disadvantage is that knowing that they are being observed causes the user to change their usual behaviour and so erroneous conclusions are drawn.

Facilitated User Workshops
Facilitated user workshops take place when a group of users come together to work through their requirement in a group meeting. The role of the analyst is to facilitate rather than lead or control the meeting, letting users effectively drive out their own requirements. These are often documented on wall charts or used to develop prototype software which is available for immediate comment and assessment. One advantage of this approach is that the presence of a group of users means that any conflicting requirements can be immediately resolved. One disadvantage is that it may be difficult to find dates where all the key users are available, which either leads to time delays or the exclusion of users with an important contribution to make.

Printing off information from the current computer system
In some circumstances the current computer system is an important source of requirements because the proposed replacement system has to contain significantly the same functionality as the current system. Hence analysts can examine the program code of the current system to see how certain information is produced, or examine outputs to determine what reports are currently available to the user and how these reports are compiled. One advantage of this approach is that it is relatively objective because the program code of the current system is unambiguous. For example, it is possible to determine how a discount rate is allocated. However, one disadvantage is that such information may be incorrect or out of date and so it must be cross-checked with the end user.

6 (a) (i) A software licence agreement is a legal agreement between the user of the software and the person or company that has developed the software. Software is usually licensed for use – it is not sold, and the agreement defines the scope and limitations of that licensed use. For example, the licence agreement defines the scope of use (educational versions may be limited to students or teachers), how rights may be transferred, how many backup copies may be made and how the agreement may be terminated. The software licence also defines the warranty and liability of the company or person who developed the software.

(ii) Ownership – because the product is licensed, not sold, the product is not owned by the purchaser. Hence the purchaser is usually not able to sell or transfer the product to another user or to provide third party services such as renting or leasing the product for commercial gain. Ownership effectively remains with the producer of the software.

Liability – The software licence defines the limited liability of the person or company that produced the software. In many cases this is limited by agreement to the amount the customer paid for the software or to a notional amount. The licence normally includes a clause along the lines of 'in no event shall the supplier be liable for any special, incidental, indirect or consequential damages whatsoever arising out of the use or inability to use this software product'. Hence, it is usually not possible for businesses to claim consequential damages as a result of business loss caused by problems with the software.

Warranty – Software licences usually define a limited warranty. This warranty defines that the product will perform substantially as defined in the accompanying user manual. Hence if the software does not meet your specific requirements then this is your responsibility. The warranty will also define that the warranty is limited to a time period after purchase (usually 90 days) and that after this period there is no warranty of any kind. Finally, because software is not purchased, purchasers are not protected by any consumer laws concerning sale of goods.

(b) (i) Data Protection legislation is about controlling storage and access to personal information about individuals and disseminating this data to third parties. A Data Protection Act usually begins by defining the scope of stored and processed information that it applies to. Originally many Acts were applied only to automatically processed information, but now most apply to manually held data as well. The Act also defines exemptions that might apply, such as data held for purposes of national security or for the prevention and detection of terrorism. The Act will also define such terms as data subject and the rights of data subjects to access personal data about them. It will also specify what legal redress data subjects have if it is found that incorrectly held data about them has led to them being disadvantaged in some way. Finally, it will also define the offences committed by people or organisations that breach the Act.

(ii) Three principles might include:

That personal data shall be obtained and processed fairly and lawfully.
Hence the information must be obtained fairly from the data subject. The data subject must be aware of what data is being collected and how it will be used. It cannot be obtained by coercion or by deception.

Personal data shall be held only for one or more specified and lawful purposes and shall not be further processed in any manner incompatible with those purposes.
So, for example, data cannot be stored for one purpose, such as the provision of a service (say providing electricity to a customer) and also used for marketing and offering other services (such as insurance), unless these purposes have been specified. This principle applies in most data protection legislation. The information cannot be collected for one purpose and then used (unknown to the data subject) for others.

Personal data shall be accurate and, where necessary, kept up to date.
The organisation holding the data is under an obligation to take reasonable steps to verify the accuracy of the data obtained. One of the best ways of ensuring accuracy is to ask the data subject to periodically confirm details about themselves.

Part 2 Examination – Paper 2.1
Information Systems **December 2006 Marking Scheme**

1 **(a)** **(i)** 1 mark for each relevant point up to a maximum of 4 marks
(ii) 1 mark for each relevant point up to a maximum of 4 marks
(iii) 1 mark for each relevant point up to a maximum of 3 marks

(b) **(i)** 1 mark for each relevant point up to a maximum of 3 marks
(ii) 1 mark for each relevant point up to a maximum of 3 marks
(iii) 1 mark for each relevant point up to a maximum of 3 marks

2 **(a)** 1 mark for each relevant point up to a maximum of 5 marks

(b) Up to 2 marks for each area. Four areas required, giving 8 marks

(c) 1 mark for each relevant feature to a maximum of 3 marks

(d) 1 mark for each relevant point up to a maximum of 4 marks

3 **(a)** 1 mark for each relevant point up to a maximum of 11 marks

(b) **(i)** 1 mark for each relevant point up to a maximum of 3 marks
(ii) 1 mark for each relevant point up to a maximum of 3 marks
(iii) 1 mark for each relevant point up to a maximum of 3 marks

4 **(a)** 1 mark for each relevant point up to a maximum of 2 marks per reason, 3 reasons needed giving a maximum of 6 marks.

(b) 1 mark for each relevant point up to a maximum of 2 marks per way, 3 ways needed giving a maximum of 6 marks.

(c) **(i)** 2 marks for correct network flow, 1 mark for the correct forward pass figures, 1 mark for the correct backward pass figures.
(ii) Correct answer: 2 marks
(iii) Correct answer: 2 marks

5 1 mark for each relevant point up to a maximum of 4 marks per technique; Five techniques required giving a maximum of 20 marks

6 **(a)** **(i)** 1 mark for each relevant point up to a maximum of 4 marks
(ii) 1 mark for each relevant point up to a maximum of 6 marks

(b) **(i)** 1 mark for each relevant point up to a maximum of 4 marks
(ii) 1 mark for each relevant point up to a maximum of 6 marks