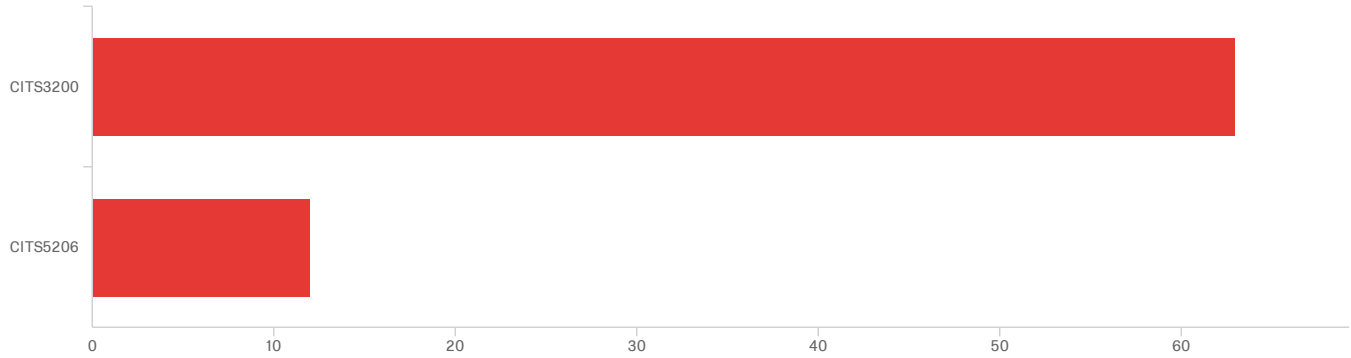


Default Report

CITS3200/CITS5206 2019 Survey - take 2

October 14, 2019 12:43 AM MDT

Q1 - Which class are you in:



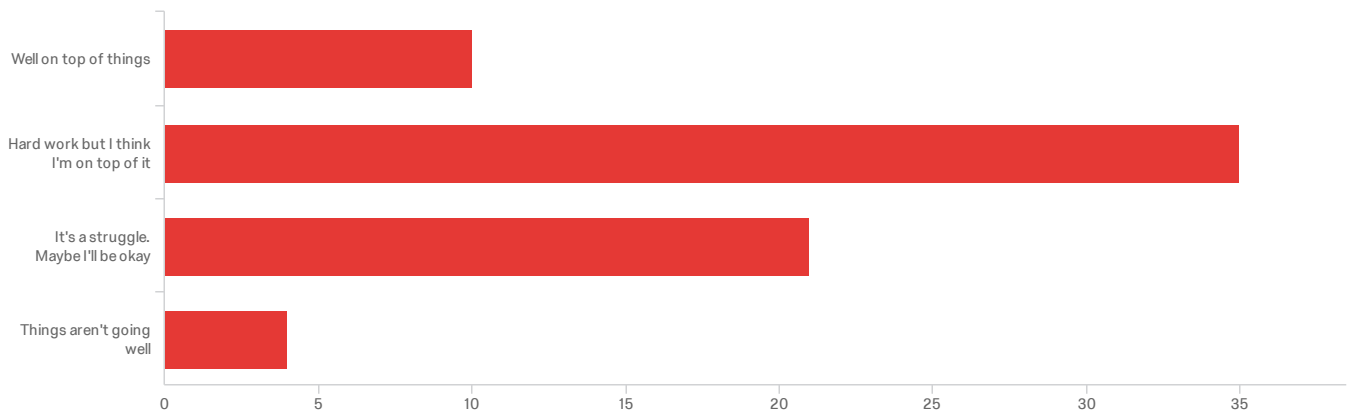
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Which class are you in:	1.00	2.00	1.16	0.37	0.13	75

