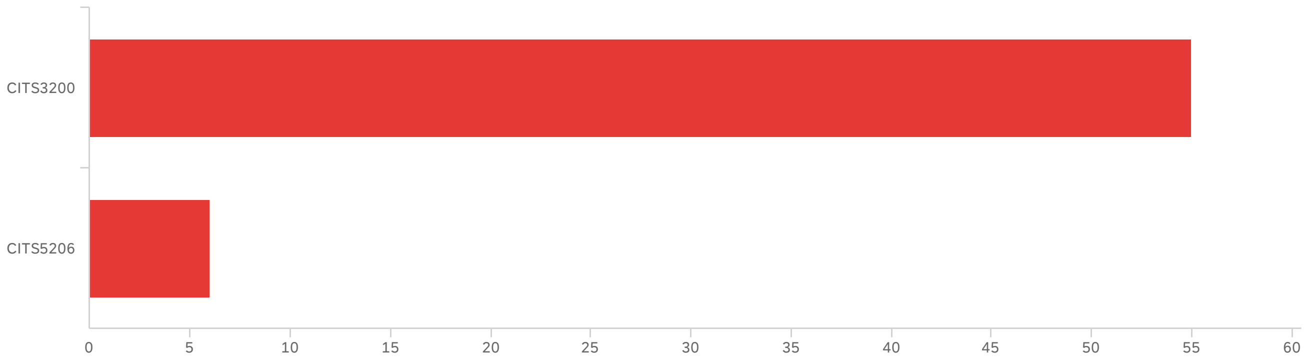
Default Report

CITS3200/CITS5206 2020 Survey

September 28, 2020 7:41 AM MDT

Q1 - Which class are you in:

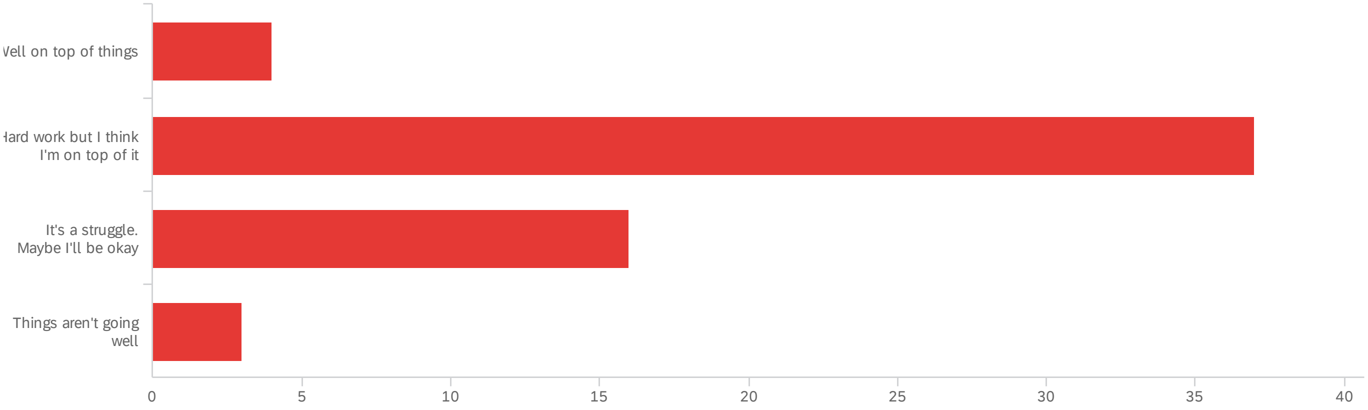


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Which class are you in:	1.00	2.00	1.10	0.30	0.09	61

#	Field	Choice Count	
1	CITS3200	90.16%	55
2	CITS5206	9.84%	6
			61

Showing rows 1 - 3 of 3

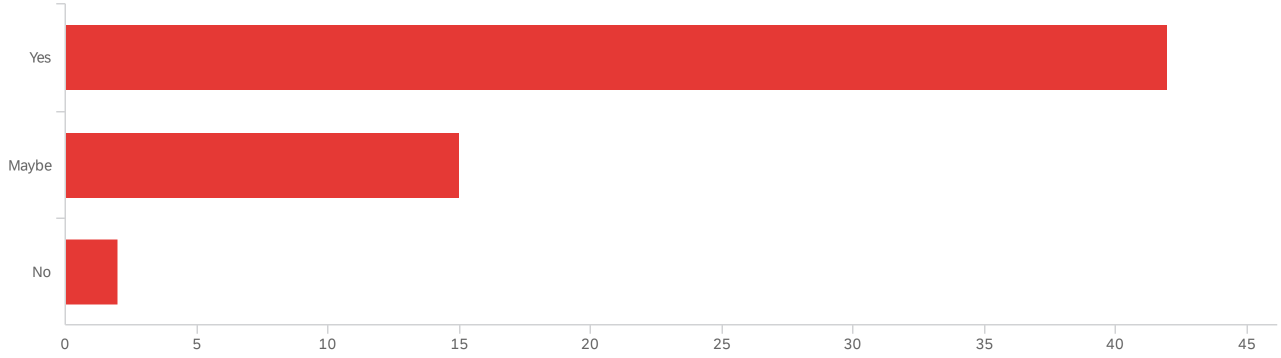
Q2 - How do you feel you are tracking in CITS3200/CITS5206



