INTRODUCTION

VCE®COMPUTING: SOFTWARE DEVELOPMENT

SCHOOL ASSESSED TASK: SAT GUIDE (PART 1)

- UNIT 3 OUTCOME 2

The competed School-assessed Task (SAT) will be marked out of 80, based on a range of criteria, and will contribute 30% of the study score.

This resource relates particularly to **PART 1** of the SAT, covering the first three components:

COMPONENT 1: Project plan

COMPONENT 2: Analysis

COMPONENT 3: Folio (design)

*A second part (**PART 2**), covering the remaining two components (Software solution and a Report or Visual plan) for Outcome 1 Unit 4, is contained in Part 2 of this SAT Guide.*

The SAT covers a broad range of **key knowledge** and **key skills** across the related outcome and this SAT Guide is designed to assist you with the management of the task to address the specified criteria.

Your teacher will also provide you with essential VCAA information for this SAT.

You will have an extended period of time to complete this SAT at the discretion of your teacher.

Your teacher will advise you of any variation to these conditions.

NAME: ________________________________

TASK

Introduction

This SAT is a major aspect of your course. It involves a number of steps and requirements for you to complete.

While your teacher may provide you with materials and information (such as this SAT Guide), as a Unit 3 and 4 student it is important that you be familiar with the documents that all teachers in Victoria must use in order to assess your progress.

The key documents are publicly available on the VCAA website and include:

- Computing Study Design (all the key knowledge and key skills required, a glossary of terms and back ground information on what and why you are learning).
- Software Development Performance Descriptors for Unit 3 Outcome 1 and Unit 4 Outcome 2.
- Software Development School Based Assessment - School-Assessed Task, which includes performance descriptors which outlines, among other things, exactly what you need to show to get the highest marks possible and a checklist for teacher observations.

You should consult these documents, with the guidance of your teacher, before and during the development of this SAT.

SAT Overview

The School-Assessed Task is designed to have you work through a full project life-cycle, from idea to analysis, through design and development, with testing and implementation and finally evaluation. Throughout, you will be expected to document and maintain a project plan in the form of a Gantt chart, taking both teacher-supplied milestones and your own generated deadlines into account.

You will then develop a software solution to a “need or opportunity’, which means you can find a real-life client with an information problem that you can solve, or if you have a great idea for an app or other solution, you can develop this with real users in mind.

You will collect real data, from real clients and/or real prospective users, for a real solution. You need to devise your own need or opportunity: *your teacher cannot give it to you.*

Task Summary, Actions and Key Dates

A summary of key components required along the way is shown below. You will need to break down many of these components and add dates supplied by your teacher at certain points of action along the way. Each point of action below is described in more detail in the pages following and your teacher may give you other specific instructions, which need to be taken into account.

The criteria levels of performance can be found in the VCAA advice and indicate what you need to do for your best result.

The actions below are for Unit 3 Outcome 2 ONLY and are covered in detail in this part of the SAT guide.

Criterion	Action	Date Due
-	Write a short proposal on the need or opportunity you would like to address in your solution and why (about 100 words).	
1	Prepare a Gantt chart of the entire project life-cycle.	
2	Propose a range of data collection techniques describing advantages of chosen techniques and disadvantages of alternatives. Describe how you will manage the data you collect in order to ensure privacy of data and information.	
2	Create your final data collection tool(s) and collect the raw data.	
3	Complete a Software Requirements Specification document.	
4	Create at least 2 alternative design ideas. Create a set of evaluation criteria to be applied to the designs and your final solution.	
4	Write a short analysis of your alternate designs making a final decision on the preferred design and justifying your decision using the criteria you created (about 100-200 words and/or annotations).	
4	Complete a comprehensive set of design notes, utilising a range of design tools to fully describe your solution considering relevant design factors.	

The final actions for Unit 4 Outcome 1 are listed and will be detailed in Part 2 of the SAT Guide.

Criteria	Action	Date Due
5,6	Construct your solution.	
7	Design test data and fully test your solution.	
7	Design your useability test plan.	
7	Conduct and analyse your useability test.	
7	Make modifications to your solution from your testing.	
8	Create a set of strategies for evaluating your final solution from the evaluation criteria developed in the design stage	
8	Write an analysis of how your project plan helped you, if it required adjusting and why.	

