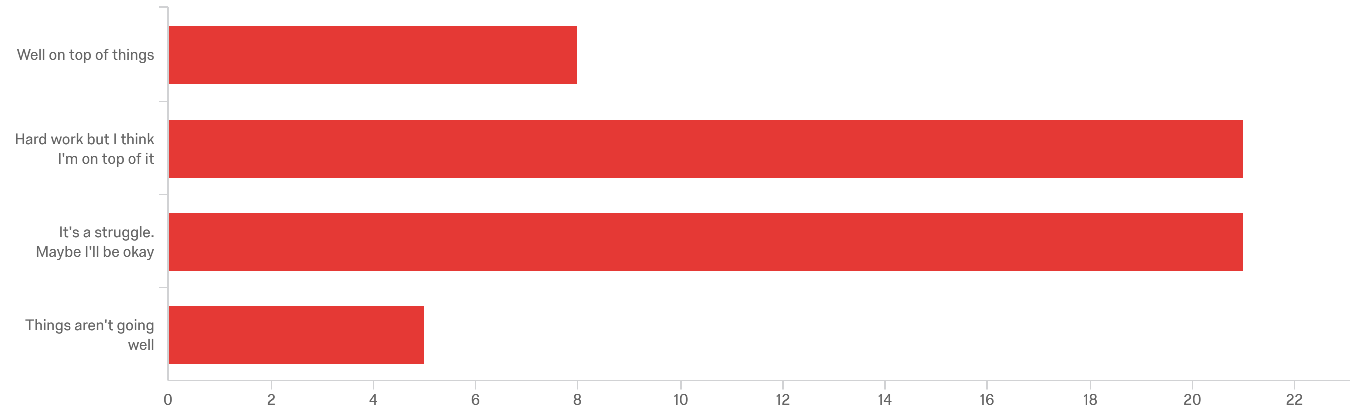
CITS3200_2018_Survey_Report

CITS3200 2018 Survey

October 12, 2018 12:36 AM MDT

Q1 - How do you feel you are tracking in CITS3200

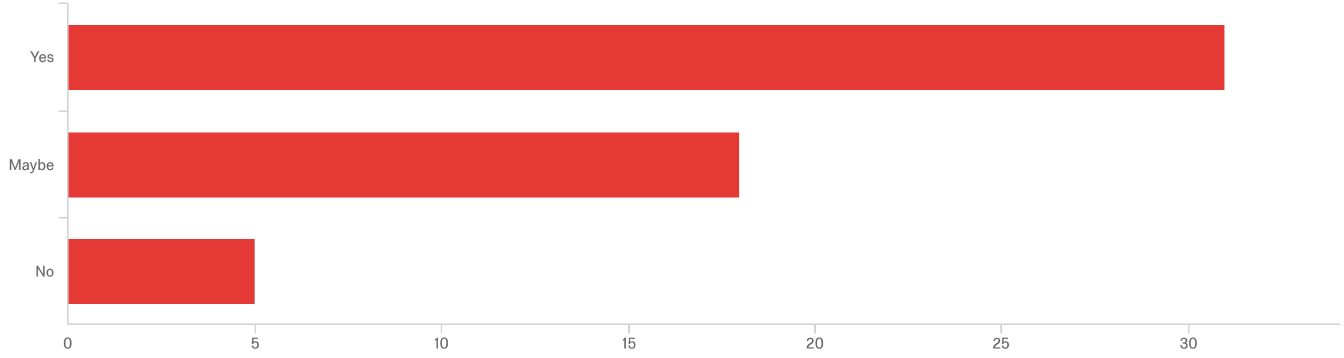


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How do you feel you are tracking in CITS3200	1.00	4.00	2.42	0.85	0.72	55

#	Field	Choice Count	
1	Well on top of things	14.55%	8
2	Hard work but I think I'm on top of it	38.18%	21
3	It's a struggle. Maybe I'll be okay	38.18%	21
4	Things aren't going well	9.09%	5
			55

Showing rows 1 - 5 of 5

Q2 - Thinking about the Project. The idea behind the Project - unlike the lab exercises and assignment you've done before - is that this is where you put together what you have learnt. Are those design goals reasonable?