#	Field	Choice Count	
1	Well on top of things	6.67%	4
2	Hard work but I think I'm on top of it	61.67%	37
3	It's a struggle. Maybe I'll be okay	26.67%	16
4	Things aren't going well	5.00%	3
			60

Showing rows 1 - 5 of 5

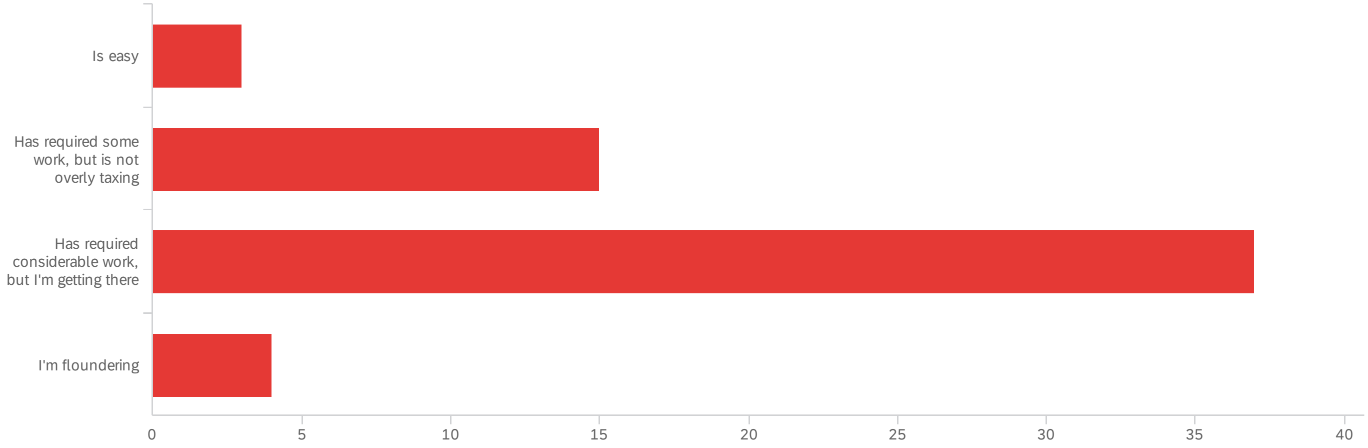
Q3 - Thinking about the Project. The idea behind the Project - unlike the lab exercises and assignment you've done in other units - is that this is where you put together what you have learnt and get the experience of different roles in a project team. Are those design goals reasonable?



#	Field	Choice Count	
1	Yes	71.19%	42
2	Maybe	25.42%	15
3	No	3.39%	2
			59

Showing rows 1 - 4 of 4

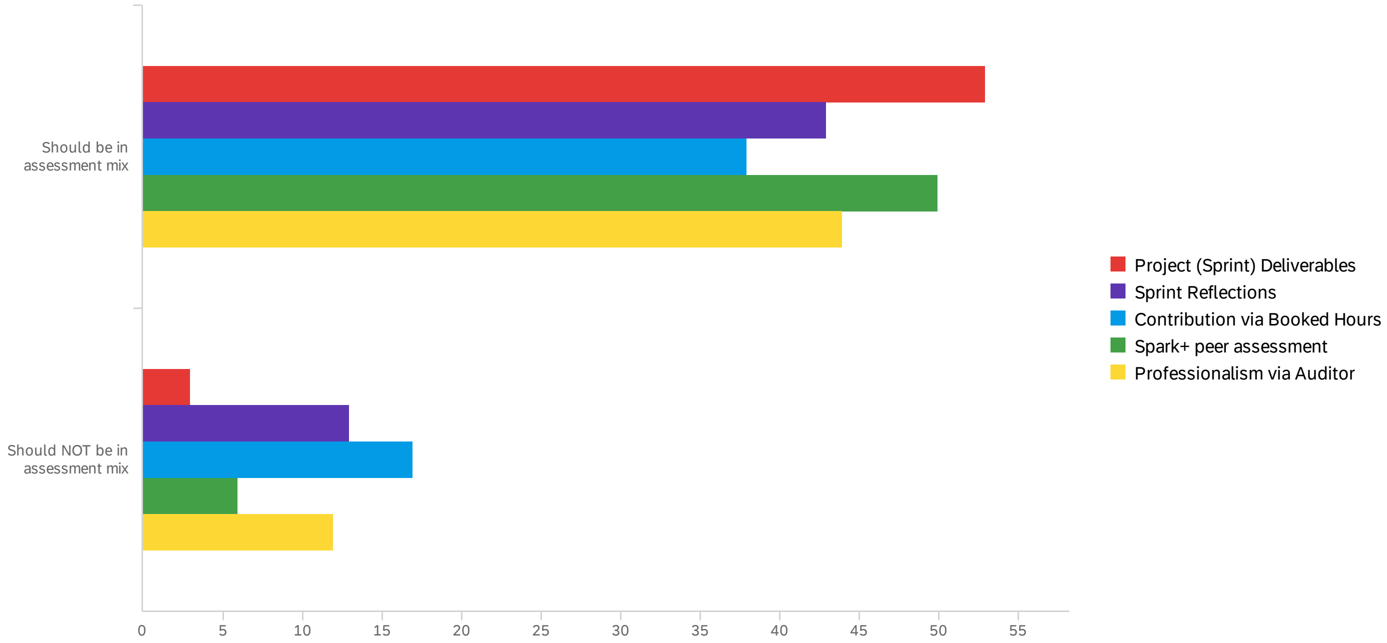
Q4 - So far, I'm finding the Project



#	Field	Choice Count	
1	Is easy	5.08%	3
2	Has required some work, but is not overly taxing	25.42%	15
3	Has required considerable work, but I'm getting there	62.71%	37
4	I'm floundering	6.78%	4
			59