Part A: Unit 3 Outcome 2 – Analysis and Design

At key points during the outcome, you must let your teacher observe your achievement of a milestone. These observations will be recorded by your teacher on a form provided by VCAA and referred to throughout this SAT Guide. They are identified in order below.

Component 1: Identify a need or opportunity

There are some examples of the types of needs or opportunities below, however you need to find an idea that interests YOU. You also need the capacity to find data to analyse and a project that suits the programming language you have been studying.

- Create a rapid fire arithmetic or spelling game that tracks user progress and compares against other players.
- Create program to store the membership of a team or club which allows for mail out or messaging.
- A program which calculates the costs of carpeting or painting, or other, a house including wastage and discounts from user supplied dimensions.
- A program to calculate required materials for a project such as landscape gardening.
- A program to calculate costs in a café/restaurant.
- A program for users to create playlists from their music library that meet criteria such as beats per minute for exercise, partying or relaxing.
- A program to schedule appointments for hairdressers, tennis coach, booking rooms at a conference centre including resources like projectors or tennis balls, clients' details and staff availability.

Your teacher may run a brainstorming session to allow your class to share ideas and feed off each other – but everyone must have their own, unique idea.

You will write a short proposal, describing:

- the actual need or opportunity;
- the client or potential users;
- how they might use the solution to meet their need or address the opportunity.

ACTION: Write a short proposal on the need or opportunity you would like to address in your solution and why (about 50-100 words).

VCCA Observation No. 1: Identification of need or opportunity
Student has identified a real-world problem that can be solved through a software solution.

Component 2: Project Plan – 10 marks

A Gantt chart is a visual representation of the entire project, including Unit 3 Outcome 2 (U3O2) and Unit 4 Outcome 1 (U4O1). You will act as your own project manager for the SAT, and as such you will need to control:

- time;
- tasks;
- people (this is of course mostly you);
- resources (such as hardware).

All these elements need to be included in your project plan.

2.1 Make list of all the tasks that you will need to do. Start with the Problem Solving Methodology (PSM) stages as your key headings and use this document to help you fill in the blanks. Even elements within this document will need further detail and break down.

Using the dates your teacher gives you, place a final date on the key milestones in your project

Now, next to each task try to estimate how long you think that task will take – this is the key challenge of managing projects, as there are often tasks required that you have never done before and don't know how long they will take.

Your table might look something like this:

Task Description	Duration	Resources	Due Date

2.2 Now, identify which tasks must be completed before other tasks can begin. In most cases this will be clear, especially around the key milestones.

2.3 Finally, use software to create your Gantt chart. There are many options available to you. You could use a spreadsheet – although this will require a lot of fiddling to keep it up to date and display all the features necessary. Much better to use software designed specifically for the task. Your teacher may require a specific software choice (such as Microsoft Project), or you could choose your own. A web search of online charts creators will reveal many such options, some free (like GanttProject, or Gantter for Google Docs) and some paid.

ACTION: Prepare a Gantt chart, using software, of the entire project life-cycle.

Note: The final task of the SAT in U4O1 is to analyse how this plan helped you effectively manage the project. You will probably need to make adjustments to your plan during the course of the SAT. That is OK, but you will need to keep detailed notes on how and why you needed those adjustments.

Typical reasons why project plans might change include:

- under or over estimation of the time required to complete a task;
- 'Scope creep' – caused by adding new ideas or implementing suggestions for improvements;
- reliance on availability of clients/users to gather data from (at either the analysis or testing stages);
- hardware unavailability, failures or delays;
- unforeseen personal circumstances.

Of course, there are many other possible reasons. One way to record your changes is by using a table such as the one below. An example has been provided below.

Date	Adjustment description	Reason(s)	Impact on timeline
	Increased analysis time	*Initial data collected did not cover all issues and additional data was required to fully assess need*	*Minimal (only this task) as next milestone was fixed (SRS) and contained sufficient time to complete.*

You might also keep a visual record of the changes you make to your plan by taking a copy each time you make a change. You could print a hard copy (with the date), take screenshots or save using different version numbers. Keep a record of the reasons and impacts as you go as this will make the final evaluation much easier than trying to remember later on.

VCAA Observation No. 2: Preparation of project plan
Student has prepared a Gantt chart for the entire project.