#	Field	Choice Count
1	CITS3200	84.00% 63
2	CITS5206	16.00% 12

75

Showing rows 1 - 3 of 3

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

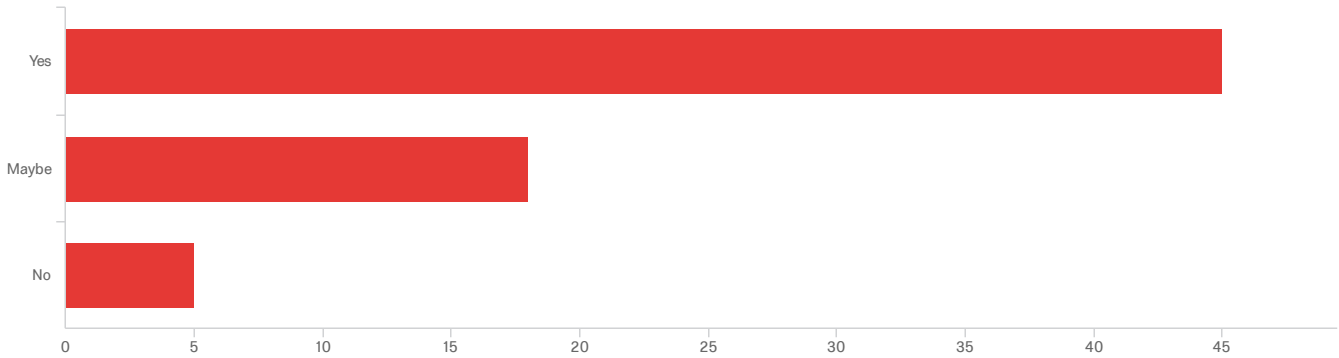


#	Field	Choice Count
1	Well on top of things	14.29% 10
2	Hard work but I think I'm on top of it	50.00% 35
3	It's a struggle. Maybe I'll be okay	30.00% 21
4	Things aren't going well	5.71% 4

70

Showing rows 1 - 5 of 5

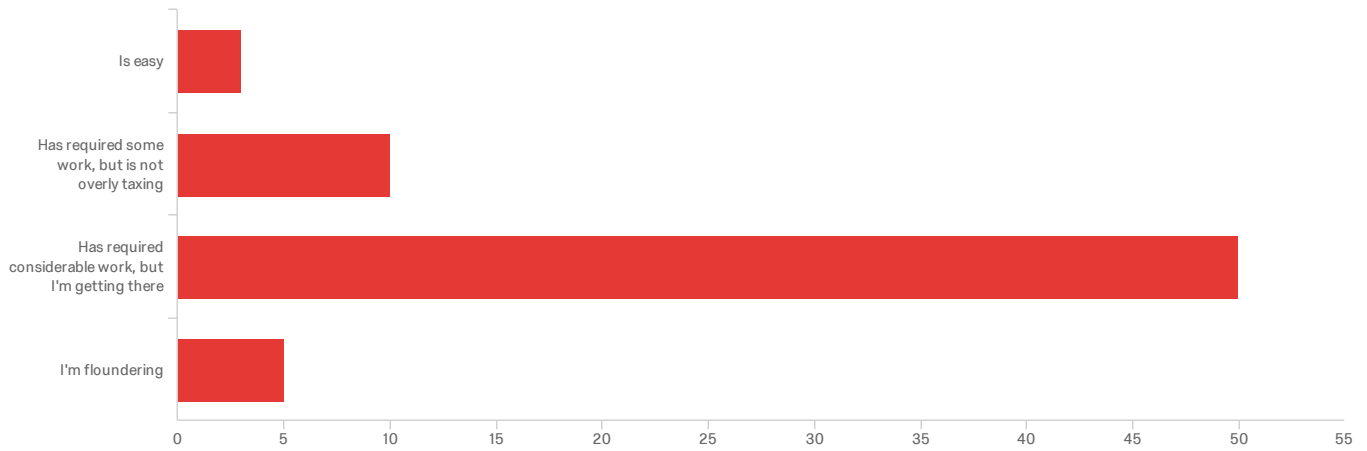
Q3 - Thinking about the Project. The idea behind the Project - unlike the lab exercises and assignment you've done in other units - is that this is where you put together what you have learnt and get the experience of different roles in a project team. Are those design goals reasonable?