Showing rows 1 - 5 of 5

Q6 - No assessment metric is ever suffiicient by itself. Each may be okay at measuring on aspect of the unit, but may have holes that, hopefully, are patched by a different metric. In other words, there has to be a mix of types of assessment. For each of the following assessment items that are currently in the mix, should they, or should they not, be in the assement mix.



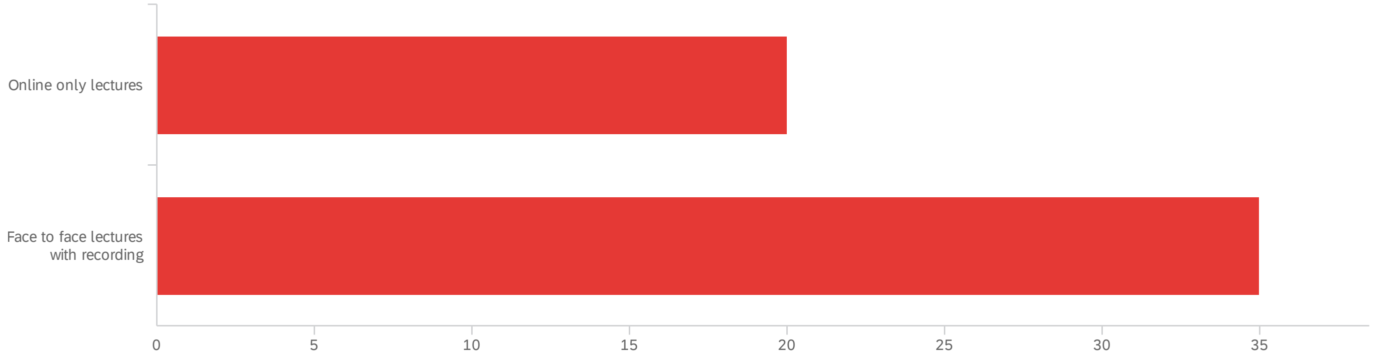
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Project (Sprint) Deliverables	1.00	2.00	1.05	0.23	0.05	56
2	Sprint Reflections	1.00	2.00	1.23	0.42	0.18	56
3	Contribution via Booked Hours	1.00	2.00	1.31	0.46	0.21	55
4	Spark+ peer assessment	1.00	2.00	1.11	0.31	0.10	56
5	Professionalism via Auditor	1.00	2.00	1.21	0.41	0.17	56

#	Field	Should be in assessment mix	Should NOT be in assessment mix	Total

#	Field	Should be in assessment mix		Should NOT be in assessment mix		Total
1	Project (Sprint) Deliverables	94.64%	53	5.36%	3	56
2	Sprint Reflections	76.79%	43	23.21%	13	56
3	Contribution via Booked Hours	69.09%	38	30.91%	17	55
4	Spark+ peer assessment	89.29%	50	10.71%	6	56
5	Professionalism via Auditor	78.57%	44	21.43%	12	56

Showing rows 1 - 5 of 5

Q13 - The big change this year was the move to an almost entire online format. Look first at lectures do you prefer

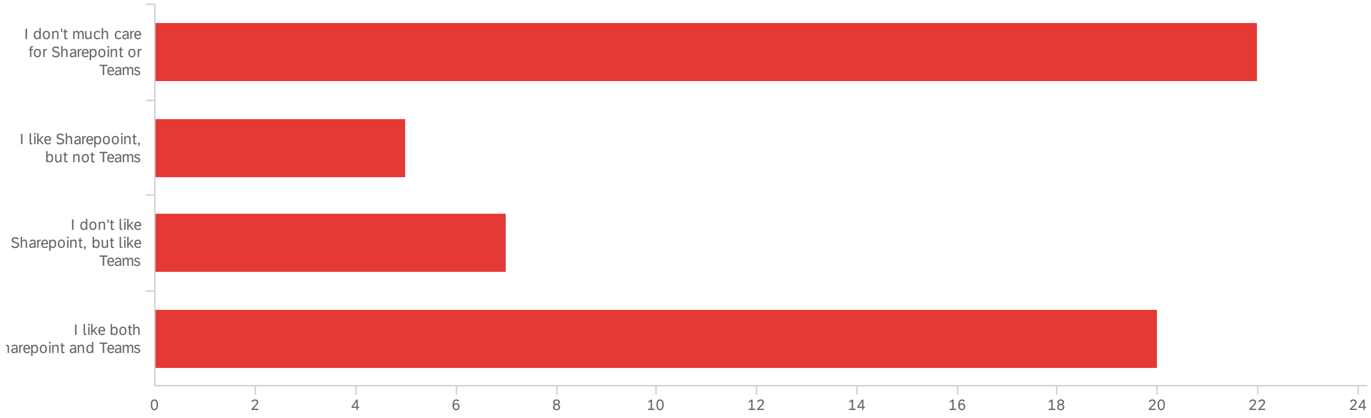


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	The big change this year was the move to an almost entire online format. Look first at lectures do you prefer	1.00	2.00	1.64	0.48	0.23	55

