This report is provided for the first year of implementation of this study and is based on the coursework audit and Victorian Curriculum and Assessment Authority (VCAA) statistical data.

UNIT 3
GENERAL COMMENTS
The majority of schools were able to develop tasks that satisfied the assessment requirements of both outcomes for this study design. In the first School-assessed Coursework audit undertaken for the reaccredited Software Development units (2011–2014), it was important to check:

- there was a connection between what was required of the students in the task and the mark allocations
- adherence to the task requirements as stated in the study design.

The structure of Outcome 1 is significantly different from any outcomes found in previous study designs, and this posed difficulties for some schools.

The structure of Outcome 2 is similar to those found in previous study designs and, in general, schools found it easier to manage this outcome.

Based on the audit submissions, there were a range of time allocations devoted to each of the outcomes. The average time allocated for Outcome 1 was 271 minutes; the average time for Outcome 2 was 327 minutes.

Much of Outcome 1 content is new to teachers including Software Requirements Specifications (SRS) reports and Use Case Diagrams. While some teachers included in their tasks SRS templates sourced from textbooks or online, others asked students to respond to questions which collectively comprised the SRS. This meant that the essential parts of the SRS were catered for, namely scope, constraints, functional requirement and non-functional requirements and Use Case Diagrams.

In previous study designs, the use of network diagrams was mandated. This is not the case in this study design. Another new topic for most teachers was the Open Systems Interconnections (OSI) model and some teachers experienced difficulties in framing relevant questions on this theory item.

Outcome 2 presented its own challenges. Foremost was the expectation to provide the student an SRS to frame the requirements of the software application. While most teachers used an SRS template, many did not. Where this occurred, some schools omitted crucial sections such as scope or non-functional requirements.

SPECIFIC INFORMATION
Outcome 1
On completion of this unit the student should be able to analyse an information problem in order to produce software requirements specifications for a solution that operates within a networked environment.
Task type: Software requirements specifications (SRS) Report

One quarter of the submitted tasks was commercially prepared ones; one third was sourced online and the remaining tasks were individually developed. Outcome 1 is strictly limited to the analysis stage of the problem-solving methodology and its associated activities.

In the previous study design project management was a significant and well understood key knowledge point. This knowledge has been easily transferred to project planning, a requirement of this study. While most of submitted tasks required identification of planning tasks associated with the creation of software solutions, it is important to note that a set list of key tasks is stated in the study design. Any content in a key knowledge point listed after ‘including’ is assessable.

Data collection methods associated with analysis has been a consistent feature of past study designs. However, in this study design, when constructing tasks it is important to require students to measure the appropriateness of those data collection methods. A number of tasks omitted this requirement. Without the justification, there is no depth to the relevant key skill and students can merely repeat what has been taught in theory classes. It is important that students apply their knowledge to the situation given in the case study.

All tasks had questions on network security vulnerabilities. There were a variety of methods to collect responses about the physical layer of the OSI and this topic was assessed almost tangentially and not directly. Some teachers asked students to prepare network diagrams, which were not necessary, as there is no related key skill.

Generally, the topic of analysis tools including data flow diagrams, context diagrams and use case diagrams was all covered well.

Teachers grasped the differences between functional requirements and non-functional requirements and set appropriate questions for students. A number of tasks required the students to supply their responses using a template. This is a new framework for the delivery of instructions and students provided appropriately structured responses as a result. An SRS is a formal business document, and the outcome statement specifies that students produce this business documentation.

The requirement to include the problem-solving methodology activity of determining the scope of software solution, and its inclusion in the SRS, was a challenge for many teachers. Typically constraints were included but not scope.

Those teachers who wrote their own tasks are to be commended. Collaboration among professionals can result in excellent tasks. When tasks were shared online, some teachers modified them to better fit their school circumstances or, where appropriate, to better fit the requirements of the study design. Teachers are reminded that it is their responsibility to ensure any task used (e.g. commercial products, work from colleagues) meets all the requirements of the relevant area of study.