#	Field	Choice Count
1	Yes	66.18% 45
2	Maybe	26.47% 18
3	No	7.35% 5
		68

Showing rows 1 - 4 of 4

Q4 - So far, I'm finding the Project

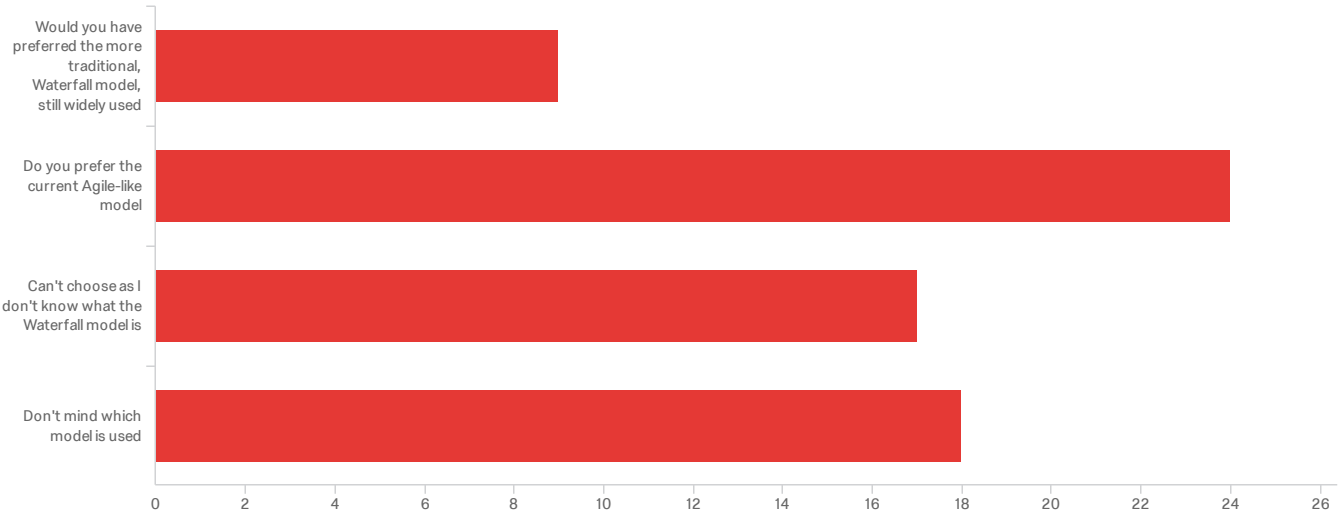


#	Field	Choice Count
1	Is easy	4.41% 3
2	Has required some work, but is not overly taxing	14.71% 10
3	Has required considerable work, but I'm getting there	73.53% 50
4	I'm floundering	7.35% 5

68

Showing rows 1 - 5 of 5

Q5 - When CITS3200 was redesigned last year, it went from a a Waterfall model of IT project management to a more Agile-like model



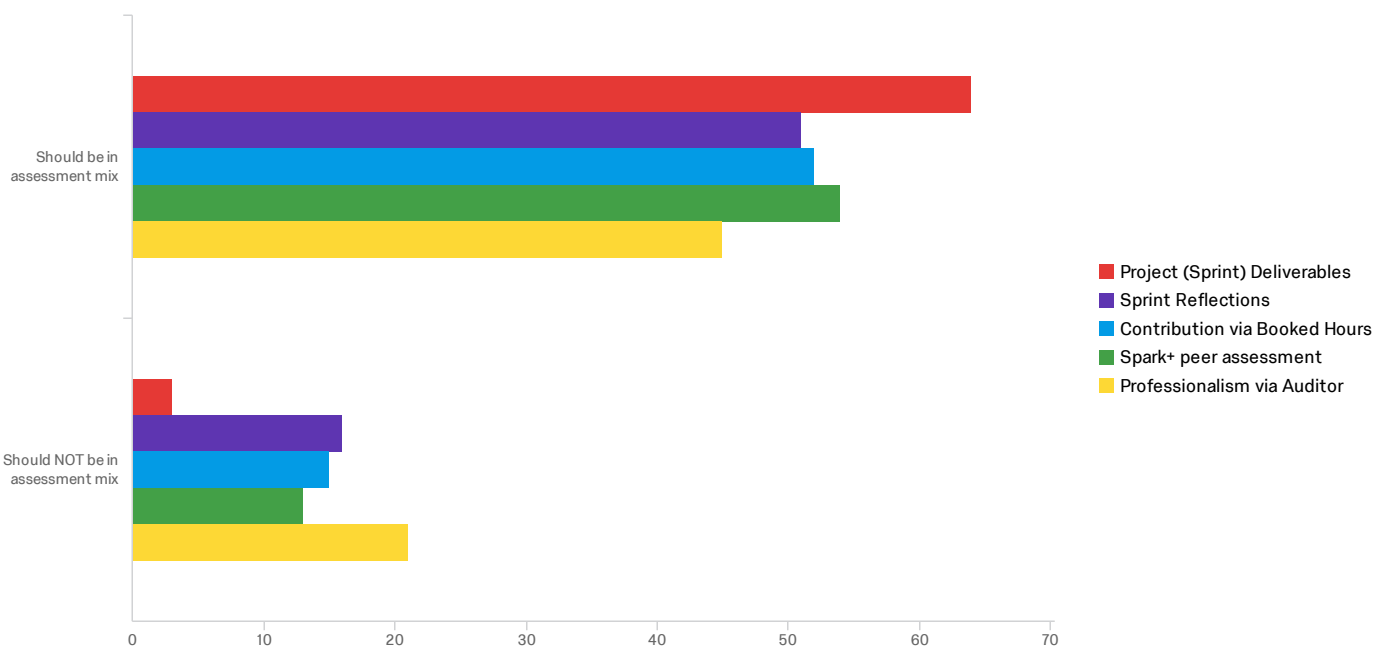
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	When CITS3200 was redesigned last year, it went from a a Waterfall model of IT project management to a more Agile-like model	1.00	4.00	2.65	1.01	1.02	68

#	Field	Choice Count
1	Would you have preferred the more traditional, Waterfall model, still widely used	13.24% 9
2	Do you prefer the current Agile-like model	35.29% 24
3	Can't choose as I don't know what the Waterfall model is	25.00% 17
4	Don't mind which model is used	26.47% 18

