Individual Instructor Report Fall 2022 Version A for C S 373 - SOFTWARE ENGINEERING (52945) (Glenn Downing)

Project Title: **Instructor Course Evaluations Fall 2022**

Courses Audience: **56**
Responses Received: **46**
Response Ratio: **82.1%**

Report Comments

Results were produced during the Fall 2022 implementation of new course evaluation questions and systems. Courses with final exams conducted on December 8th, 2022, may have overlapped with the evaluation window.

Guide to the Interpretation of Course Evaluations at UT Austin
The goal of course evaluation process at UT Austin is to drive teaching excellence and to support continuous improvement in teaching and learning experiences. The two sets of scales used for core evaluation questions and the associated weights are:

Strongly Agree (5)
Agree (4)
Neutral (3)
Disagree (2)
Strongly Disagree (1)

Excellent (5)
Very Good (4)
Satisfactory (3)
Unsatisfactory (2)
Very Unsatisfactory (1)

The Mean is calculated by adding all of the weights for a single question and dividing by the number of respondents. The course workload question is not averaged.

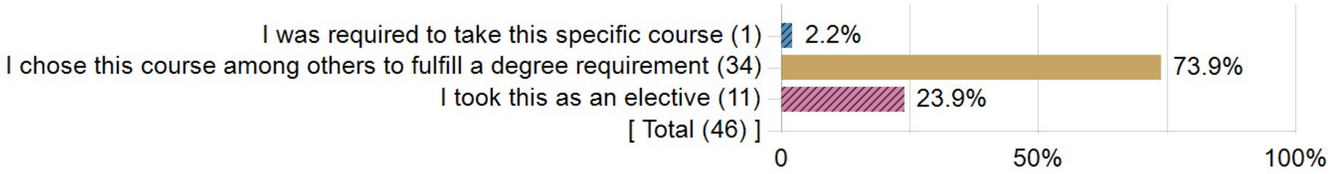
The number of students (e.g. Respondents) marking each option is reported for each of the items. These frequency distributions provide information about the level of student ratings and the spread and shape of the class distribution of responses. The distributions thus provide a picture of student perception of a course.

Course evaluations provide snapshots of student perspectives on their course-level learning experiences. Most experts on teaching evaluation advise that no one method gives the complete picture of an instructor's teaching effectiveness; multiple and diverse measures, on multiple occasions, are advised to give a full picture of the teaching effectiveness of a particular instructor. Moreover, other factors, such as size of class, level of the class, and content of the course, can cause small variations in the ratings. Therefore, student perspectives for a particular instructor or course should be interpreted as a snapshot, and as providing complete information on the teaching effectiveness of that instructor.

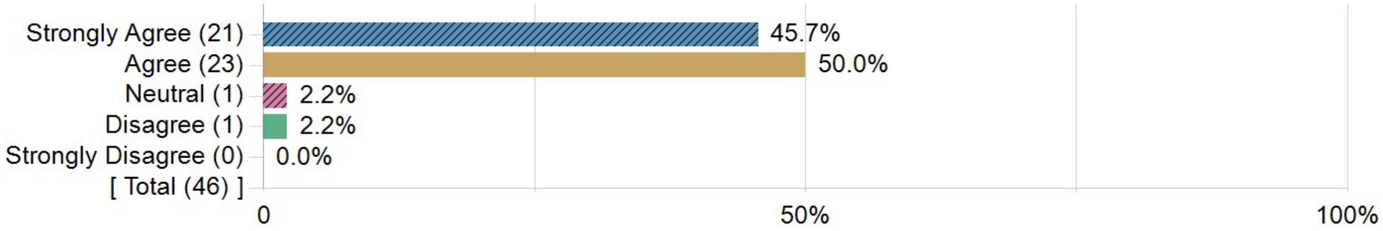
Creation Date: **Thursday, January 5, 2023**

Course Questions

Why did you take this course?