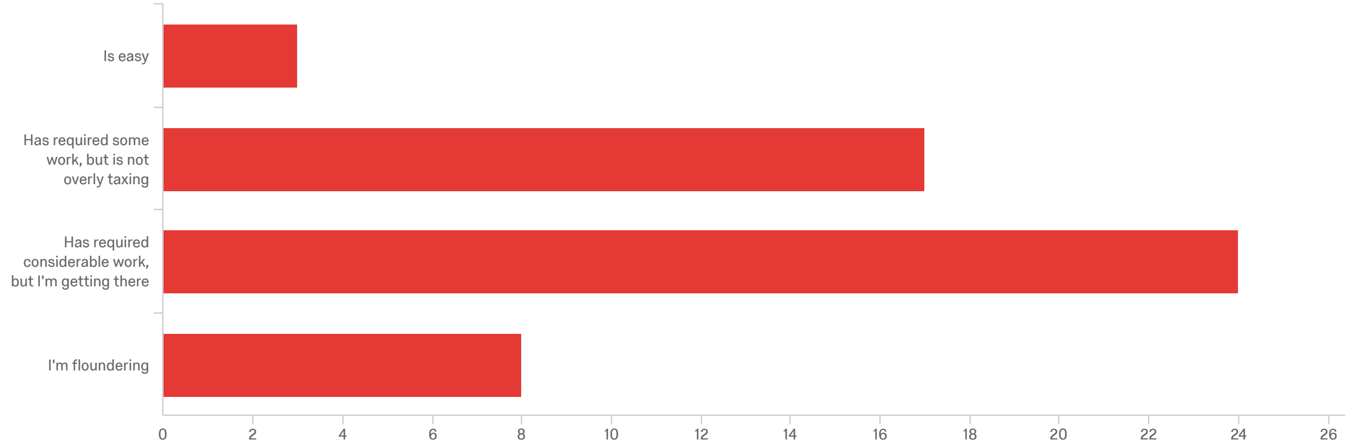


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Thinking about the Project. The idea behind the Project - unlike the lab exercises and assignment you've done before - is that this is where you put together what you have learnt. Are those design goals reasonable?	1.00	3.00	1.52	0.66	0.43	54