68

Showing rows 1 - 5 of 5

Q6 - No assessment metric is ever sufficient 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	1.04	0.21	0.04	67
2	Sprint Reflections	1.00	2.00	1.24	0.43	0.18	67
3	Contribution via Booked Hours	1.00	2.00	1.22	0.42	0.17	67
4	Spark+ peer assessment	1.00	2.00	1.19	0.40	0.16	67
5	Professionalism via Auditor	1.00	2.00	1.32	0.47	0.22	66

#	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	95.52%	64	4.48%	3	67
2	Sprint Reflections	76.12%	51	23.88%	16	67
3	Contribution via Booked Hours	77.61%	52	22.39%	15	67
4	Spark+ peer assessment	80.60%	54	19.40%	13	67
5	Professionalism via Auditor	68.18%	45	31.82%	21	66

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 unit places appropriate emphasis on teamwork and professional development. I've learnt a lot about how I can be a better team player and all the requirements involved in creating a product/service for an actual client.

Team work has been fun and the roles are interesting and good to learn about

Sprint reflections help me derive the whole part of the program and find how I can be better in the next team.

The reflections have been helpful, the fact that we have to change roles during the project

You get to choose an interesting project to work on from beginning to end.

Close to the real life project process.

Getting to experience real work environment where there are no labs and few lectures.

Valuable insight working in an organic team. It's really good setting our own direction and enjoyable working with partners.

It has been a new experience for me in a project oriented work environment, one that I feel will be useful in the future.

Working as a team. More freedom in how I work

Working in a real project environment and dealing with problems that are very typical of a professional computer scientist.

lets me learn things that is actually required by real clients

I like that the maximum team size is greater than two. Other units should learn from CITS3200 because a team size of six reflects the real world much better and allows us to take on different roles.

I have found the unit useful for learning how to quickly go from strangers to team mates with people on my team within a matter of days. Also tying together all the different technologies into a single project has been useful for my understanding of programming.

Real clients and projects

The lecture content was good.

The auditor has been really helpful. Our client tried to change the licensing agreement and the auditor provided us with good advice (mostly by reassuring our ideas).

It has been a good experience meeting and working with new people. It's also been nice to have the opportunity to talk to more experienced members of the computer science community.

Engage with real project from industry is one good way to introduce student to what exactly work is.

All the experiences within the groups are very valuable and good for resume and future career opportunities. Pretty much it's a fine and fun unit.

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

I liked the way the lectures were industry oriented, they gave a lot of insight that other CITS units barely touched on. The spacing between different events and deliverables on the timetable is reasonable (e.g. having all of week 1 to get settled into teams and pick a project) and didn't feel too rushed. I liked how groups were able to choose their own projects.

.Learn about Professional behaviour . How to work with team .Time management .Gain many knowledge about different types of softwares

Team Work Professionalism Learning New technologies/Languages Management skills

Project management skills and working in a team environment

learning new stuff about working as a group and see the challenges

The unit provides a much more realistic software developer experience than other uni units, which is excellent

I think the lectures that related directly to software project management was good and writing reflections about it helps everyone have a better understanding of completing projects. My industry mentor is also really helpful in relating this project to real-world projects.

Agile methodology, client interactions and source control.

It's a good test of how hard you can carry someone, but if I'm honest, the unit is more frustrating than useful.

Only mentor meetings and accreditation

Good to be working on an actual project.

Its good experience for me , I have been able to interact with client. This has given me confidence to move forward. I thinks its good.

Industry experience

The experience of working for a client has been a good experience.

Nice to work in a large team for a real project that may be used by our client

the intention

So far the change to the unit has been good, I have done the CITS3200 in my undergraduate so I am familiar with the older scheme. I do prefer the current one, the agile methods certainly feel more modern and inline with the current industry trends as I think this unit should follow. Not having to provide a full design and plan of the project this time was good, it follows the agile philosophy and indeed the philosophy of software process that the specifications, plans and requirements will change. The organization of groups and projects was just as successful as last time quite a feat given the 150+ students you likely have compared to my last experience with CITS3200.

Great to experience industry-like work experience & practice professional interaction

Dealing with client, mentor, auditor. Teamwork skills

Mentor meetings are very useful. It's also been a good 'realistic' perspective of teamwork.

It has really helped me put together all of my required knowledge and put me in a situation that I have never been in with group projects that require to work with a client.

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

Nothing.

The lectures on ethics, copyright, planning. Opportunity to work in a team environment.

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

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

Lecture slides for all lectures would be helpful for writing the PDP. More information about what to expect when working on the project would be nice to have (I'm aware it's part of the learning process but a heads up about whether we're expected to deploy and other related issues would be nice ahead of time.)

The first few questions in this survey were confusing and worded weird

Auditors could participate more in these auditor meetings and give some professional advice about the performance of the whole team and individual.

