2007 School-assessed Coursework Report

Information Technology – Software development GA 1 and 2: Unit 3
Information Technology – Software development GA 1 and 2: Unit 4

UNIT 3

This report is provided for the first year of implementation of this study and is based on the coursework audit and VCAA statistical data.

GENERAL COMMENTS

Overall, most schools were able to satisfy the requirements of Unit 3 Information Technology: Software development. Some schools used past assessment tasks (related to the previous study design) without making the necessary modifications to adapt them to the current study design requirements.

The case studies used in the assessment tasks came from a variety of sources. It is important to note that if teachers are using assessment tasks prepared by someone else, they must still ensure that the tasks are compliant with the study design requirements. For example, for Outcome 1 a number of tasks required students to complete a series of structured questions rather than preparing a report.

As a general comment, tasks should clearly state what students must do and what they must do to be successful. This means identifying evidence associated with varying levels of performance.

While schools were not required to identify the type of programming language used by their students (40 per cent of schools were in this category), approximately 36 per cent used VB 6 or VB.NET, 18 per cent used PHP and 5 per cent used C#.

SPECIFIC INFORMATION

Unit 3 Coursework

Outcome 1

Analyse an existing networked information system used in an organisation, and propose physical design specifications for a new or modified networked information system.

Task type options

Task 1: In response to a given design brief a written report (including documentation of analysis and design) (40 marks)
Task 2: In response to a given design brief a labelled visual representation of a networked information system (10 marks)

A variety of design briefs were presented; however, some did not describe a *networked* information system. This typically occurred when teachers used tasks relevant to the 2000–2002 study design. The key knowledge and skills generally were well covered in the tasks with the exception of the skill of representing visually the components of the proposed networked information system. Some tasks asked students to visually represent the *existing* information system rather than the proposed one.

It is very important that a report is the task format for Task 1. Students should be informed of the scope of the report, but they should not be provided with specific questions relating to analysis and design.

Assessment

The VCAA performance descriptors were generally considered; however, some tasks did not provide adequate opportunities for students to describe the physical specifications for the new or modified system.

In most cases, the weighting scheme reflected the suggestions made in the *VCE Information Technology Assessment Handbook*.

Outcome 2

Produce a software module suitable for implementation on a portable computing device, in response to a design specification, verify its performance against this specification and explain how the program has taken into account an ethical dilemma or a legal obligation.

Task type options
Task 1: A software module, in response to a design specification (40 marks)
Task 2: A written report or a test (10 marks)

A variety of design specifications were presented, though few were original. It is important to build into the scenario the fact that the module is for use in a *mobile* computing device. The key knowledge and skills were generally well covered in the tasks; however, careful consideration needs to be given to the following areas:

- Opportunities must be given for students to design their own graphical user interface; in some instances too much direction/advice was given.
- Students should not be given too much support as it denies them the opportunities to demonstrate key knowledge and skills.

Assessment

The VCAA performance descriptors were generally considered and overall, the weightings for the various aspects of the tasks were appropriate and reflected the depth, complexity and detail required.

UNIT 4

This report is provided for the first year of implementation of this study and is based on the coursework audit and VCAA statistical data.

GENERAL COMMENTS

Unit 4 Coursework

In general most schools were able to satisfy the requirements of the *VCE Information Technology Study Design*. Exemplary tasks typically included case studies developed for this current study design, rather than being adapted from case studies used for the previous study design, and were accompanied by clear instructions.

Case studies are very important in this study design because they provide a context within which the teachers can assess the application of student knowledge and skills. Half the teachers used case studies from previous study designs and neglected to modify them to meet the requirements of the current study design.

The content and structure of a case study should provide a structural framework for writing the questions. It is also important that there is a strong relationship between the instructions to students and the assessment sheet. In a number of cases, the assessment sheet bore little resemblance to the student instructions for the task. In most of these cases, the assessment sheet had been copied from a commercial source and was not tailored to the task.

The ability for students to target specific identities in a case study or practices is important in real life and especially for the final exam. Teachers are encouraged to write questions so that the student must analyse the current situation as described in the case study. Systems analysis cannot be done with poor context.