#	Field	Choice Count	
1	Yes	57.41%	31
2	Maybe	33.33%	18
3	No	9.26%	5
			54

Showing rows 1 - 4 of 4

Q3 - So far, I'm find the Project

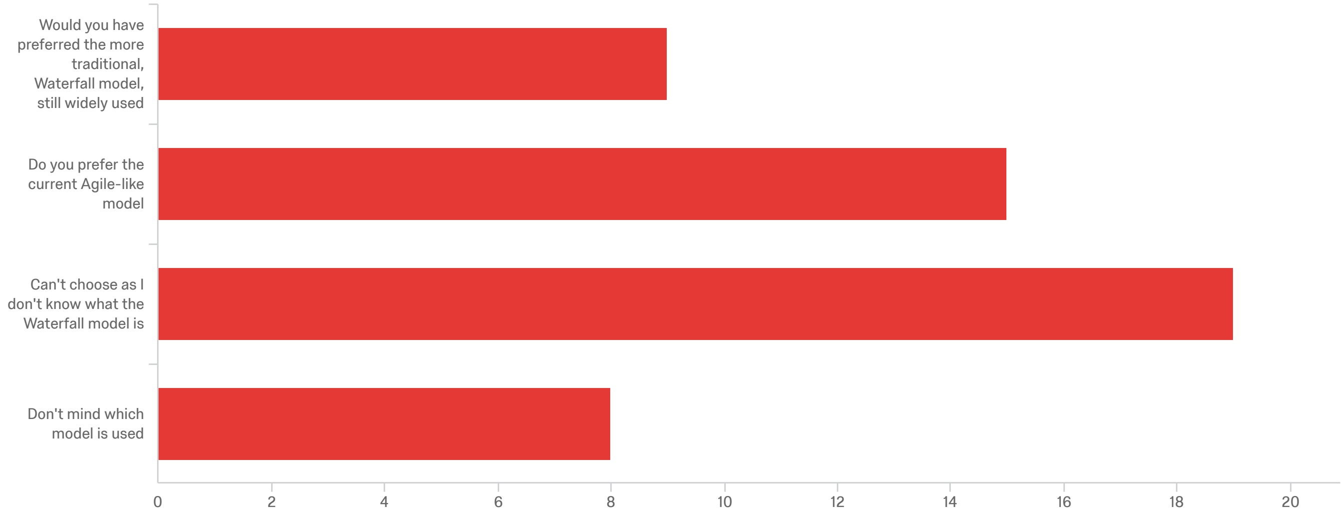


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	So far, I'm find the Project	1.00	4.00	2.71	0.79	0.63	52

#	Field	Choice Count	
1	Is easy	5.77%	3
2	Has required some work, but is not overly taxing	32.69%	17
3	Has required considerable work, but I'm getting there	46.15%	24
4	I'm floundering	15.38%	8
			52

Showing rows 1 - 5 of 5

Q12 - In the run up to Second Semester, I took the previous version of the unit which was based on a Waterfall model of IT project management, and moved it to be more Agile like.

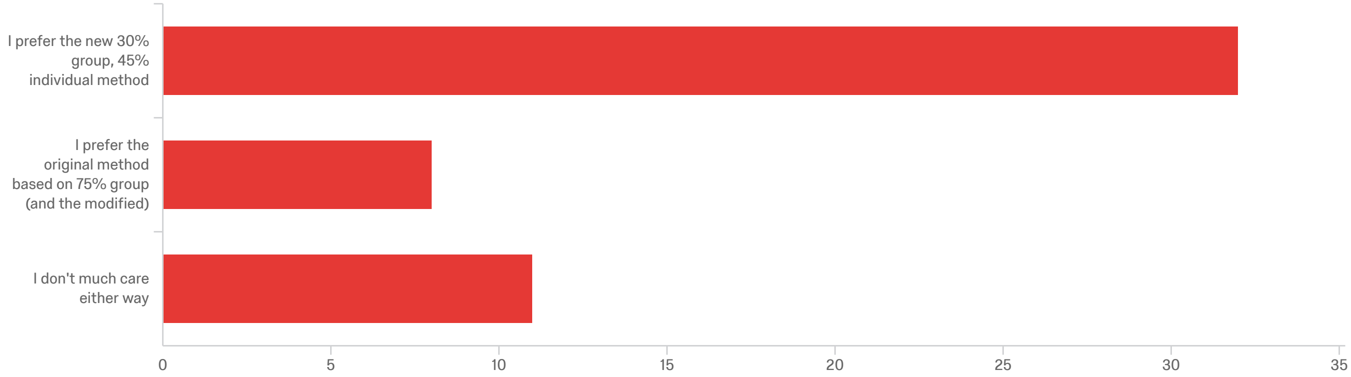


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	In the run up to Second Semester, I took the previous version of the unit which was based on a Waterfall model of IT project management, and moved it to be more Agile like.	1.00	4.00	2.51	0.96	0.92	51

#	Field	Choice Count	
1	Would you have preferred the more traditional, Waterfall model, still widely used	17.65%	9
2	Do you prefer the current Agile-like model	29.41%	15
3	Can't choose as I don't know what the Waterfall model is	37.25%	19
4	Don't mind which model is used	15.69%	8
			51

