Hi all,

A few teachers have emailed me directly (especially those new to the subject), asking what we covered in the January meeting I held last week for SD teachers. I’ve tried to recall and write some possibly useful notes below. Take/leave/disagree with whatever you like. Sorry for the length!

We chatted mostly about challenges experienced running the course for the first time last year, some of which are resolved due to actually having the SAT Assessment Criteria now. For others we shared how we’d taught particular sections. The use of templates to guide and speed student responses was noted (eg. for user evaluation strategies, solution evaluation, etc.)

**U3O1**   
we compared the task breakdown… the scope of each task seemed highly dependent upon the length of class periods at each school (75min to 120mins). My periods are shorter, hence I broke U3O1 into six tasks. We all agreed that each folio assessment should be completed within a single period for authentication purposes etc.

Ensure that you provide the design tools for students (eg. IPO, data dictionary, pseudocode/flowchart, maybe a template for testing & trace tables, images/csv files/forms required for development. Bonus – this speeds up & focuses students on development, as well as familiarising them with the tools.)

We all gave different weighting to diff tasks.

I used a modified version of this assessment rubric from digipubs for each task, incorporating specific abilities they needed to demo: <http://deecddigipubs2014.businesscatalyst.com/vcaa/vce-computing/vce-computing-units-3-and-4-software-dev-unit-3-performance>

Rough overview of the tasks I’m planning to run this year (let me know if I missed something!). Note that everyone was doing something a little different, no problem with that!

1. IF, CASE, numeric calculations
2. Loops, Trace table
3. Function, CSV file read, Array
4. CSV file write and read, display to page
5. Linear search, trace table, loop
6. Data sort and display, Binary search

All require a testing table, quality internal documentation.

**U3O2:**   
Most have already started students on finding a potential need and client. Would guide students away from choosing an opportunity unless they are excellent students, due to the difficulty of gathering real data to form the basis of the SRS analysis.

**Updated SAT Criteria**If you missed Claudia’s email – the 2017 SAT Criteria have been released. Note particularly that Criteria 7 has been divided more evenly amongst Crit 7 and 8, with Crit 7 now focusing on Testing and Crit 8 on Evaluation. Wording changes have been made throughout to clarify requirements. Some criteria have been swapped in order.  <http://www.vcaa.vic.edu.au/Documents/vce/computing/SBA_Software_Development.pdf>

Apart from the criteria themselves being tweaked and updated, additional information and clarification to guide teachers, has been added to the descriptions on pages 1-5, so that’s worth reading carefully.

**The Authentication Sheet**   
is required to be kept during the SAT for each student and stored securely. However, I believe that you can make changes to it as you see fit (adding Observations / reordering them) as long as you still have the required comments/date/signature columns and bits at the top & bottom. The order of Observations as it stands though is pretty logical.

Last year I used authentications as checkpoints, setting dates for students to have achieve these by. Kept them on track. (A bit of healthy competition to hit all the checkpoints on time doesn’t hurt either.)

**Student Gantt and recorded progress:**We took various approaches to this, but would agree that a regular journal, a printed Gantt, sticky notes and comments on the printed copy, are most useful. I will require my students to print their Gantt at school so I’ve actually sighted it, then they can either take it home or stick on the classroom wall to note changes.  
Some students preferred to make versions of their Gantt, however later comparing and assessing multiple files was a pain.

**My Timeline**

I’ve added my planned timeline to this email, we are all doing something similar. Each school’s terms may be a little different in length. While there is a solid week for theory in May, I’ll also scatter theory lessons through U4O2.

The reason for splitting U3O1 is to

a) get students well started on the SAT, so they can utilise Term 1 holidays

b) get the students’ heads back into coding before starting SAT part 2, and

c) give yourself 2 weeks to mark the SAT Part 1 and return to students.

Note that while students can then make changes to any glaring/major issues in their SRS/designs, to assist themselves in Part 2, any changes would NOT be remarked.

**Free Poster**

Also attached a poster I made on the conceptual layers of software… feedback on wording would be helpful. Source if you’re wondering: <http://www.vcaa.vic.edu.au/Documents/vce/computing/SofDevProgrammingRequirements.docx>

Hope you all have a smooth and positive start to the year!

Kind regards,

**Lisa Kerekes**Secondary Teacher (Computing, Media)