Component 3: Analysis – 20 marks

The final product of this component is the production of a Software Requirements Specification (SRS).

3.2 Analyse the current situation by collecting data

3.2.1. Identify the most appropriate method(s) of collecting data, and justify why that method is better than alternatives. You might:

- interview a range of people involved to get different perspectives;
- collect current system documentation (including manuals, inputs and outputs);
- survey current/potential users, clients etc.;
- observe people doing their current work to find inefficiencies;
- have a discussion (in person or online) to find out about how people might use such a solution or what they would expect it to contain or work.

The data you will need to collect will differ from idea to idea, as will the method of collection and the people you choose to collect it from. Think about who will use the solution – this will impact the interfaces, why they would use it, where they would use it, what alternatives does your solution compete with (such as current methods or other, similar products). Again, the list of considerations will be as varied as the ideas you generate.

Be very clear about what you need to produce following your data collection. The last thing you want is to have to go back and conduct more interviews or have people complete another questionnaire. By utilising multiple collection tools, you can get data from different points of view which you will then use to formulate your design ideas. Don’t assume that all the data you need can be collected from a simple web survey and don’t assume that everyone will WANT to complete your survey. Also, consider if your clients/users are similar to those in your class who might also have a survey to complete – lots of surveys will be boring and your responses might not be as useful.

You will need to consider the privacy laws as they apply to your solution including taking necessary steps to protect the individual’s rights and the data itself.

ACTION: Propose a range of data collection techniques describing advantages of chosen techniques and disadvantages of alternatives and outlining how your methods will enable you to analyse the need or opportunity.

Describe how you will manage the data you collect in order to ensure privacy of data and information.

VCAA Observation No. 3: Data collection
Student identifies data sources and collection methods for analysis.

3.2.2. After getting feedback on your data collection method(s) and/or tool(s) (from your teacher or others) create the final version of the collection tool(s) and actually collect the data, *taking into account your privacy and data management strategies.*

ACTION: Create your final data collection tool(s) and collect the raw data.

3.2.3. From your data collection, analyse the need or opportunity and document the proposed solution by outlining the following.

- Purpose and audience of the Software Requirements Specifications (SRS).
- Stakeholders of the new solution
    - Identify any relevant organisations and their structure
    - User characteristics of the eventual users.
- Environment characteristics
    - Describe and justify the chosen application architecture (mobile, rich client, peer-to-peer and internet applications) and document the network requirements (using network diagrams where appropriate).
- Requirements of the new solution
    - An introductory comment or “statement of purpose” for the solution. What need or opportunity will the solution meet?
    - Functional requirements
        - Describe the way your solution will manage data input, transformation, storage and output in the form of a context diagram and data-flow diagrams
        - Describe the interactions between your solution and its users and other system in the form of a use case diagram or diagrams.
    - Non-Functional requirements
        - Describe the key attributes the solution should possess such as useability, reliability, portability, robustness, maintainability; be sure to link these to characteristics of the users/potential users themselves and/or the operating environment where appropriate.
- Constraints imposed on the new solution or its development, such as:
    - economic, legal, social, technical and useability factors;
    - vulnerability to security threats;
    - data protection and authentication;
    - accessibility.
- Scope of the new solution
    - Consideration of features that may not be feasible or essential to the implementation of the proposed solution and which may provide opportunities for further development.

ACTION: Complete a Software Requirements Specification (SRS) document using the key dot points as subheadings in your report. You may include the diagrams within the text, or as an appendix to the document. Either way, ensure they are labelled appropriately.

VCAA Observation No. 4: Development of software requirements specification
Student has documented the analysis in the form of an SRS.

Component 4: Design Folio – 10 marks

The final product of this task is a set of design notes ready for passing to the developer (i.e. you in the next task) and a set of criteria to evaluate the solution. Tasks 2.1 and 2.2 could be completed in either order, or even be worked on concurrently.

4.1 Design generation

You must produce several (at least two) different alternative design ideas for your solution. The difference could be alternate user interface layout, colour schemes, functionality, work flow, algorithms, strategies or other factors. These differences should be "distinctive, feasible and original" and therefore the result of significant thought and effort – not just swapping colours. What you present to your teacher needs to demonstrate an understanding of the design process and use of design tools, including annotations.