Have found that having 'real life' clients makes it quite unfair, as every team gets a different marker and some may be more lenient than others. Would prefer if there were only a handful of projects to pick from that the unit coordinator made, so that the project could be more structured, and the assessment more fair.

Personal marks (even if its a group project, there's no reason if you've proved you've tried all methods to make it work) Plenty of checks for people trying to game the system, none for simple incompetency eg. Someone in the group might take 45 hours for a 3 hour task, I don't doubt they put in the time, but they just can't do it or to be of quality, even with hand holding. Some rubrics are vague at best Lacking of any feedback throughout the semester

You shouldn't have to max out your contribution hours to get full marks for that. If the group is performing well and efficiently, we should get rewarded with more marks for efficiency and effectiveness as the group complete the project before the deadline (which is what you want). This may not be always the case but there shouldn't be some kind of one-formula-fits-all to calculate the scores. Sprint reflection and professional development portfolio shouldn't worth more than 30% of the unit as students will lose a lot of marks because they have no idea what is expected or English isn't their main language. Be more specific of what is expected in the reflection at the very least because the description is too generic, consider a marking rubric or list out some topics to be discussed. Spark+ should have a more discrete scoring system. I dislike the idea of dragging the scores along the bar. It should be 5 discrete options for students to choose, just like in surveys. The reason being student can click anywhere inside a box (without realising you're supposed to drag the scores) and the system will take it literally. Misclick may happen and it will messed up the responses.

It would be really good to have formal lab times where the group has to work together. Assigned maybe twice a week on CAS, where the group comes together and maybe there's a lab instructor. The project needs to be worth more marks. It's ridiculous the amount of works sprint 1 and 2 are compared to the the reflection. Figure out a way to make the work worth it.

There are a considerable amount of reflections to do, and the project portfolio especially can be quite long winded. There are some lectures which could be cut in the future, as a few repeat similar principles and ideas (such as the 2 ethical issues lectures). This makes it harder to write reflections with actual and original substance not already mentioned in another reflection. These reflections require a considerable amount of effort to complete and often feel more like a chore than a learning experience. Writing redundantly also cuts into time that could be spent improving the project or completing other work. As such, this is likely my biggest gripe with the unit.

It would have been good to have lectures on: how to properly include people in group discussions, how to delegate and discuss tasks without being condescending. Lectures on diversity/inclusivity about addressing subconscious biases I think would also be useful. I think SparkPLUS should score people based on the expected level of work done, rather than the average level of work done. For example, if the entire team is disorganised, they will all have an average level of organisation, but that average is low. I don't like that part of the professionalism mark comes from the auditor meetings. I am a quiet person and having an extra person listening in tends to make me more quiet and the group does not accommodate for this by giving me extra chances to talk.

The assessment requirements could be displayed more prominently, better formatted and with a little more detail

I think the information and layout of the unit webpage is very confusing. Would be easier to understand what is required if this was more simply layed out.

No offense, but maybe the unit web page. It was a little confusing understanding everything about the unit and so a quick summary would be handy.

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

This may seem bias, but I feel the people doing the Data Science stream come into the project much less prepared than the Computer Science stream. I have had many team members not know how to use basic technologies or have any understanding or programming data structures or algorithms. This is frustrating because I feel I have spent most of the project explaining to them trivial details, or I end up just completing the work for them (they end up booking 8 hours for a task that realistically can and should be done in no more than an hour). I have no idea how this could be improved.

Group allocation Random group allocation isn't all too fair Should be GPA/WAM based similar to engineering or self elected

Relating work done to booked hours seems to penalise efficiency and is busy-work. Goals-achieved would be a more useful metric for marking purposes.

More guidance was needed in the first sprint. My team jumped straight into full development of our project and it has been a very stressful experience. The client should have also been made aware that we should not be expected to deliver a working product by the end of sprint 1!

The unit would be better if it takes a proper agile approach. The 'wagile' approach we are currently using is confusing. Sprints should be 2 or 3 weeks and not assessed individually. The auditor and client could mark these as a progression of work rather than 'sprint deliverables'. My team feels under pressure to complete everything said at the start of the sprint even though now we know it is probably not the best route to take. Also since it is a deliverable, there is a scramble to complete tasks but possibly in a 'hacky' way that will need to be fixed next sprint. This is not in the spirit of agile but is a result of needing to have a set deliverable 'due date'. (We are not really making very hacky code but you probably see the issue that could arise)

Better coordination between all stakeholders. Very messy unit, that has a lapse in communication and rather amusing assessment structure. Still not sure what the unit is trying to teach....