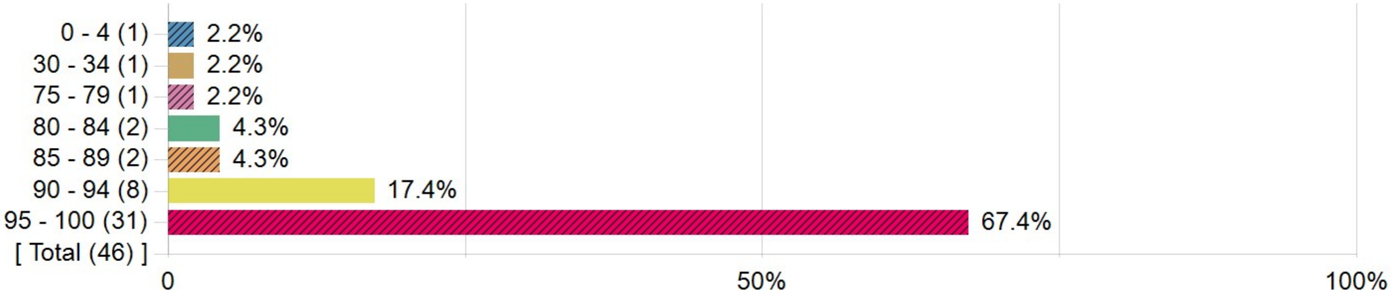


During this course, I gained a deeper understanding of the subject matter.



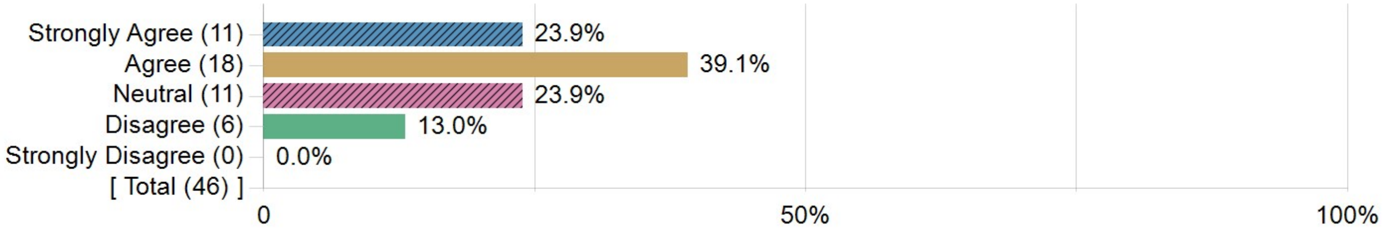
Statistics	Value
Mean	4.39

Overall, approximately what percentage of the course meetings did you attend or complete (online, in person, or asynchronously)?



Statistics	Value
Mean	90.87

The course assignments improved my ability to express my ideas in writing.



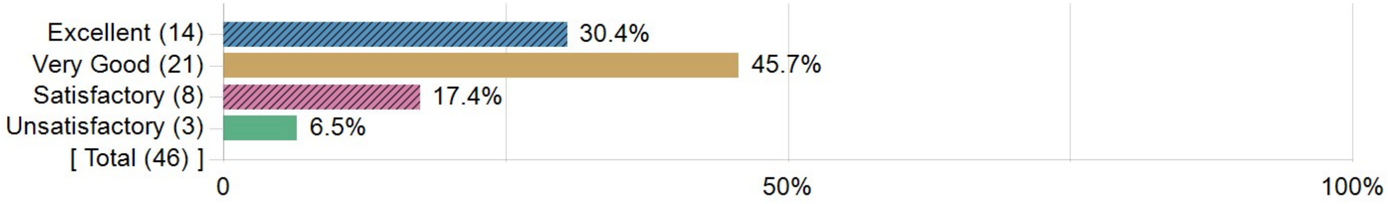
Statistics	Value
Mean	3.74

Instructor Questions

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	54.3%	45.7%	0.0%	0.0%	0.0%	46	4.54
The instructor fostered an inclusive learning environment.	58.7%	34.8%	2.2%	4.3%	0.0%	46	4.48
The instructor effectively explained the concepts and subject matter in this course.	47.8%	37.0%	10.9%	2.2%	2.2%	46	4.26
The instructional techniques kept me engaged in learning.	54.3%	39.1%	2.2%	4.3%	0.0%	46	4.43
The instructor checked for student understanding of the concepts presented in the course.	60.9%	37.0%	0.0%	2.2%	0.0%	46	4.57