Consideration should be given to useability, affordability, security, interoperability and marketability of your solution, again with the users and operating environment in mind. In other words, all aspects of the solution need to be covered in the design ideas. However, note every alternative need to address ALL aspects of the solution. You might have three different user interface (UI) alternatives, but only 2 data storage alternatives. These design ideas do not need to be detailed or fully developed, rather they are supposed to allow you the opportunity to think broadly to seek the best solution for your circumstances.

There is a range of tools that are suitable for you to use: which tools are most appropriate will depend on your specific task. These tools are described in more detail in section 2.3.

ACTION: Create at least 2 original, distinctive design ideas for your solution.

4.2. Design selection and evaluation criteria

4.2.1. Develop a set of criteria against which the alternate designs can be measured, the preferred design can be chosen and the final solution can be evaluated, in terms of efficiency and effectiveness.

You should consider the data collected from clients and/or potential users as to preferences and possibilities. Efficiencies might include load times, data transfer times, data entry times, processing times, reduction in paper required, requirements of storage, multiple-screens vs single long screen, suitability for mobile and desktop use (therefore reducing code duplication), costs associated with different solutions for hardware or infrastructure (for example a web and/or data hosting solution)

Effectiveness measures might include completeness, readability, security, attractiveness, clarity, accuracy, accessibility, timeliness, communication of message, relevance and useability.

With these considerations, you should be able to devise a set of criteria.

ACTION: Create a set of evaluation criteria (in terms of efficiency and effectiveness) for your design selection and product evaluation.

4.2.2 Using your criteria prepare a short report explaining why your preferred design was the best choice. You should include why the alternate design did not meet the criteria, or did not meet them as completely.

One way of using the criteria to evaluate is to create a weighted criteria decision matrix.

1. List all the criteria in the first column of a table.
2. Determine the relative importance of each criteria against the others (a weight scale of 1-5 where 5 is most important and 1 is least important).
3. Then for each design give a score - again use a scale of 1-5.
4. Multiply each criteria weight by the score in each design alternative.
5. Whichever design has the highest tally met the criteria more completely.

Criteria	**Weight**	**Design 1**		**Design 2**	
	(1-5)	Score	Total	Score	Total
1	5	3	5x3=15	4	5x4=20
2	5	5	25	4	20
3	1	5	5	3	3
4	2	3	6	3	6
Sum			51		49

In this case, Design 1 appears the better option.

Of course, it may be that a combination of the alternative designs is, in fact, the best solution. By having a comprehensive set of criteria you should be able to articulate how you reached your decision.

ACTION: Report on your preferred design selection.

4.3 Solution design

4.3.1 Now you will prepare the way for your development during PART 2 of the task.

There is a range of design tools available for you to use including those listed below. Your teacher will provide you with more detail about the use of these tools and may help you select those that are most appropriate for your circumstances.

Any comprehensive solution design will involve the use of multiple tools to accurately describe the solution.

The more detailed and precise your design work is at this stage, the more time you will save in the production of your solution. As you are designing, try to imagine actually creating the solution and, from your knowledge of the programming tools you will use, predict challenges you will face and plan around them.

A solution design will typically involve:

- Identifying what specific data is required and how the data will be named, structured, validated and manipulated. There are a number of tools you can use to do this – you may need to use more than one to fully describe your solution.
    - Algorithm representation (pseudocode) –written representation of the logical sequence of processing steps required for the solution. This should be used to describe the total solution, broken into logical sections or modules.
    - Data dictionaries – list all the variables and data objects, their data type, purpose and scope.
    - Object descriptions – similar to data dictionaries, but also include details of events and methods relevant to that object.
    - Other tools could be used in addition to these such as IPO Charts, data structure diagrams and Flow Charts.
- Identifying how various components of a solution relate to one another. Design considerations such as useability, including robustness, flexibility and ease of use, and accessibility, including navigation should be factored into the design notes. Different solution types will necessary require different tools such as:
    - storyboards, site maps/screen relationships or user flow diagrams;
    - queries, forms and reports relationships for data driven solutions (note this does not only include the use of formal databases);
    - hierarchy charts of modules, procedures and objects;
    - data flow diagrams can be used as a design tools as well as an analysis tool.
- Determining the appearance, position and size of text, images and graphics, font types, colours and other enhancements in the user interface. Design principles such as alignment, repetition, contrast, space and balance should be considered. These can be represented using tools such as
    - Annotated diagrams/mock-ups
    - Layout diagrams or wireframes

While there are a lot of tools shown here, you will not be expected to use them all. Rather, part of the assessment for this section is your ability to select **the appropriate tool for the appropriate purpose** and then to **use that tool correctly**.