#	Field	Choice Count	
1	Online only lectures	36.36%	20
2	Face to face lectures with recording	63.64%	35
			55

Showing rows 1 - 3 of 3

Q14 - This year also saw the introduction of MS Teams, both for holding meetings and also for sharing files (via Sharepoint). Please choose one of:



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	This year also saw the introduction of MS Teams, both for holding meetings and also for sharing files (via Sharepoint). Please choose one of:	1.00	4.00	2.46	1.34	1.80	54

#	Field	Choice Count	
1	I don't much care for Sharepoint or Teams	40.74%	22
2	I like Sharepooint, but not Teams	9.26%	5
3	I don't like Sharepoint, but like Teams	12.96%	7
4	I like both Sharepoint and Teams	37.04%	20
			54

Showing rows 1 - 5 of 5

Q7 - What have you found to be good/useful about the unit?

What have you found to be good/useful about the unit?
The experience working on a project from start to finish, as well as working for a client has been very valuable.
The unit provides a good opportunity to take ownership of a real software product for a real client. It provides you with a lot of freedom in how you architect and implement the final product.
Learnt the reality of teams not working as planned
I liked the relative freedom in picking your project and the ability to maintain your own schedule.
some of the lectures are good, copyright,estimation etc freedom about technology to choose random assigned group simulates work environment pretty well
Get the opportunity to learn new skills.
The project concept and ideas behind it are cool and useful, being thrown into the deep end with random people and being force to "make it work" is a frustrating but very useful practical skill. Of all parts e.g. marked items e.g. sprint work, reflections, the portfolio and information on such as roles of auditor, team members and client I feel are overtly convoluted in how they are described and what they are ultimately are and need to be much more clearer and to the point.(far more so for the marked items but same applies to roles). The only downside of the unit is that it feels like most of the lectures and there content are only there because the unit requires lectures, most of the content feels useless and pointless outside of cramming it for the portfolio. However in most lectures there are snippets of very useful golden information useful for the outside industry. All in all a very useful unit but a lot of the unit content feels unnecessary.
Nothing. It's just one of those time-waster units that you are forced to work through – and everyone knows it (although they don't say it). Please leave us alone to actually learn something, rather than charging us money for nonsense units.
Haven't got to experience teamwork in this way before, so I've been finding it a good learning experience
The first real look into the different problems and stresses involved in a real life project that can be more open ended and time consuming
The experience has is a good example of real client work.
Good layout. Helpful Unit Webpage and help forum. Organised.
Working in bigger teams on a Computer Science project is something new for me which I appreciate in the unit. I enjoyed the lectures which were a part of the unit but would appreciate face-2-face lectures than online-only versions.
Rich experience and communication with team members
Everything aside from the lectures.
I have found the long term project good, it means that i am not just waiting for a deadline but continuously trying to improve how we work together.
practicality
The professional environment

What have you found to be good/useful about the unit?

- I have a really good team - I like that the project is structured around the sprints as it helps to stay on top of things - I have learned a lot about teamwork - It's refreshing to have a unit that reflects the "real world" work process

Time management, I have been having serious issue on time management for a long time, It has not been addressed but I do believe It is getting better due to something I have learned from this unit.

Good experience of teamwork and web development

It's difficult for team member to have a face to face meeting because of we study online, don't need to go campus. As a result, the chance to see team-mates, teacher and client is reduced. It may reduce our work efficiency to some extents, even though we can communicate online. Then we have to spend more time in each units, I feel that compared to last year in school, I spend more time studying every day, a little tired, but acceptable. I have learned a lot during this semester, especially when I participated in this unit CITS5206.

Time management, teamwork and communication skills, putting together learning and skills to work with others, having a flexible project without a rigid approach

By completing this unit we will leave UWA job-ready

- The self managed team has so far run very smoothly, and is one of the most organised teams I have had in uni projects so far. This is mostly due to how the teams were made, providing a diverse team of varying skills and experiences. - The client has been very open and accommodating, which makes it easier for us to make decisions and keep moving forwards with the project. - The professionalism aspect of this unit is a useful and also unique experience. - The template for the whole team booked hours is provided, which otherwise we would have no idea where to start. - Michael has quickly responded to help forum posts, which is extremely helpful.

I feel like I'm getting a lot of experience on how to communicate with the client. I am also learning a lot about project development and how to conduct yourself professionally, which I find useful. Whilst this unit is daunting, I know/like that it's seems to mimic a workplace environment which will be better for me in the long run.

The unique experience. I wish my other major have it as well.

The ability to work on an industry-level project in a team. Such experience will be useful for the workforce.