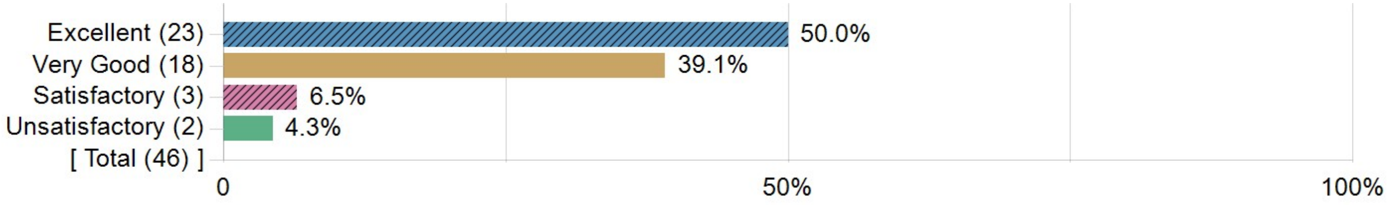
Overall Questions

Overall, this course was



Statistics	Value
Mean	4.00

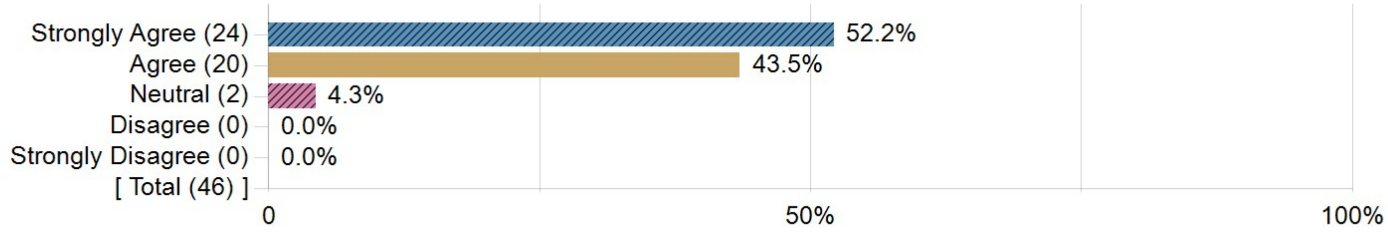
Overall, this instructor was



Statistics	Value
Mean	4.35

College, School, or Unit Questions

The instructor fostered a classroom environment in which all students could feel free to participate fully.



Statistics	Value
Mean	4.48

Comment Questions

Identify aspects of the course that were the most effective in helping your learning.

Comments
The clear grading rubric, technical reports, designating project leaders for group projects.
N/A
Exercises in class
I found the exercises to be the most effective. They helped me get hands–on experience with the subject matter.
Group collaboration. Lectures. HELP SESSIONS
Lecture time
The most effective aspects of this course were the exercises that required students to put into practice what they had just learned. The exercises helped me retain information better and understand the concepts.
The Socratic method Dr. Downing uses for teaching was very effective at helping me pay attention.
I enjoyed working on the team project and I liked the quizzes.
I really enjoyed the practice problems we would do in class and how he would explain them at the end of the class.
Quizzes "forced" me to attend classes.
I think the project was the most helpful to learning
Deep dive into Python during lectures was very informative. The project was a very real learning experience across the whole stack.
The IDB project was extremely effective.
The lectures consistently readdresses topics and poses questions for the students to answer, so it really makes the concepts really rememberable. Additionally, the way the lectures are delivered and the overall structure of the course actively encourages participation from the student and prevents moments where the student can procrastinate and skate by the class while still allowing for some wiggle room with forgiveness policies.
The interactive nature of the lectures kept me engaged and following the ideas step–by–step.
The projects for sure.
This course helped me improve my comprehension of documentation and API knowledge. Overall t really just helped refine my self learning skills.
There was a lot of support from the TAs, though I often did not feel the need to use them. The professor's lectures were clear and I HIGHLY appreciated that there were online recordings to refer back to.
cold calling
The exercises helped a lot in learning the course material
The cold–calling was absolutely key to my learning in the class – it kept me focused, engaged, and willing to understand the content at all times.
I think working in groups made it a lot easier to learn all of the new frameworks.
Learning through doing was by far the most helpful aspect of this course. Lectures over React, Flask, etc.. are simply not as effective as just going out and doing it yourself. It was also extremely helpful to work in a team and have access to previous group's submissions. This reflects the flexibility and communication aspects of real–world software engineering.
I appreciated the cold calling more than I thought I would.
Lectures were filled with content but in consumable chucks since the class period was 1hr. Tutorials on how to use certain tools and frameworks was helpful.
I enjoyed how well the professor lead the conversation during Socratic method moments. I was afraid of being exposed for not being quick on my feet when called on, but he made it easy for me. Having him go over specific code examples in front of me was excellent. I learn best from practical methods rather than theory, so it was very good for me.
I thought the python lectures were very interesting, and I learned a lot of interesting features in python.
I think the aspect of the course that was the most effective in helping my learning is the projects, since it promoted many good software engineering practices.
The course projects were by far the place where I learned a lot. Most of the lecture content I only retained for a few days – enough to get me beyond the quiz that would cover it. I learn most by doing, and since we didn't do many assignments beyond the rushed in–class exercises related to the lectures, I did not finish the course remembering what we covered in class throughout the semester.
I liked the back–and–forth style of lecturing with periodic participation from students. This allows us to learn a great deal of the course material while also filling in any cracks of knowledge anyone may have.