Showing rows 1 - 5 of 5

Q13 - In the run up to Second Semester, the assessment mechanism for the Project component was changed from 75% group mark over 4 team-based sets of deliverables (final personal mark modified after the fact to reflect major differences in contribution) to the new system of 30% group based and 45% based on what you do as an individual. Ignoring the fact that the University insists on a maximum of 30% group mark, what o you prefer?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	In the run up to Second Semester, the assessment mechanism for the Project component was changed from 75% group mark over 4 team-based sets of deliverables (final personal mark modified after the fact to reflect major differences in contribution) to the new system of 30% group based and 45% based on what you do as an individual. Ignoring the fact that the University insists on a maximum of 30% group mark, what o you prefer?	1.00	3.00	1.59	0.82	0.67	51

#	Field	Choice Count	
1	I prefer the new 30% group, 45% individual method	62.75%	32
2	I prefer the original method based on 75% group (and the modified)	15.69%	8
3	I don't much care either way	21.57%	11
			51

Showing rows 1 - 4 of 4

Q10 - What have you found to be good/useful about the unit?

What have you found to be good/useful about the unit?
My teamwork skills have definitely improved
Teamwork is really helpful
1- Very well organised 2- Project is interesting and project supervisor is very passionate about the project and helping us succeed (Obviously varies between groups) 3- Mentor has been incredibly helpful and supportive of us 4- Allows for the exploration of a wide range of technologies.
Mentor meetings are insightful. Personal reflections are good for expressing your true feelings about the team but should probably not be assessed.
Learning how to present project information formally, and professional practices from working with the mentors.
I feel as though I have not learnt anything. The essay was supposed to be 25%, yet we learnt nothing that related to essay writing, there is no CS unit where essay writing was involved in the past. The mode of assessment for the project is unclear. The unit feels like slave labour, we are being forced to work overtime with 0 pay and I would like to not consent to paying the unit to work for the "clients" for free while learning nothing useful. So far I have found nothing good or useful about this unit.
Learning how to tackle a programming project with a small team is something I have found to be very useful
The fact that we are working on a real project with the potential for actual use. Experience with working with and coordinating a larger team.
one of my group members has done nothing thanks
I do not believe this unit has contributed to my education in a meaningful way. University is not the place to gain skills such as these. I have found that a few months of vacation work has taught me far more about "professional computing" and this unit does not accurately reflect interaction in the workforce. While I understand this unit is required by the ACS, I believe and additional unit of theory/practical IT related content would be significantly more beneficial to my education than this unit is.
It has been great to work with talented people and learn group management skills. I like the emphasis on team member delegation and client interaction.
I love that we are exposed to real client work and mentors from industry that makes me feel like I'm learning what its like to be out there in a real working environment whilst testing the fundamentals of what I've learnt in university so far.
Team management, experience in working as a team with various set of skills, working on projects associated with real world problems.
The whole idea is excellent and so much more useful than the other CITS units (although they are also necessary). I think this concept deserves two units. Having to work with 6 people is tough, and it's unrealistic to think we can actually split the coding work evenly over each of us. But I love the dynamics of it. I have a good group though, it might not be as enjoyable for others. The fact that the lectures were pretty much optional is greatly appreciated.
The opportunity to work in a team and with a real client - it's almost a true working experience (as far as I can tell).
I enjoy how the unit has taught us to thoroughly interact with a team. In addition, I have learnt a lot about agile methodology and what real life projects are like. However, I find it unfair how the projects all differ in level of difficulty and that some groups have projects that require barely any time or effort whilst others are incredibly difficult and time consuming. In addition, some of the mentors, clients and auditors are difficult to contact and meet up with sometimes which makes it difficult to keep up to date.
Being able to have our code open-source, and working in a semi-professional manner like real-world programmers.

Teaches organisation skills- less last minute work.

It provides an idea of what software development is like instead of just teaching concepts

This unit aims to prepare us for working in the software industry, which is exceptionally useful in terms of giving us the experience of working with a real-world client and using real-world software engineering practices like Git and proper workflow (team communication, creating branches, pull requests etc). I think it's invaluable and this experience will serve us well in the future.