Outcome 1

Apply the stages of software development to produce purpose-designed software that takes into account a networked information system objective and the needs of end-users.

Task type options

Task 1: Information technology solution (including internal documentation) in response to a design brief (35 marks)
Task 2: User documentation and an explanation of how purpose-designed software may cause conflict between program developers and end-users in one of the following modes: (15 marks):

- a test
- a written report.

Most tasks were either commercial publications or ones posted of the listserv <edulists.com.au> Many of them were excellent but a few had serious omissions. Teachers are responsible for checking the suitability of tasks, regardless of the source. Very few teachers wrote their own case studies.

Generally the instructions to students regarding internal documentation, validation and testing were of a high standard. Teachers in this subject have a solid understanding of programming structure and essentials and their depth and extent of their knowledge was reflected in the tasks. Especially well done was the use of algorithms and data tables. The purpose of data tables is to work out how much memory the program will use and to work out the choice of variables and when/where to declare them. Most teachers encouraged students to prepare such data tables.

Overall the tasks allowed students to demonstrate their skills in developing algorithms, data tables, validation rules, internal documentation and testing strategies. Some tasks did not allow students to demonstrate their knowledge of information system goals and objectives, due to limited information in the design brief. Without knowing the system objectives, it would have been extremely difficult for students to propose evaluation strategies and for them to discuss design and implementation consideration.

Of real concern was the complete omission in some tasks of the requirement for students to explain how purpose-designed software may cause possible conflict between the individual or organistions responsible for developing it and end-users. This key skill has been in the previous two study designs yet some teachers included no hints at all in the design brief about ethical or legal dilemmas for programmers. When the task asks students to discuss ethical and legal problems in general this then makes it difficult for the students to achieve full marks when they do not know how to tailor their response.

The task must allow students to write user documentation relevant to the design brief. While the task mode can be a test or written report it is important that students are not just required to demonstrate their knowledge in a general sense – all or most questions must relate to the design brief.

A large number of tasks did not adequately provide the opportunity for students to propose evaluation strategies to determine the extent to which the programs fulfil the information system objectives. Again, this links to adequacy of details regarding goals and objectives. This is a new skill and as such may take time for teachers to learn how to manage the teaching and assessment of it. Where teachers used design briefs and questions associated with the previous study design, this skill was omitted entirely and was a cause for concern. Again, when the system objectives are clearly delineated, it will be clear for students to manage evaluation strategies.

Assessment

The VCAA performance descriptors in the *VCE Information Technology Assessment Handbook* were generally applied with the following exceptions. Only one programming language is to be taught throughout the year. The only exception allowed here is the occasional use of javascript to expedite validation procedures in PHP. Instructions to the student should include the name of the programming language. At times it was difficult to ascertain which language, if any, was used. Many teachers sought a database development in their task and all languages support the interrogation of a database or simply via the use of external files (read, write, append). When the solution to a problem can be generated solely through the construction of a database and which has no connection to a programming language, then the students are not given the opportunity to achieve full marks within the guidelines of this outcome. Teachers are urged to explicitly state the name of the programming language so that further instructions are crystal clear.

In most cases, the assigned weighting scheme reflected the emphasis shown through the performance descriptors in the assessment handbook. In some instances, where assessment sheets were used there was a misalignment between criteria and the questions asked of the students.

Outcome 2

Propose and justify strategies for managing, developing, implementing and evaluating the introduction to an organisation of a networked information system that will operate in a global environment.

Task type options
Written report or test

Overwhelmingly teachers did a good job with this task. If previously used case studies without a networked environment were used, they were modified to include such a global network. The areas of project management, support documentation, changeover methods, testing strategies and training were particularly well covered in the tasks. Teachers must take care, however, to ensure that the tasks adequately address the areas of security and what factors influenced the acquisition of specific hardware and software components and personnel.

Assessment

The VCAA performance descriptors in the assessment handbook were generally applied. Most teachers used the weightings suggested by the assessment handbook.