For Outcome 2, the study design clearly states that an SRS should be given to students as their stimulus material. This is necessary because the outcome focuses on design and development; hence students need to be given the findings of an analysis in order to create a purpose-designed software solution. Yet, less than half of the tasks audited presented the required stimulus information in the format of an SRS. Teachers are encouraged to model their SRS on good examples found online or within textbooks.

Assessment

Nearly all teachers provided evidence of the basis for their assessment, namely the performance descriptors from the assessment handbook or their own marking scheme.
Outcome 2
On completion of this unit the student should be able to represent a software design and apply a range of functions and techniques using a programming language to develop a prototype solution to meet a specific need.

Task type: a prototype IT solution
Nearly all submitted tasks required students to consider a wide range of data types. All submitted tasks required students to create a prototype solution. Where students were supplied with an SRS, this prototype requirement provided opportunities to discuss the scope of the solution, namely its parameters. It was pleasing to see that all submitted tasks required students to create internal documentation and plan and execute testing strategies.

Data structures were not evident in approximately one-third of the tasks reviewed. The data structures could be arrays, files or records and they are clearly listed in the key knowledge.

All submitted tasks required the use of a range of software design tools: data dictionaries, data structure diagrams, object descriptions and pseudocode. It should be noted that within this study design, pseudocode is the only prescribed method to depict an algorithm. While Input-Process-Output charts, flowcharts and Nassi Schneidermann diagrams are also software design tools, they are not mandated within this study design. These tools can be taught, and then assessed internally, but from an external assessment viewpoint they are outside the boundaries of the study design.

When tasks included a well written SRS, with its sections clearly labelled and explanations supplied, students were afforded the opportunity of performing at the highest level. In some instances, where tasks did not include an SRS there was insufficient guidance for students to complete the task.

While it is possible that teachers will have taught validation in this unit, students should not be assessed on it in Outcome 2; it is a requirement in Unit 4 Outcome 1.

User interface design is also not a part of this outcome. A number of schools assessed students on their ability to design effective user interface – again this is a requirement of Unit 4 Outcome 1. If students are being assessed on knowledge and skills not related to an outcome, they are being denied time opportunities to demonstrate what is actually required of them for an outcome.

Assessment
Nearly all the submitted tasks used the VCAA performance descriptors as the basis for making assessment judgments. Where these were not used, some schools included in their marking schemes requirements from the previous study design, for example roles of personnel within an information system. The focus of this study is now the creation of software solutions that need to operate within a networked environment.

Final comments
Some teachers used as their tasks the detailed examples in the’ Advice for teachers’ section of the study design. These are incomplete, and if they are used without modification, some important key knowledge and key skills will be missing. Those schools that used the fundamental programming aspects of the detailed example for Outcome 2, and then incorporated the missing features provided an excellent task for students.
It is essential that there is an alignment between what is being asked of students in the task and the basis on which the work is being assessed. For example, the task should not include requirements for which there are no corresponding marks.

The Software Development teaching community is well-serviced through the willingness of teachers to share files and discuss study content. It is important that schools view all materials with care and match them against the study design. In particular teachers need to read the problem-solving methodology (pages 16–18 of the study design) in conjunction with the areas of study. The methodology also serves as a set of signposts for writing assessment tasks.

UNIT 4
GENERAL COMMENTS
The majority of schools set tasks that met the assessment requirements for both outcomes for the VCE Information Technology Study Design 2011–2014. As this is the first year of the new study design, the following items were under scrutiny:
• the four stages of the problem-solving methodology and its associated activities
• matching of task instructions to the marking scheme
• adherence of the task requirements to the study design.