Client interaction

Letting us take ownership of the prokect

Learning about project management. I think it's really beneficial for students to have this experience prior to entering the workplace.

Good practical experience; project management etc.

It is an interesting unit that is quite fun. Provides some useful insight into the real world jobs of comp sci

I found the agile scrum methodology interesting and the tactics for project management very sensible. The discussion on ethics interesting as well. Git and the intellectual property lectures were will done as well.

Interacting with the mentor and interacting with a client entity (especially one who is not very tech savvy). Much of the lectures were good also.

Experience with having a client and working as a team to meet deliverables.

Learning about the agile methodology. Gaining useful knowledge that can be applied to real world scenarios. Getting better at working in group environments, especially larger groups.

The fact that meetings and planning are major parts of the unit, as in other CS units those elements were not assessed, meaning that any time spent on those was time 'wasted'.

I have done ALL the coding work despite trying my best to make it easy for others to contribute. They just know NOTHING, enums, callbacks, compilation just to name a few are foreign concepts. Wth!?! How is that even possible for 3rd years. They are _useless_ in terms of writing code, and it would take me months to make them even remotely productive. So the idea that groups should have a mix of 'gun coders' and 'not so gun coders' is a bad one, it causes the good students to either do poorly because they don't put in 6 * 60 hours, or very well in this unit but bad in others for the same reason. Rank students by WAM and form groups. Let the dump people who wouldn't get jobs in CS anyway fail like they should... If they actually knew how to code they'd have a better WAM.

Nothing so far, I am sorry Michael. I am struggling for this unit as my teammates are using the knowledge from Agile Web Development and I don't know anything about that.

Great real world experience, it's a rarity in uni study.

Q7 - Are there things that can be improved, and if so, what?

Are there things that can be improved, and if so, what?
Probably, but I don't think I'm knowledgeable enough to accurately point out what or how to fix it
Reduce the amount of meetings.
More feedback from the auditors/markers
Proper lectures and proper learning outcomes to be assessed. No slavery based model. Would prefer a fair unit where everyone has the same project and marks are based on something concrete.
More lectures focused on how to effectively code as a team More of a structured project description from clients as time was wasted trying to get in contact with the client and figuring out what they actually wanted More resources/real examples of how the agile methodology is conducted
I think there is no need for the "mentors". The auditors provide that sort of guidance anyway, and I feel there is already enough administrative load with the regular meetings with the auditor and client (and the publishing of their minutes), time-sheets etc.
Ensuring reasonable guidance and requirements are provided by all clients.
I find working with so many people challenging due to the requirements to have everyone meet twice a week averaged over the semester. I appreciate the leniency given in the first few weeks, but it was initially very stressful needing to arrange the client, auditor and mentor meetings in the first week - due to scheduling issues we met most group members for the first time at the first mentor meeting. I appreciate what this unit is trying to achieve, but it was by far my most demanding unit when I acted as project manager. I think the timesheet should've been explained more clearly because I was unsure about when to include tasks, what the estimated time values referred to, etc.
I feel as though my team is not balanced as there are couple people including myself doing the vast majority of work while some people are essentially really heavy, don't go for meetings, don't do their one job of documentation right or shown no attempt at all despite delegation of tasks and offering to teach how to however my attempts to communicate with them have been ignored. It infuriates me a lot as although I love the work given to me that I feel overwhelmed covering the parts that my teammates refuse to do or fail to submit upon deadline thus penalising the marks for my hard work. Love the unit, but this has been he worst case of team work I've ever experienced.
Maybe an "attendance counted" weekly group meeting times in CAS or in the unit's weekly timetable. So that everyone will come to the group meeting and there will be no conflict with other Units. I know there is a supervised meeting for every two weeks, but it is not frequent enough for other members to keep up with the team. Although we use social media to communicate, it is better to meet everyone in person.
Some more help on how to split the workload could have been handy.
1) Most, if not all, the projects are based on UI development (and using, for example, HTML or App development software), which some of us have not covered before (Data Science majors have (through Agile Web Development), or people who have outside experience - but many do not have these). I'm not sure if this was intentional, or the reasons for this, or perhaps this is the nature of these kinds of projects. Being unfamiliar with this kind of project makes it difficult, which is not necessarily a bad thing but makes things more challenging to plan and to do. 2) The website is difficult to navigate, especially to find what we need to do and submit, and the documents for those. It may be helpful to consolidate all assessment material into one place, or to make assessment details more obvious (for example, the essay link is buried within the timetable and not obvious to find).
It would be great if the proposed projects were of similar difficulty and we had people with computer science/software engineering backgrounds to consult with on the actual projects themselves.
This years essay topic was a bit strange, the only place the ACS CoE applied to was the people who develop AI systems, it has nothing to do with influencers
it seems good