A criterion-based marking scheme should be provided for future assessments as the current assessment rubrics violate the criterion-based marking system adopted by UWA. In the marking policy: "11.1 Marking of all assessment items/tasks must occur on the basis of pre-specified criteria consistent with a criterion-referenced framework, and not determined in relation to the performance of other students, nor in relation to predetermined distribution of grades/marks". Taking the "Professional Development Portfolio" marking scheme as an example; "10 Stunning work (use this mark sparingly)" gives no distinction between a mark of 8 and 10 and invites a comparison between your work and that of other students. With regards to auditor and client marks, there is no mention of moderation in the unit outlines. It is a requirement that there is moderation between assessors (see section 11.4 of the assessment policy) and a description of the moderation methods that you intend to use would be appreciated. As stated in section 15.1, "Students must be provided with feedback on all assessed work". We should not have to request feedback from our client and auditor as it should be provided when we receive our marks (our auditor seemed quite surprised when we asked for feedback about our results).

The Timetable. Everyone in our team is really under pressure all the time to get everything done. Besides the Portfolio which is due week 9, it has to be moved to week 10 (after the BREAK). Also, I don't see any value in doing the Portfolio. But if it's a requirement that must be in the unit, I think a sooner due date is more appropriate. For example, it should be due a week after the lectures are done.

No actual structure of the unit. A lot is to be assumed as prior knowledge, when its not. Rely on Internet tutorials that aren't even right most of the time. Wouldve preferred easier projects so that we can focus more on the groupwork and less on struggling to finish.

i think we are doing good so far

A lot of the project descriptions were quite vague, so it was hard to judge if they'd be something I'd be able to tackle, even as part of a group. I understand that the purpose of client meetings is to then further explain and clarify the project, but you don't get to that stage if you don't pick their project. Tell the clients that when they write their initial project descriptions, to be as clear as possible. This is because theirs will be one of many and students will judge them like "judging a book by its cover". If I don't understand what they asking for, I will lose interest and move onto another.

I think, everything is fine and no need to improve anything.

Everything is going good so far

Auditors time sheet related stuff More clear directions for the sprints and what we have to submit to the uni

lectures were not necessary for the master students and also have 20 per cent for portfolio of lectures for master students is not a good idea! it needs to be a difference between undergraduate students and postgraduate students

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

Some of the assessment is a little difficult to keep on top of, not that its overly difficult just that there is so many different pieces of assessment it is challenging to keep on top of them all

Having mentor, auditor meetings not in the same week as sometimes we have 4 meetings in one week, (mentor, client, auditor, team) then one in another week.

The lectures and lecture reflections feel either irrelevant or like a completely different unit.

This unit survey; survey should ask for confirmation before submission in case someone wished to go back to spend more time on feedback after completing all the multiple choice sections. The webpage, similar information should all be found in roughly the same locality, eg When looking for information about weekly submissions, A student shouldn't have to find relevant information scattered between the unit homepage, the project page, and the time table, the information should all preferably be found in a single page. Scattering the information increases the likelihood of missing information and requires a lot of time searching for information during the early weeks to ensure it has not been missed. The selection of projects and composition of groups: typically involves a mix of computer science and data science students. The skillsets of computer science and data science students differ significantly on the languages and technologies known. This mix of skills forces a compromise when choosing the project, which results in a project both types of students are partially weak on. Part of the problem of this compromise is that most available projects are web or database development projects and only a few are backend development projects eg Making an agent platform using sockets. Computer Science students will struggle on front end projects, while Data Science students will struggle on back end systems level projects. Preferably there would be a proportional number of front end projects to data science students and a proportional number of back end projects to computer science students and a team formation of data science students together to do front end projects and a team formation of computer science students together to do back end projects so that both groups are working towards their skillsets. This should only be done if there are enough back end projects, as with a high proportion of front end projects, computer science students alone will struggle learning the front end skills required without the guidance of data science students. Michael to discover that the stable marriage problem is not NP, the stable marriage problem is known to have a polynomial $O(N^2)$ solution known as the Deferred-Acceptance Algorithm. Please stop telling students that the stable marriage problem is NP. A modification on the algorithm may also be used to save you time allocating teams with a final manual check through to check for fairness, there are pros and cons, using the algorithm seems to be an executive decision. Please stop telling students that the stable marriage problem is NP.