Identify the aspect of the course that you found most challenging, why you found it was challenging, and suggest one thing that could be done to help future students meet that challenge more effectively.

Comments
I found learning the topics that were outside the scope of class the most difficult because it while there were links to for tutorial to learn it, it still was difficult to learn it in an unstructured format as opposed to the classroom learning experience. It's a skill of learning how to learn that is difficult to teach to each individual because it is a unique skill for each person.
I found the disconnect between lectures and projects challenging. It was challenging because I feel like there should be base line on how to do the projects and not just throw us in the deep water. I feel like it would help future students if you were to teach them about MySQL, hosting on AWS, react, and javascript and not only teach python which we barely used after project 1. Also I think cold calling should be replaced with participation points with speaking and asking questions rather than just randomly being called on.
Learning new tools with no assistance. An intro lecture on these topics
Blogs and Readings ended up becoming a monotonous cycle with very little learning value.
I thought it was very challenging to self–learn everything. It felt like I was taking 2 courses in one. I wish we had more support in class to learn about how to use React and other front–end tools instead of just learning about data structures, which we've already taken a course over.
Daily Quizzes. Too tight of a time constraint even though there were many classes where we ended up leaving earlier than scheduled. Hard to measure knowledge with too little time to process the question
Project, some technologies could be introduced to students
I found the projects to be the most challenging due to the great amount of tools I had to learn on my own. One thing that can be done to help future students is to incorporate more of these tools into lecture time and provide more resources to help learn these tools.
There were plenty of parts my team had trouble on, and that I was unsure of, when working through the project and the first assignment. I'd definitely recommend that students feel free to ask questions to the TAs and in class discussions.
One part that was challenging was the quizzes and I think that it would be nice to have practice questions in class.
One aspect that I found the most challenging were the timed quizzes at the beginning of class. I think having test anxiety didn't help with the fact that we only had 3 minutes to complete them since I need to read a question a couple of times to understand it.
The IDB project was challenging since we were expected to learn a lot of crucial concepts on our time outside class with little guidance. As a team with very little experience in web development, this was very frustrating.
The daily quizzes were pretty challenging. I thought the questions were made tricky on purpose, and because of the way the course is set up right now, even if you do well on other categories the quizzes could end up tanking your grade.
Learning how to deploy. basically had to self teach by googling and hours of trial and error. Maybe have a high–level overview of how deployment works, from domain name management to server hosting to database hosting; though these can't be too specific to allow us freedom to try different things like AWS or GCP and even the multiple possible solutions within those platforms. I think a big part of the value of the course is figuring out things ourselves without guidance.
A lot of quizzes had questions that were kind of complicated. Since we only had 4 minutes there wasn't really time to think through questions. This was unfortunate especially since these quizzes were for attendance. I would suggest either giving us slightly more time or making the quizzes a bit simpler.
The aspect of the course that I found challenging was working with a large group of students and learning how to coordinate and assign tasks to different members of the group while ensuring everyone was participating. This was challenging because there was no effective structure in place to penalize the students who were completely reliant on other group members to finish their work, so it felt like being randomly placed in a certain group could end up more punishing rather than rewarding. One thing I would suggest is to just add some additional metric that punishes those who freeload in the projects.
The project was the most challenging part, but I think we were given all the tools to tackle it effectively.
At times I felt the lectures left something to be desired. Much of our daily lecture time was devoted to administrative issues (in class quizzes, in class exercises), to the point we often weren't getting much real lecture time at all during certain periods of time. The lecture content was also pretty irrelevant to the projects as a whole. I probably could have done well in the class without going to the lectures even once (minus of course the required assignments, such as exercises, that took up much of our lecture time). Overall I don't feel as though I actually learned anything concrete from the class that wasn't self–learned through doing the IDB project. A lot of the content (first half of the semester), was copied from Prof Downing's OOP course, which felt very gross to me and as though I wasn't getting my money's worth for my tuition between these two courses. I honestly think the lectures should focus more on the type of things we are doing in our projects, or at the very least project development concepts such as those you would need in a real–life environment. As it stands, it just feels like an intro Python class with the IDB Website Project tacked on to the side.
The thing I found the most challenging was having to work with a completely random set of team members with different schedules. I would suggest trying to reach out to your team as early as possible to make sure everyone is on the same page.
There are just a lot of deadlines and assignments that albeit the fact that they're simple, they just feel a tad unnecessary (the blogs