The structure of Outcome 1 is similar to Outcome 1 in the previous study design. It requires an IT solution for Task 1 and a report or test focusing on security/legal matters for Task 2. Also included in Task 2 is user documentation. As in previous years, this outcome is worth 60% of Unit 4. The focus is on the design and development stages of the problem-solving methodology and teachers followed this well.

The content of Outcome 2 is different to the previous study design although its structure is the same. Currently the focus is entirely on recommending strategies to evaluate the quality of the solution and the efficiency and effectiveness of the network within which the solution operates.

The average time allocated to Outcome 1 was 410 minutes and the majority of schools allowed students to bring notes into class. Interestingly, if we assume that class periods are 45 minutes, the number of periods devoted to this outcome ranged from 5 to 13 periods. Half the schools used their own case study material; the others used shared or purchased commercial publications. The average time allocated for Outcome 2 was 90 minutes and only a few schools chose the report format. Two-thirds of these schools chose materials shared by other teachers or purchased commercial publications.

Regardless of the source of materials, it is the individual teacher’s responsibility to check that all tasks meet the requirements of the VCE Information Technology Study Design 2011–2014. With a new study design, it is daunting to write new materials and easier to use the work of others. Those teachers who wrote their own are to be commended.

SPECIFIC INFORMATION
Outcome 1
On completion of this unit the student should be able to apply stages of the problem-solving methodology to produce a solution for use on a mobile device, which takes into account technical and legal requirements.
Task 1

A solution (including internal documentation) in response to a design brief

The following key knowledge points were almost universally well done.

Stages of the problem-solving methodology

The relevant stages, namely design and development, were followed by nearly everyone. A formal SRS was not required in the design brief given to the students, but most teachers ensured that functional requirements, non-functional requirements, constraints and scope were presented as part of the analysis.

Types and characteristics of mobile computing devices, including PDAs, mobile phones, laptops, gaming consoles

Mobile computing devices have been in the previous study design so the assessment requirements of this content were well handled. Some teachers used the generic term ‘mobile computing devices’, while others actually named and described the capabilities and capacities of the device being used.

Methods and techniques of expressing software designs

This key knowledge was typically covered through the requirements of the use of algorithms in pseudocode, data dictionaries and interface designs. Some teachers required object descriptions where this matched the problem to be solved.

Forms and uses of data structures to organise and manipulate data, including two-dimensional arrays, stacks and queues

This key knowledge requirement augmented the level of difficulty of programming skills in this study design. For the most part, teachers chose 2D arrays but objects, stacks and queues were also well represented. Nearly all work samples indicated that this increased level of programming had been assessed thoroughly.

Techniques for searching, including binary search, and techniques for sorting, including bubble sort and quick sort

Either sorting or searching was required and this was universally met.

Techniques for checking that coded solutions meet design specifications, including construction of test data

All reviewed tasks addressed this area very well.

Task 2

User documentation and an explanation of how security and legal matters are managed when creating solutions

In Task 2, the following areas need improvement and should be considered priority areas for 2012.

Forms and types of user documentation, including printed, online Internet site (forms) and quick start guide, tutorial, content sensitive help and manual (types)

In nearly all cases, the details of this key knowledge point were not well addressed in the tasks.
It is the study design which instructs that the content must relate to the outcomes, but the assessment handbook provides some excellent advice about depth and breadth. The relevant key skill in the study design states that students should ‘use software to write appropriate user documentation’ and the assessment handbook elaborates ‘appropriate’ by suggesting it relates to the ‘intended use and audience of the software solution’.

Many teachers tended to use a blanket instruction such as ‘Create user documentation for the software’. There are, as seen above, many different types of user documentation and each can fill a specific need within the organisation. A more directed approach would allow students to differentiate among the various forms of user documentation.

*Explain how solutions have taken into account legal obligations of programmers*

This task also required students to explain security and legal matters in either a written report or a test format. One third of teachers neglected to indicate which format they were using. For audit purposes, schools must indicate the selected format, because failure to choose a task from the list in the assessment table can result in the audit submission not meeting requirements.