Firstly, randomising teams does not give an accurate representation of how you work in the industry. I understand that teams might not be 'balanced' if picking teams is allowed, but for the people that actually put in effort to find teams/already have teams, their experience would much more mimic what would happen in industry, as it's not just a 2 good people, 2 average and 2 that do nothing. The templates are not clear and outdated. As an example for sprint 1, the testing template was something from 1999, and it wasn't known if we were to write every test or to just briefly explain tests for the finished system. Peer assessment is not a good indicator either, as some teams literally just give each other 5/5. The auditors should actually go into the projects and judge how much work is actually done, or in the future, have it mandatory for the auditor to be in discussion chats. Even if someone contributes nothing and the rest of the team get the project to 100%, the person who contributed nothing still gets a decent mark. This isn't fair. Furthermore, allowing second year students to do this subject is just completely unbalanced on the teams that get them. Even if they're good students, not having the knowledge from 3rd year subjects just makes them more of an anchor than a helpful force. The whole hours based contribution is horrible as well. 1 hour of my time compared to somebody with 0 experience in some technology that we are using is extremely different. He may have done 5 hours, but I can complete the task in 1, but it looks like I've contributed to the overall project less? This would again be fixed by the auditor looking over things personally. Another frustration with peer assessment, if I have a problem with somebody, why would I tell them? They will just mark me down in peer assessment intentionally. The system is inherently flawed.

The unit webpage. SparkPlus assessment, should be not be relative to other team members (zero sum) where allocating a mark to a member will effectively detract from a mark of another member; An effective well working Tiger Team of Nobel Medalists should get 5/5 on their SparkPlus assessment, rather than 4/5 because they are all working relatively equally well. Non-relative still allows for penalty if a team member is disruptive, but also allows a well functioning team to get full marks.

Different clients have different marking scales. Especially clients from outside of uni are tougher than the uni ones. Our group has been on top of everything and Auditor has been really please with our work compares with other groups. But clients wasn't really giving us a relatively good mark (under 70%). We are trying to communicate with the clients to see where are the parts that they are mostly satisfied about.

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

Things that could be improved actually don't follow the agile methodology, having some documentation focusing on development, reviews and retrospectives is good. However, I think certain things should be added, such as an outline submitted by end of each sprint to show plans for the next sprint. A project backlog being developed and submitted in sprint 2 and 3 would be good too. This will make it more in line with the Scrum process which seems to have somewhat inspired the unit's structure. Also, while I'm familiar with GitHub, some of my team members despite the good resources you have provided do not seem to know how to use it. A suggestion is to incorporate a lecture on Git and some of its basics, committing, pushing, branching, merging etc, to ensure that people understand the importance of learning how to use it. Also, this one is particularly for CITS5206 students, these lectures are very similar or almost the same as the ones from my previous CITS3200 experience. So, I as a Master's student do not get a lot out of them due to having completed GENG5505 Project Management, CITS5502 Software Processes and CITS5501 Software Testing and Quality Assurance which cover most of these topics but in greater detail. I'm not sure whether every CITS5206 student has completed these units however since they're all core units, its likely that most CITS5206 students will find the lectures more like revision and reminders of the fundamentals rather than learning tools (which admittedly during CITS3200 I found them enlightening). Whether we should have different submissions or assignments instead of the professional portfolio compared to CITS3200 I'll leave up to you. In listening to Terry I almost knew what he was going to say before he said it.

Website navigation is confusing and inconsistent. Booked hours does not always reflect contribution (if a member is more productive)

The goal of the project is to emulate a real-world project. However, there are a few things that everyone needs to learn/be aware of in order to even work on the project (Git, GitHub, Agile methodology etc). Whilst I understand that these are skills that are used in the industry and should be taught, it does present a steep learning curve for a lot of students. As a result of this, the unit is also assessing students' understanding of these tools - which is not a goal of the unit.

Spark+ and even GitHub contributions are poor measures of team participation, and alternatives should be pursued. Also, every unit assumes that some other unit teaches git. This would really be a good unit to officially teach git, because I'm surprised to see that not many students are comfortable with it.

The marking guides need to be definitely be updated. A lot of things are not worded clearly and the site definitely needs an update to have a better view of the whole unit. A lot of things are easy to miss and it is hard to get around without looking for it for ages.

The unit should be crafted with the help of current industry professionals. Nothing in this unit feels like it is applicable to professional software development.

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...

Trello for taskboards and lists is pretty handy.

It depends because each project is different for example im using python, so I've been looking back at python lecture notes and books. But my friends are doing back end and front end web development so they need different resources

Web site. In this unit, I have to learn much new knowledge about doing a real product and web site is a good choice for me.

The only resources I have found helpful were the ones about PHP as we had to learn it for our project

Trello and Slack have been helpful in organising group work and increasing communication and accountability.

Slack. Good communication tool.

W3Schools, GeeksforGeeks, Tutorials Point and Django website for their tutorial.

I found the website Scrimba useful in self-learning new languages that may be required for a project. It offers free courses with hours of video and an in-video live demonstration of coding and the code itself. It also lets the viewer pause the video and use the editor in the website directly to test things as they go.

Honestly, I just Google things as I go. That said I always recommend extra credits "fail faster" video

stackoverflow

heroku

W3Schools has been a big help throughout the project. Also the main Django website with the tutorials.