It is completely different to other units and seems to be industry relevant

Some of the discussion around professionalism

Experience in gathering requirements and real proper planning of a project has been a good experience. Being able to work freely with technologies is cool.

This unit is very well run and I think really gives us, as students, a good idea of what lies ahead in our future careers

- practical knowledge - have an overview of many roles in projects - gain some knowledge in other fields

The weekly deadline

Having a real client is really rewarding.

Some of the lectures have had useful content that has been applicable to the project management process

Actually meeting and interacting with strangers

What have you found to be good/useful about the unit?
Has definitely expanded my understanding of teamwork and working on bigger projects
It gives some real-world experience and has given me the opportunity to learn things like using Google Cloud Platform which I wouldn't have learnt otherwise.
Literally nothing, it's goals may be good, but it has failed in actually achieving those goals, the project is not challenging me to use what I have learned, it's just boring web-dev (which in general is a terrible area of CS imo) and the group part has failed to insight any kind of learning in that matter, the only thing I got from being the PM is how terrible I am at that role and a bunch of stress.
Can do a real project like professionals.
Have liked learning react and node for our project
The mentor is useful, the help forum has also been helpful
Learning and developing a real world software project.

Q8 - Are there things that can be improved, and if so, what?

Are there things that can be improved, and if so, what?

There should only be one place that we submit (excluding sending documents to the client). For example, our weekly time sheets and minutes should be emailed to the auditor OR uploaded to teams - not both. The group spreadsheet is very confusing and has not helped us plan our project, as we were never really sure we were using it correctly. If this had a more logical format I think it could have been useful in managing our time.

The algorithm to make is likely very good, i think i maybe just got unlucky

Yes, whilst I can appreciate the Professional Development Portfolio encouraging students to watch the lectures, it becomes more about a grind for the mark as opposed to a learning experience. Students should also be told to round their timesheets to closest 15 minutes. It has been a pain to compile the weekly hours with some of my team members having x15 five minute entries per week. I have also found the weekly timesheet a hassle to compile and not that useful. Perhaps a different way of logging hours would make more sense at least for the weekly compilation. The current timesheet has a weird format and not all that clear to understand and interpret. I feel that there should also be a lecture on cloud frameworks since most projects seem to require hosting. Also having a lecture on using github workspaces for sprint planning would be nice.

some lectures repeated themselves and are not as structured the group timesheet is so hard to understand please modify please for future student could possibly get auditor to get file from cssubmit, potentially reduce the need to submit file multiple times the unit website sometimes can be a bit tricky to navigate through when trying to find a certain piece of information

It would be nice to have some updates on how our professionalism mark is tracking. I would like to know what things I'm doing well and what things I need to improve on. It would be great if the auditor could give us this personalised feedback, though I understand it is non-trivial and time-consuming.

As said in previous answer. 1. A bit more clearer and to the point of everything that is required so it isn't a confusing start 2. Trim most of the lecture material down to not include unnecessary, (and sometimes) pseudo science professional skills(e.g. in the project management where the lecturer weirdly tries to sell a pen).

If you want to have a "professional" unit that actually benefits students rather than acts as another worthless "busywork" unit, consider turning this unit into a type of industry-based learning internship unit (even if it's unpaid); other universities already do this.

My team found some of the Agile terminology a bit confusing, so perhaps a lecture going through that could be useful

Perhaps less of a fear factor to motivate staying on top of things from an assessment point of view and instead make auditor/mentor meetings more consistent and perhaps make attendance required. Also having the reviews of the lectures be due earlier and one at a time would ensure all members have got across the content in a timely matter.

I think the unit should have more strict marking guides and templates that allow students to know where and how to get the marks within the assessments.

Maybe a bit too much work early on in semester. The lectures combined with project take up a tremendous amount of time. Also wait until week 2/3 to assign project teams. Have had 3 people pull out even after reassignment.

I believe that it would be better if the students are better matched on the projects based on the background experiences and what they are studying because the students doing this unit now are not majorly Computer Science ones sinch UWA has also introduced Data Science a couple of years ago. I believe that the current timesheet structure can be improved a lot and maybe it would be worth checking out dedicated software for that such as Atlassian Jira which are used in industry.

Client communication is something that I have found to be lacking. It is emphasised to have good client communication in a few of the lectures, but specifically what that entails is never detailed. A lot of my project issues have come from a lack of good client communication and because we want all members in my group to have a go at the role, I have noticed that there is some confusion. As I have had a fair amount of client handling roles in the past, I also notice that not enough emphasis is placed on making sure the client understands what we are doing, and that we understand what they want from us. Perhaps make a compulsory (but non-assessed) quiz on what we could do for good client communication before we start going full speed ahead with the project

Are there things that can be improved, and if so, what?

there would be a tutor as helper helps the group member whom is a bit of struggleing with the project because lack of some specific ability

Lectures should be prerecorded 1 week prior not 1 year prior.

yes, i would really like the lectures to be in person (or at least have the slides), this is a very interactive unit and not having the lectures being interactive inhibits how i learn. also, i would prefer if we got feedback for the 'professionalism' mark after each sprint, perhaps from the auditor? I am finding it hard to convince my group to keep up a high standard of minutes taking, risk register etc because they thought the first sprints work was overkill, where i thought it was professional work.