There is no incentive to attend lectures, I'm not sure if the unit coordinator noticed personally as he wasn't there but attendance dropped off sharply after the first few. There is no exam for this unit, and while the lectures relating to the ethics of computing would have been useful for the individual essay, almost all of that information could also be found online. The only useful lecture (IMO) was the one with David Glance (?) where the topic was about using Git, GitHub and the workflow for developing a software project. Alex Reid's lecture was marginally useful in terms of essay preparation and the ACS Code of Ethics. The others, not so much. Sure, they were informative and interesting, but certainly not a priority to attend as we had other units to focus on, not to mention the project which was actually worth something. I just watched them much later on in my free time. Either remove the lectures altogether or make them all recorded.

N/A

There was too much going on around the beginning of sprint two (the sprint, then reflection then essay) - I think sprint two should have been made longer to account for this, then sprint 1 made shorter

I think our project in particular is very difficult, so our challenges are limited to our group only. I would say that the unit is run brilliantly all things considered, but we were just unlucky with the requirements of our specific project. I think that maybe the difficulties of the projects need to be standardised because ours is definitely not feasible within the Semester. But honestly I sincerely appreciate all the way that that the unit is structured and the parties involved.

there should be a capacity to have some choice in the members of the group. While it might be your responsibility to offer fair assessment (as you say not to createpauper groups), itis not your responsibility to make sure poor quality studentshave a good student to carry them through the unit, pauper groups are not an issue, as it is not your responsibility to make sure the students sitting the assessment try and are intelligent, that is not under anyone's control but their own.

Have meeting times set for the unit just like labs have them pre allocated. Lower the amount of work required for the unit and maybe make some sprints longer. They can get a bit short and hard to complete on time with other units and mid sems

The lecture on team dynamics felt over the top and forced. Doing the candy search went horribly wrong and the lecture overall felt pointless. Ethical issues with Alex Reid had so many case studies but he did not break down the ethical issues for each case. The cases were put to a survey vote of the room which don't really give any insight apart from what the majority thinks on an issue. We shouldn't be taking time out filling surveys if there not expanded upon for greater insight. I don't feel there's enough accountability for the blurbs on the project offering page. Some of the descriptions are very vague and it feels like there's not enough time to clarify those descriptions. Maybe you could ask students to begin looking at the projects before the semester starts and submit questions to the forum. I also disagree with the random assignment. My team only had two people in the team present on the first lecture, myself included. This ment that we had to start deciding what project we were to take without knowing the capabilities or preference of the other members. By the time we could get into contact with the other members it was too late. This felt like every other group was already ahead. This has a flow on effect as the other group members feel less motivated as they didn't own the decision. I think the random selection should take into account if the students are present or not on the first lecture. The absent students should be assigned later on.

Yes, there are. The individual marks allocated for the project do not correspond nicely to the required task e.g. 5% for a 2 paragraph reflection does not seem appropriate. Additionally, we are required to have way too many meetings - sometimes 3 in a week. This is compounded by the fact that these do not always occur on campus.