Generally the questions in the task relating to ‘legal obligations of programmers’ were poor. It is a requirement of the outcome that students produce a solution that takes into account legal requirements. This means that the design brief should contain clues that allow the students to focus on specific legal obligations of programmers rather than allowing them to write a nonexclusive response. The use of imprecise or generalised questions has repercussions for the final examination where a case study will be presented and students will need to interpret the context and address the issues within that context.

Part of an audit submission is detailing the basis on which assessment decisions are made. Some schools made reference to the performance descriptors in the assessment handbook; some provided a marking scheme and others provide nothing. It is vitally important that there is a direct link between the task requirements and assessment tool. For example, in some cases the marking scheme allocated marks for something that was not required of students in the actual task. It is also important that an appropriate mark allocation is given to questions within a task so that discrimination of performance can be fairly determined. An appropriate mark allocation also indicates to students a sense of the scope of the response.

**Outcome 2**

On completion of this unit the student should be able to recommend and justify strategies for evaluating the effectiveness and efficiency of solutions that operate in a networked environment.

Outcome 2 focuses on the evaluation stage of the problem-solving methodology. The topics of system support documentation, efficiency, effectiveness and training have been in past study designs, but the new study design also requires students to have knowledge of how to measure the success of a purpose-designed solution and the networked environment within which it will operate.

Several key skills caused problems in task design. It is essential that connections are made between the key knowledge and the key skills – the key skills describe how key knowledge is to be applied when demonstrating the outcome.

The set task required students to ‘describe, using technical language, the networked environments within which solutions operate’ (key skill). Most of the tasks submitted for review asked questions which required students to draw on the following key knowledge when describing the networked environments:
• technical underpinnings of intranets, the Internet and virtual private networks
• characteristics of wired and wireless networks.

There were, however, some tasks that referred to ‘information systems’ rather than ‘software solutions’, an indication that teachers were reusing case studies related to previous study designs.

The key skill ‘propose criteria and recommend techniques for testing the security of networks’ is related to the key knowledge ‘techniques for measuring the reliability and maintainability of networks, including audits, error logs and software tracking tools’. Most questions regarding the testing of security of the network were suitable; however, there were few questions focusing on the criteria for testing.

The key skill ‘justify the appropriateness of recommended strategies for evaluating the extent to which users’ needs are met’ is related to several key knowledge, one of which is ‘strategies and techniques for acquiring evaluation data about the quality of solutions’. In Unit 3 Outcome 2, students need to design evaluation criteria for their software solution, however many of those tasks asked a generic question rather than a specific one. The same problem occurred in some Unit 4 Outcome 2 tasks where students needed to provide quite specific strategies and techniques for acquiring evaluation data, but the task requirements were often general in nature.

The above key skill is also related to the key knowledge ‘types of training for the users of solutions, and techniques for measuring the suitability of training programs’. Training regimes have been included in past study designs, but in this current study students are also required to measure the suitability of them. The task must allow students to apply their knowledge of this content to the case study in order to determine the suitability of the programs.

The key knowledge ‘types of system support documentation offered to users and criteria for determining their appropriateness for users’, was a cause of concern. The audited work indicated that all teachers understood how to ask a question about system support documentation but many neglected to address how to determine the appropriateness of the documentation for the users. Even when the case study is about a small organisation, there are still stakeholders who could benefit from purposeful documentation.

The key skill ‘justify the suitability of strategies for managing ethical dilemmas’ is linked to the key knowledge ‘suitability of setting codes of ethics, imposing sanctions, education programs and the use of decision-support frameworks as strategies for managing ethical dilemmas’. This key knowledge is new to the study design while the study of ethics is not. In the material submitted for review, this key knowledge was poorly addressed. Adding a few lines to the case study with intimations or suggestions of unethical behaviour would have allowed the students to apply their knowledge of how to manage such dilemmas.