SOme of the lecture content feels lacking. Almost like its targeted towards students instead of the graduate/ entry level worker the rest of this unit seems to be targeting.

lectures are a little redundant

Definetly make the marking rubrics more informative, making the criteria for getting a 10/10 - Use this sparingly. Gives me no information on how i get that

- I would like it if we could be a bit more flexible in how we document things (the timesheet in particular is difficult to work with), i would prefer to be allowed a bit more freedom to personalise the document for my team - the marking criteria for the sprints so far has been a bit vague (perhaps by design?) which my team struggled with

For peer assessment SPARK, having a mark ranging from 1-5 will not be useful from where I think, as we can see from sprint1, almost all of us get 4, no one's at 3 and just one got 5, if that's the case, at the end of this semester, all of us will have 4*3 mark, indicating that all of us contributes to the project evenly, which is not possible at all. Since UWA required us to have same mark on group project, I do not really know how to address it, and actually I do not that mind this, it does not affect me that much, but if there are someone who deserve a 0 and get 4, this 80% extra marks may matter something, maybe? may not? Just to let you know this, and I am okay with it. Thank you.

N/A

I hope to return to campus for classes, although this is still difficult to achieve. Because we study at home, the efficiency may be reduced, we may spend more time on each units. So I hope to reduce the workload a bit.

I dislike the reflection aspect for each lecture, I feel all the work each week for the project is already taking up my time and I feel overly taxed by this unit because of too many things to do at once each week, when I'd rather be focused on delivering a good project outcome. I think the professional development portfolio is asking for too much on top of already working on the project.

I think for this unit, that just the project on its own (i.e. not the lectures/ lecture reflections) is enough. Maybe the lecture content can be covered in previous units

- I understand it is a big job, but it would save a lot of time from a team's perspective if the csse website could be more structured, more succinct and if possible searchable. For example, we often have to dig through the various pages to find specific information (e.g. it could be under Project, or Project Instructions, or Sprint Description, or Sprint Marking Criteria, or clarified on help3200). - There is also uncertainty about when weekly deliverables are due/when we would get penalised (e.g. 'Friday (but certainly no later than Sunday evening)' does not indicate at what point in time we would get penalised as 'late'). - It is also unclear why we need to submit everything to cssubmit as well as MSTeams as well as email to both auditor and client. Submitting to several locations when the client and auditor only require an email seems to be unnecessary. However, I suspect this is because we do not have an understanding of who needs to see the documentation in each location, in which case we would appreciate an explanation. - Additionally, it would be appreciated if mentors and audtiors were provided with clearer instructions before the project commences, e.g. according to csse we need to have a mentor meeting on specific weeks, but it is also unclear why they need to be held on specific weeks rather than be flexible? E.g. sprint 2 retrospective is due to cssubmit at the end of sprint 2, however the auditor has a different understanding and believes it should only be done after sprint 2 is complete (which I also believe is more logical given the retrospective is a reflection on the work done in sprint 2). - Minor things but would mean we would waste less time: the individual member's booked hours template asks for a specific format, but isn't formatted correctly in the cells, which seems counterintuitive for a template. The formulas for calculating the total hours has also not worked for any of our members (Mac or Windows) so we implemented our own ways of making the sheet work. The whole team timesheet template caused a bit of confusion at the start, with the requirements suggested and the estimated hours we were supposed to put, but which we have now ammended.

Are there things that can be improved, and if so, what?

It can be a bit daunting to be thrown into a project like this when you have no industry experience. The biggest obstacle I find is not that my technical skills are lacking, but rather my general experience and knowledge of various apps, API, libraries is. I don’t want to reinvent the wheel, but all computer science assessments prior to this unit have made us do that until now, so I don’t really know how to approach this project when all other compsci projects I’ve done have just been me using only an IDE and coding for hours on end. If this unit could be a little more personalised, that would improve it a lot in my opinion.

Cant think of any right now

Certain requirements of the project may require skillsets which we do not learn in our course (e.g. mobile computing, which is only taught in a Masters-level unit instead of an undergrad unit, so undergrads may struggle in this area). I feel more consideration needs to be put into education level before assigning us to a project.

Documents For the first sprint, it would be better if the style/format of the documents were given to teams. Also, it would be good if there were example documents filled with some examples from a previous project of what is to be expected (E.g. The requirement document with an one or two examples of written requirements). I understand the unit is about putting all that is learnt at university together and as such, has a large scope, however, it has to be assessed and it is difficult from reading the current criteria about how you could get a HD. Marking Guide / Criteria Further on the above point, the marking criteria is very judgmental and I believe it is harder to understand exactly what the marker is looking for. I would suggest are more quantitative marking criteria. For example, for the Professional Development Portfolio: 2-3 = Related lecture to 1 example of project and no example in future 4-5 = Related lecture content to 2 examples of the current project and 1 example in the future ... 8-9 = Related lecture content to 3 examples of the current project and 4 examples in the future 10 = Completed to level 8-9 and has shown outstanding application in industry