A lot of the project is very specific and does not ultimately coincide with every project - for example our mentor and team agreed that our project is very difficult to split with group tasks and everyone coding - so we have had to work around this and create new roles and new tasks that people can work on instead.

Longer sprints, currently they feel far too short. Could be improved by using the weeks that are not currently included in sprints (study break week and week essay was due). Better organisation of the unit website. It is confusing sometimes to find certain content. Eg: It would've been nice to have all the essay and its related materials in it's own category in the side bar.

I understand the lectures were set as compulsory for respect of the speakers, however i found 3 lectures a week too much, even though i wish i could have gone to them all (topics were interesting)

The group timekeeping spreadsheet is hard to understand. The unit website can be better organised, for example assessments should not be spread out over several pages (there is assessment information on the home page, project page and timetable page).

Team allocation logic is horrible. Sort students by WAM, take 6 relatedly. That way smart people don't have to do all the work to because the 'dummies' they get lumped with can't code, write documentation or do anything really...

Team up freely. Arrange significant units like Agile Web Development in the second year.

I know it's a lot more difficult, but standardizing project difficulty. Our group seems to have hit one of the harder projects as well have to comply with UWA IT standard if our project is to be used, other groups finished a week or two ago.

Q8 - 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...
CSDN which is a Chinese website
w3schools is the best site for learning beginner web development skills
Slack (online team communication), GitHub.com, GitHub Desktop (managing local repositories and interactions with the 'Origin' on github.com), Google Drive (shared non-code group files, for admin and reports etc.)
Nothing general aside from StackOverflow.
General API documentations specific to my projects cloud services, forums, etc.
If using Google Cloud based products (for databases, APIs, cloud computing, etc), it is well documented on how to use them. If choosing which service to use, take a look at their product description and documentation (some might have actual code examples) before deciding to use them.
StackOverflow GitHub Microsoft tutorials Youtube tutorials
Ahhh... the mentor was the best resource, that and Google. SCRUM video was good. TBH didn't really use the resources, just learned from other people.
Online tutorials for web development: https://www.w3schools.com/
Stackoverflow and other similar sites have their obvious uses, but I personally like to use the different language manuals open or downloaded. For me Zeal is a good app to keep all the documentation I need downloaded and in one place.
i mainly used google, no specific websites were notably better than others
The project requires a LOT of self-study. None of the frameworks required for the assigned project our team has used so far has even been taught in the preceding CITS units, which means that all of us had to basically learn several new languages/frameworks in a short timespan and apply it to the project. There were too many resources to list, but in general we looked at the official documentation for the frameworks used e.g. Vue, Vuetify, Node.js etc. YouTube videos were also helpful.
Forums and coding websites (Stack overflow, Microsoft documentation for C#)
Udemy courses were needed for our project, but just speaking to the mentors have helped immensely.
https://guides.github.com/activities/hello-world/
YouTube: Justice with Michelle Sandel. Very useful for understanding ethics.
https://lab.github.com/

Trello or Jira for project management. Google Drive for storage of sprint deliverables not suitable for Github eg: Most if not all of Sprint 1's deliverables. Google Docs for collaborating on documents. Slack for communication between members.