My group did a website, so we relied quite heavily on the Flask Mega Tutorial which was used by Tim French is Agile Web Development. That resource has been the holy grail for us. Other than that we simply googled what extensions Flask was compatible with and used the most relevant to our requirements.

freeCodeCamp on youtube for learning JavaScript, node.js, react.js, express, and many other web development info. I have had to learn all of this from zero prior knowledge.

Atlassian Agile documentation (would be better if actual agile was used) Trello (for story management) Slack (for communication) Discord (we have daily stand-ups (10mins or so, 5ish times a week) to keep everyone on track) Github is useless - I would try switching to another free platform with a better UI. Other resources are specific to our project.

websites - tutorials

Which resources - books, web sites, apps - that you have found useful in d...

I have only used the relevant language documentation.

Trello for Scrum

We're using Django.

In my part, I use html, css and javascript so the website which helped me in my part is W3school. This is fantastic website to improve the knowledge of student.

For creating Front end of project in HTML and CSS, W3 schools is best.

Too many to count

Stack overflow general documentation of frameworks etc

I've been keeping a record so here: https://chromium.googlesource.com/chromium/src/+master/docs/documentation_best_practices.md
<https://dev.azure.com> GitHub: https://www.youtube.com/watch?v=0fKg7e37bQE&t=3s&ab_channel=LearnCode.academy
https://www.youtube.com/watch?v=oFYtZwMyAg&ab_channel=LearnCode.academy <https://www.codacy.com/blog/how-to-code-review-in-a-pull-request/> Automated testing: https://www.youtube.com/watch?v=vIBuNM6Wzic&ab_channel=MicrosoftVisualStudio <https://dev.to/ykyuen/web-ui-testing-in-nodejs--kda> ReactJS: <https://reactjs.org/tutorial/tutorial.html>

Github, Slack.

W3Schools

Google -> Stack Overflow

I have been using all lecture notes from my previous semesters. This has help me a lot, am able to relate to the topics even better.

W3 Schools Stack overflow

- Django's website on creating an app - Stack Overflow has been a lifesaver

get rid of all the useless stuff on diversity etc etc, especially the mansplaining nonsense

Its been really hard finding resources as different resources contradict each other. For example the project scope of work template. The one provided was too old. Different websites said different things so it was hard to come up with a template for the project scope of work

None specifically to this unit, I have previous textbooks to rely on especially from GENG5505 that has a fantastic text book, but it isn't free.

The most useful resource so far has been our mentor in explaining how Agile is applied in practice

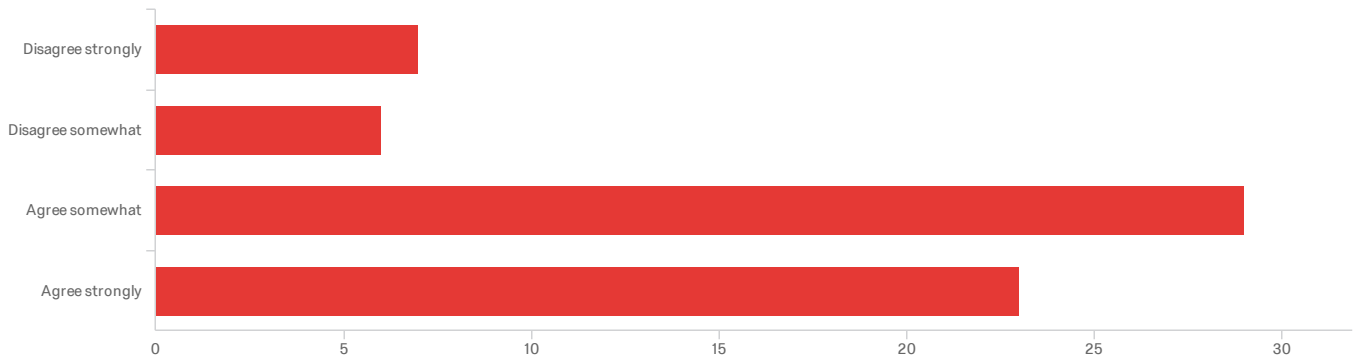
Atlassian Git Guide <https://www.atlassian.com/git/tutorials> GitHub Desktop <https://desktop.github.com/>

Trello has been pretty fantastic for managing user stories and keeping development structured and organised.

Stackoverflow has been a great source of help, but also youtube tutorials for any beginners can start there easily. Finding a few guides such as theNewBoston do great videos for beginners if someone needs to learn a new language quickly.

Resources given by mentors

Q10 - Overall, CITS3200 has been a worthwhile experience



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Overall, CITS3200 has been a worthwhile experience	1.00	4.00	3.05	0.94	0.87	65

#	Field	Choice Count
1	Disagree strongly	10.77% 7
2	Disagree somewhat	9.23% 6
3	Agree somewhat	44.62% 29
4	Agree strongly	35.38% 23

65

Showing rows 1 - 5 of 5

End of Report