Simpler group timesheet Feedback on progress built in to assessment

Summary: good students are punished for doing more work than others. There is no solution to this issue without changing how groups are comprised and how projects are distributed. For starters a big issue is almost every project is make an app, and the current cs degree gives us ZERO skills for practical app building. I have found that the majority of a group will be comprised of students who only do their assigned course work and nothing more and as such are left helpless when asked to build something useful. As a result students who go out of their way to expand learning are punished for doing so by having to do a disproportionate amount of work whilst still being subjected equally to other forms of assesment. Furthermore if a group is judged on quality of the product would it not make sense for the team to do whatever is necessary to get the best product for the best mark. Yet if one student puts in over 100 hours they will be penalised for doing so so they are insecticides to lie about hours for their own mark, or not put in the work required for a good product, both of which do not reflect what a real industry situation would be given a deadline and puts a lot more stress on well performing students than others. Finally let's say a good student wants to solo the project or do it with the other good student in the group. Because of the team marking system and hours they are encouraged to help weaker students learn to do things. Which in terms of the limited hours available is not ideal as double the time will be spent teaching the weaker student how to do it, fixing the issue when they stuff up and redoing it afterwards so it actually works as opposed to just doing it once. Potential solutions: Allocate groups not with a normal distribution of marks but an even distribution, so some groups will have all good students and others all bad students. In this way all groups can partition workloads more fairly while still retaining the differences in individual skills that makes allocation worthwhile. Furthermore change what projects are allocated to what class of group, given that there is a vetting process for each project it can be assigned a difficulty and then only a specific set of projects will be offered to different levels of groups for example a 'hard project' is given to a good group (with good results leading to good reputation of the unit) and more simple projects to worse groups such that they are not absolutely incapable of producing a product. I believe implementing the above will result in not only better project outcomes but better student satisfaction as everyone will be in a group of their calibre whist still having individual strengths to leverage. High performing students wont have the burden of carrying a team and low performing students will be given adequate time and space to learn. I doubt this idea will ever be considered because something along the lines of 'doesnt represent a real life team scenario' to which I disagree because in a technical team at a high performing organisation all members of a team will have appropriate competency and ability to contribute (unlike the current teams where some members are frankly useless). If you wish to discuss this with me personally I would be happy to. Simply make a help forum post referencing the very long response and I will email you.

Assigning members are not suitable. My project is about mobile app. But no one in the group has experience in mobile application.

More clear guidelines for each deliverable eg the reflections and clearer instructions for the auditor and mentor

Are there things that can be improved, and if so, what?

I am enjoying the project and the unit thus far. 1. The fact that none of us in the project have done something like this, and the fact it is randomly assigned, makes me feel that this entire unit is "the blind leading the blind". Though we have mentors, in 1-hour meetings only 3 times in the entire semester, it is unreasonable to ask this individual how to develop web applications in a team efficiently and of high quality. I believe having at least one individual in each team who had done a large project would be incredible as we have a guide. 2. The expectations of the unit are spread across the unit web page far and wide. I have read the webpage multiple times and feel that there are still things I might miss. For example, The wording of sprint 1 was in that of a computer science major. Things like 'acceptance tests' and 'scope of work' are used to describe technical documents, and there are links to technical documents as examples eg 'Requirements Document' and a 'test manual'. Our client told us that upon marking sprint 1, he could not understand parts of it because it was too technical. It is only mentioned on the 'roles' page that the client is marking parts of these deliverables, not in the sprint marking criteria or description page. There is mention of our client viewing the deliverables on these pages but the auditor also sees them and isn't marking the deliverables. I would have liked to have that made more clear so that my group and I could change our language so that he could understand it, and thus mark it with a better understanding. Simply repeating information like this on relevent pages I believe would solve this issue entirely.

I think communication on what is expected in terms of deliverables, providing more case based examples like from previous years to get a better idea especially with sprint 1. Same with conducting meetings/feedback directly from auditors would be helpful

a way to ensure that groups that have very different project requirements don't get a lower mark due to a harder project

It would have been nice to nominate 1 or 2 people we would like to have in our group rather than it being completely random.

1: DO NOT MAKE >90% OF THE PROJECTS WEB-DEV PLEASE I AM BEGGING YOU (though it's too late to save me), web-dev frankly sucks and many people like myself would much much rather a more technical project that actually requires thinking and problem solving (doesn't help though when you get stuck with a team who only wants to do web-dev see point 2) 2: Let us pick our project preferences, THEN assign us to teams so that those of us who want to do technical projects can have a team who want to do a technical projects and those who just want to do web-dev can do web-dev. (You can still trying to work in some equal dist of skill but foremost should be the interest of team members aligning) 3: Find some way to prevent the group from throwing the least qualified person under the bus to be the first PM, I knew I would be bad at it but the group all "dibs not"ed the role so I was basically forced to do it first and I am too agreeable to say no. As a result the team did worse in the first sprint and I got a bunch of unneeded stress as I failed to be good at what I needed to do.

Yes get rid of booked hour sheet and hours estimation

Clarity in the assessment outline; emailing due dates, reminders etc. Also clarity in who is marking what.

I think the 20% personal reflections is worth too much of the grade. A lot of the unit is in writing reports And deliverables and not the final product. This I take some issue with. I believe final product functionality should have a larger weighting. Having a 20% essay writing component on top of this is unreasonable for a computer science unit I believe. Additionally there are issues as always with different members working different amounts. If it comes to learning new languages/ frameworks, if a team member falls too far behind they will never be ahead and able to contribute. This is not something I take issue with for the unit. It’s clearly reflective of real work. I’m just not sure how to Best deal with it.