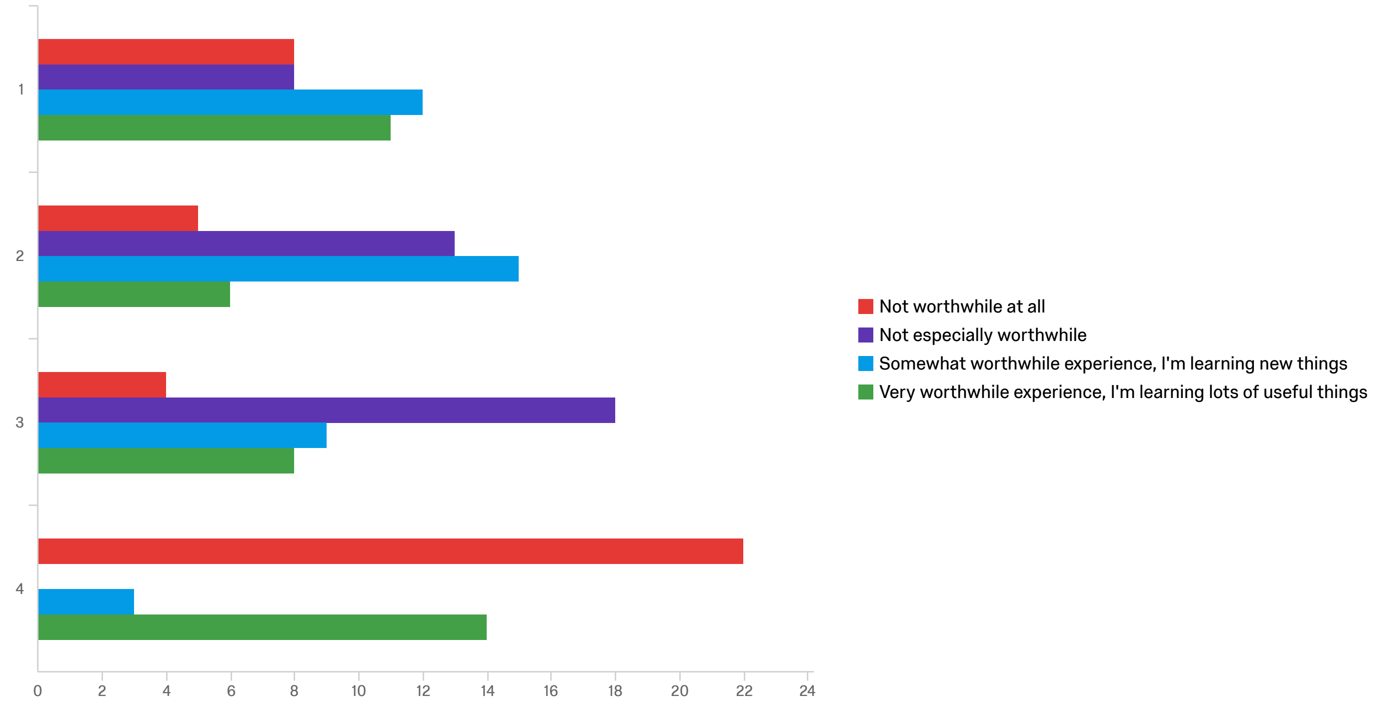
Seems like most projects are web dev based, with a lot of students havung little to no experience with nodejs etc

Bootstrap is useful to avoid writing a lot of CSS.

Stackoverflow

Almost all of my outside knowledge is via Udemy, but it's a paid service. failing that, docs are always good.

Q9 - Overall, I feel that the unit is:



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Not worthwhile at all	1.00	4.00	3.03	1.23	1.51	39
2	Not especially worthwhile	1.00	3.00	2.26	0.78	0.60	39
3	Somewhat worthwhile experience, I'm learning new things	1.00	4.00	2.08	0.92	0.84	39
4	Very worthwhile experience, I'm learning lots of useful things	1.00	4.00	2.64	1.23	1.51	39

#	Field	1		2		3		4		Total
1	Not worthwhile at all	20.51%	8	12.82%	5	10.26%	4	56.41%	22	39
2	Not especially worthwhile	20.51%	8	33.33%	13	46.15%	18	0.00%	0	39
3	Somewhat worthwhile experience, I'm learning new things	30.77%	12	38.46%	15	23.08%	9	7.69%	3	39
4	Very worthwhile experience, I'm learning lots of useful things	28.21%	11	15.38%	6	20.51%	8	35.90%	14	39

Showing rows 1 - 4 of 4

End of Report