Comments
and occasionally the papers). For our main project, every phase's instructions got combined with the previous ones which made them kind of difficult to parse
project
The disconnect between the projects and the class material was very challenging, and I believe there should be a bit more content covered in class itself going over the general basics of web development.
Some of the quizzes were rather difficult, even when I reviewed the class notes pretty heavily. I actually wouldn't mind if we had fewer attendance quizzes if the quizzes themselves were more manageable to review in 4 minutes. Occasionally, I wouldn't be able to confirm my answers were correct and missed points that I wouldn't with more time.
I think the most challenging thing about CS373 was being able to successfully communicate between a large group of five people. It is difficult to begin the project when nobody is certain of any other group members' capabilities. I think that prior experience in certain technologies that was polled via CATME at the beginning doesn't necessarily reflect their abilities to write code for this specific project. In my group's experience, our prior abilities were all very limited despite the fact that we had all chosen varying answers for our level of prior experience. Therefore, I would pair students not by their prior experience with react, SQL, etc.., but instead with their overall experience level with programming as a whole.
The projects were extremely difficult, because my group would never do their part and I had to finish each phase by staying up and working for 3–4 days straight for the last few phases.
The self–learning required to complete portions of the semester project were often challenging. Having different TA's specialize in different sections of the semester project would be a good use of our time and theirs.
Lectures were completely unrelated to the projects. Lots of the concepts taught were rushed, explained unclearly, and or outdated.
Learning everything that we were not taught for the projects was definitely to most challenging. I don't think the professor needs to do more to teach us those subjects, but doing more to point us in the right direction would be nice. For example, giving us links to useful tutorials and solutions to common beginner problems would be nice. I also struggled a bit with the daily quizzes. I lost a lot of points over syntax and things that would take one simple google search to check. I'd prefer if he had given us syntax reminders on the quiz and just asked us to work through a problem.
I think it is hard to be expected to attend every class, especially with COVID still spreading around campuses.
I think the aspect of the course that I found most challenging is also the projects, since there was little guidance on how it should be done. I think this is actually a benefit for future students, because being able to self–teach all these technologies helps a lot in the future.
The disconnect between the course projects and the lecture content. I honestly found it beneficial that we didn't cover the specifics of the projects in class, though talking about full stack engineering and CI/CD would've been helpful. Not spending time outside of lecture on the lecture content was detrimental.
The independent project was quite daunting. I think it was handled relatively well, but it would be nice to get some more support with our tech stack and maybe more TA supervision to make sure we don't go in a completely wrong direction.