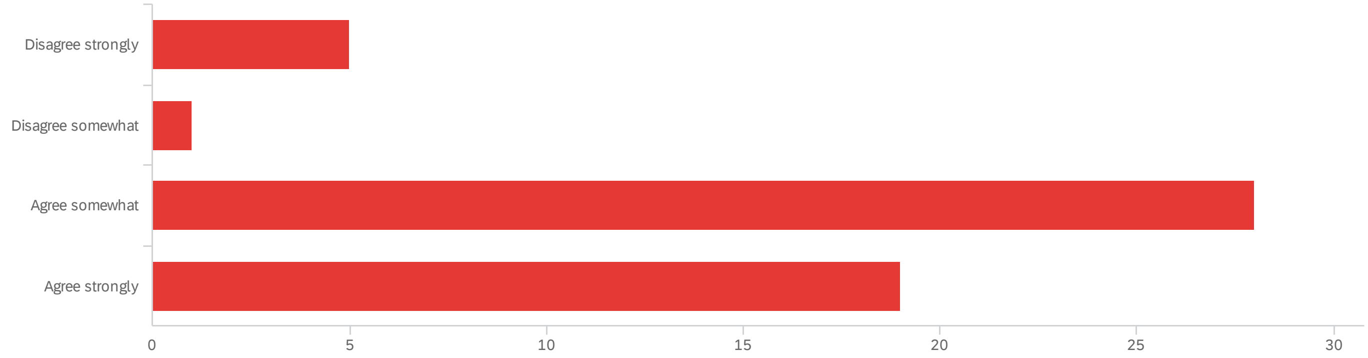
Very difficult for some students due to lack of knowledge on required topics but then very easy for other students due to previous experiences. Makes it very difficult for some students to contribute in a meaningful way

Q9 - Which resources - books, web sites, apps - that you have found useful in doing your part of the project. While some resources will be specific to your project, others may be generally useful. I'm thinking of collecting them and adding them to the Resources page

Which resources - books, web sites, apps - that you have found useful in d...
I mainly used online tutorials for JavaScript and React. https://docs.microsoft.com/en-us/visualstudio/javascript/tutorial-nodejs-with-react-and-jsx?view=vs-2019
Learning Agile: Understanding Scrum, XP, Lean, and Kanban (9781449331924) Though I have found this book rather overkill, it is a pretty good general overview anyways. https://github.com/features/project-management/ https://medium.com/coletiv-stories/github-boards-for-project-planning-2a054d5b3eed Azure & Google cloud docs for cloud hosting and computation.
Google Cloud Platform API Various Open Source Python Libraries
Outside of first page googling of queries none come to mind
I've mostly just used stackoverflow
w3schools.com is amazing for code.
Most of the resources I used until now have been a part of the project I am working on. The key takeaway point on that is that I am using a lot of documentation from previous work already done along with extensive documentation (which is too much) of Unity 3D.
any information about Azure and hosting online resources that UWA offers!! also because a lot of the projects seemed to interact with the existing UWA system, how much access we have to UWA staff resources or student resources for coding, e.g. a license to use some project management tool or some IDE or some access to see more of the UWA network
w3 schools and stack overflow for code
the best ever website for web development: https://www.w3schools.com/
for our 3-d DIC: https://www.gom.com/metrology-systems/aramis.html
https://trello.com
web sites is really useful, when I have some questions while doing project. I can get lots of resources from websites by search the key words. This is more efficient than reading.
Notion FreeCodeCamp
Specific to project: - For Web Application development using Flask and Python https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world (Corresponding GitHub https://github.com/miguelgrinberg/microblog/tree/v0.1) - For getting started with Virtual Environments with Python https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/
I think this is a really good idea. In my project we are using: Docker Spring PostgreSQL Webflow Balsamiq Might use powerautomate

Which resources - books, web sites, apps - that you have found useful in d...
Youtube, google, github, Azure, Stackoverflow.
Tutorial on Web Application Development with Flask: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world
https://firebase.google.com/docs/firestore https://reactjs.org/tutorial/tutorial.html https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
Resources are very very project dependant. Googling the relevant technology is about as good as it gets.
We have been using Django to build our web application, thus the documentation is helpful, however, I think this resource is too specific.
Mosh hamedani
trello is a nice app for managing projects
Google Documentation for Firebase, App Engine and their various API's is comprehensive and complete. I have found this to be my main resource so far. For web development I have also made use of W3schools.
Google? What do you think I'm buying books or something for this?
Online tutorials on youtube.
Udemy for learning node
I have used youtube, W3Schools, Khan Academy and many other resources
W

Q10 - Overall, while there is some way to go, CITS3200 has been a worthwhile experience



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Overall, while there is some way to go, CITS3200 has been a worthwhile experience	1.00	4.00	3.15	0.86	0.73	53

#	Field	Choice Count	
1	Disagree strongly	9.43%	5
2	Disagree somewhat	1.89%	1
3	Agree somewhat	52.83%	28
4	Agree strongly	35.85%	19
			53

Showing rows 1 - 5 of 5

End of Report