The final goal of the design stage is to provide sufficient detail so that *another* developer could take your notes and create your solution without having to ask you many, if any, clarifying questions.

ACTION: Complete a comprehensive set of design notes, using a range of design tools to fully describe your solution considering relevant design factors

VCAA Observation No. 5: Folio
Student has developed the folio of design ideas, criteria and detailed preferred design.

Teacher Advice

General

This SAT guide (PART 1) relates mainly to Criteria 1-4. A second part will be provided later as a separate document. The intention is to compliment VCAA SAT requirements and assist teacher and student management of the task.

VCAA documentation

It is essential to use this SAT Guide in conjunction with the SAT information provided by VCAA. Specifically, the Performance Indicators for each criterion, authentication record form and assessment sheet, as well as authentication advice which can be found on the VCAA website.

Authentication

Authentication is a challenge when students do not complete all work in test conditions. Therefore, accurate, continual and consistent record keeping is vital for teachers to be able to reliably assess the students' progress and final product.

Within the context of ongoing lessons to cover course content, teachers should meet regularly one-on-one with their students to sight progress, provide feedback and take copies of student work along the way.

The VCAA Authentication Record Form found in the SBA document provides the official documentation you must provide. In addition, teachers might take multiple copies of student work ensuring no large leaps in quality or changes of style are detected that might indicate undue assistance. This is particularly challenging during the development stage.

Teachers should provide regular class time for students to work on their SAT in class. That way you will be able to see (and authenticate) students actually making progress on their work, rather than just the product of a week or so on their own. Be sure that students know why they are getting class time (to check progress), and give them advance warning so they can ensure they have everything they need to work on the task.

Students would not be able to complete the task only using class time and should be encouraged to use previous work and targeted web searches for assistance. As a teacher, you must be careful to not provide undue assistance. However, once an element has been submitted and assessed you could provide information to help students improve that section, in order to ensure no follow-on marks are lost. Students cannot get additional marks for re-doing a section after teacher assessment, though.

Using the criteria

When marking the criteria, it is important that teachers are careful to ensure students meet the lowest levels of performance. In some cases, an element listed in the lower levels is not repeated in higher levels – so if a student does not satisfy that requirement, teachers need to consider if a student would still qualify for a higher level of achievement for that criteria.

Each criterion is scored out of 10, and students' scores should be recorded on a personal assessment sheet found in the SBA document. Each criterion is equally weighted, but does not always represent an equal weighting of time and student effort. Some criteria are discreet in nature (e.g. #1), some may represent effort across a number of tasks (e.g. #3).

Teachers should consider the merits of providing students with both the Performance Criteria as published by VCAA and the teacher notes of this SAT Guide so they are fully briefed on expectations.

You may select a programming language, or allow students to choose their own. Either way, be sure the language selected meets the VCAA guidelines published each year.

Marking Advice

The following table will assist with teacher marking of Criteria 1-4 of the SAT. A similar table will be provided in PART 2 for Criteria 5-8.

Criteria	Focus	Marking Advice
1. *Understanding of project management concepts and processes*	Constructing a Gantt chart using software with all relevant concepts and process Documenting all stages of the problem solving methodology and identifying all required tasks. Creating an accurate, complete and coherent chart.	Students should use this guide to create their project plan. The stages of the PSM provide a good starting point and some detail about what is required in each stage for the SAT is important. One key challenge is that students will be trying to estimate the time and resources required to complete tasks about which they have little or no knowledge (consider the SRS for example). If they use your due dates, and sufficiently break down the stages into manageable and appropriate subtasks with milestones and dependencies they should score well. The use of the particular software selected is assessed here and they should be able to demonstrate the creation of activities, milestones and dependencies. Note the highest level of performance states "accuracy, completeness and coherence" – these are the benchmarks.
2. *Skills in using analytical tools to depict relationships between data, users and digital systems*	Identifying and collecting data using a using a range of methods. Identifying features of the analytical tools. Making connections between data, users and systems'. Correct use of the analytical tools.	Students have a wide range of options when collecting data, such as observation, questionnaire, interview, document collection/analysis, comparable product analysis, focus group, vox pop. However, students tend to rely overly on online questionnaire tools (like Google Forms or SurveyMonkey). Encourage them to actually speak one-on-one with their clients (if they have one) or search far and wide for potential users of the "opportunity". They should be able to demonstrate use of open/closed questions, Leichardt scales, language and questions appropriate for audience, and you should consider the range of relevant people selected to actually complete their questions. They should identify network hardware and software and/or device specifications as appropriate for their proposed need or opportunity. In assessing the diagrams, there are some considerations: • Does the diagram accurately depict the situation as described in the other documentation? • Has the student over-simplified the situation in order to draw a simpler diagram? • Finally, does it follow the rules for the particular diagram?

		For UCD: has system boundaries, actors (both users and other systems if appropriate), uses cases, connectors, <<includes>>, <<extends>> Context shows input and outputs of system and key external entities. DFD (probably to level 2) with storage, data flows, entities and processes. In using the tools, have all the connections be made correctly (i.e. mixing up <<extends>> and <<includes>>. No illegal data flows. Formats and conventions followed.
3. *Interpretation of data to identify and document the software requirements specification.*	Drawing conclusions from the data, as depicted in the analytical tools. Describes functional and non-functional requirements, constraints and scope of the solution. Describes the technical (network/hardware) and organisational (personal/ strategic/ business) aspects of the solution. Collation of SRS.	Does the SRS describe the purpose of the solution and its ability to achieve the goal or meet the need or opportunity? While there is no SRS standard, in the Advice for Teachers, VCAA provides a list of elements that must be included in an SRS – the purpose and audience of the SRS, user characteristics (general characteristics of eventual users), environment characteristics (technical description of environment within which the solution will operate), the scope of the solution, functional requirements, non-functional requirements, constraints, context diagrams, data flow diagrams, use case diagrams. This is reflected in this guide and teachers should assess that all components exist – although the order is not important. Some elements could be a requirement and a scope consideration and a constraint. For example a functional requirement to run on iOS would also be a constraint in that it must run on iOS which limits the possible programming language (excludes VB.Net for example) and also part of the scope in that it does not need to run on other OS's. All requirements, scope and constraints need to be specific to this solution. General comments like "must be easy to use" will not score highly – rather, consider the characteristics of the users, their needs and the particular features that must be included to meet them. Does the SRS actually provide sufficient and accurate detail to complete the design stage without error or omission?
3. *Skills in generating design ideas and designing preferred solutions.*	Creating evaluation criteria for design selection and solution evaluation. Generation of at least two distinct design ideas.	The key words here are "distinctive", "feasible" and "original" and the design ideas must be about more "functionality and appearance" – so not just UI. The performance indicators suggest 2-3 ideas, but these don't need to be completely worked, rather different components of the

	Selection of preferred design based on criteria. Completely describing the solution design using a range of tools with consideration of relevant factors.	total design might have multiple options. Students need to demonstrate they have thought broadly in their designing. They need to provide alternatives that would make sense (not silly alternatives that would not be considered). Students may find this challenging, as they have a clear picture in their mind about “their” way. That is fine, but in order to satisfy this criterion they must be able to demonstrate alternative idea generation and systematic selection via criteria. Evaluation criteria need to address a wide range of circumstances. While good criteria could adequately assist in selecting from the given designs, exemplary criteria could be applied to any number of designs and still provide the most complete design or solution. They must cover both effectiveness and efficiency and should address all the functional and non-functional requirements, the scope and the constraints. They should cover the legal, social and technical considerations. This criterion is assessing the students’ ability to correctly use all those great criteria. One simple way of doing this is by creating a decision matrix. If they choose NOT to use the matrix, you must be satisfied that the criteria were used to make the final design decision. Finally, the students’ ability to use all the design tools is assessed. This is a large section. There are many tools. In all things, students should pretend that *someone else* will be doing the actual development and write their design for them. What questions would they need to ask? Hopefully, none! Consider the appropriateness of the selected tools, consider the completeness of the final design, and consider the accuracy and skill with which the student has used their chosen tools (even if they choose a poor one).