Evaluating Group Work in Large CS Classes

Joël Porquet-Lupine

2022 Illinois CS Summer Teaching Workshop – August 15–16, 2022

UCDAVIS
COMPUTER SCIENCE

Motivation

UCDAVIS
COMPUTER SCIENCE

Group work in ECS 150

- ECS 150 (“Operating Systems and Systems Programming”)
  - Core class in CS curriculum
  - Offered two/three times a year, enrolling ~200 students/quarter
  - I have taught this class 11 times since WQ17
  - Typically get three TAs (20-hour/week appointments)
- Considered one of the hardest classes in the CS curriculum
  - Lots of difficult, abstract concepts
  - Complex programming
- Assessment on two different aspects of the class
  - 50% on “theory”, via written exams (i.e., midterm(s) + final)
  - 50% on “practice”, via two-student group projects

UCDAVIS
COMPUTER SCIENCE

Benefits of group work

Large-scale projects are the norm in the computing industry

- Provide students with more complex, interesting projects
- Help them build many of the soft skills needed in industry
  - Organization, cooperation, communication, confrontation of ideas, etc.
- Refine understanding of class material
  - Explaining to others, being explained by others

- Upside of instructors too!
  - Less staff-provided support required (e.g., office hours, online forum)
  - Improved quality of grading if fewer project submissions

UCDAVIS
COMPUTER SCIENCE

Group work struggles

- Finding a good partner…
- Working together efficiently…
- Grading group project fairly
  - Many complaints about group project grading being unfair

    “My partner didn’t do anything, yet they got the same grade”

  - Project submissions themselves don’t include reliable evidence of individual contributions (*and contributions to a project is more than code anyway*)
  - Need a way to capture individual contributions better!

UCDAVIS
COMPUTER SCIENCE

Bimodal group work assessment strategy

- (1) Oral interviews
  - Individual assessments conducted by TAs
  - Once, at the end of the quarter
- (2) Student reporting
  - Self and peer evaluations directly from students
  - Completed after each project
- Equally factored into class final grade
  - 50% of final grade on “practice”, via group projects
    - 80% for project submissions (same grade for each partner)
    - 20% for interviews + evaluations (individualized grade for each partner)

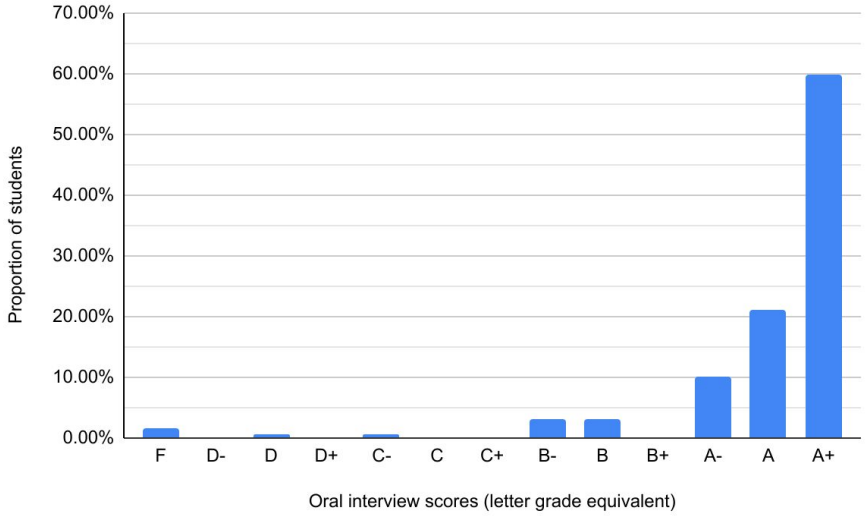
UCDAVIS
COMPUTER SCIENCE

Oral interviews

UCDAVIS
COMPUTER SCIENCE

History

- Actually started using oral interviews since WQ17
  - At first, on the entire class after every single project!
    - Partners evaluated together, 10 minutes per group, but individual scores
  - Then, on half of the class every other project
    - Reduce the extra work for TAs
- Only one rubric, not very well-defined…
  - Difficult to assign individual scores when partners are interviewed together
  - Grading unreliable
  - Student frustration still running high



Current form (since WQ20)

- During last week of the quarter (week #10)
  - Last group project is due at the end of week before (week #9)
  - Last group project is fully auto-graded to free up TAs
- 10 minutes per student with a TA
  - 30+ hours worth of time slots available!
  - Registration for time slots announced in advance and opened at the same time for everyone
- Students quizzed on two projects (among 3)
  - One of which they choose, one randomly selected in the moment
  - One or two main questions per project (+follow-up questions)

Grading rubric (excerpt)

Q2.1 Understanding of concepts
32.0 points

1 **+32.0**

Exemplary level (A+ - A)
Understanding of the concepts is **excellent**.

- Concepts covered by projects are perfectly understood and can effortlessly be explained.

2 **+28.0**

Accomplished level (A - B)
Understanding of the concepts suffers from **a couple of important flaws**.

- Most of the concepts are well understood, but difficulty to understand/explain a couple.

3 **+24.0**

Developing level (B - C)
Understanding of the concepts suffers from **a few**

Q2.2 Understanding of code
40.0 points

1 **+40.0**

Exemplary level (A+ - A)
Understanding of the code is **excellent**.

- Submitted code is perfectly understood and can effortlessly be explained (even parts that the student didn't write themselves). Student barely looks at code, only to remember very specific details if necessary.
- Student is very comfortable talking about the high-level design of the code, as well as the code's internals.
- Student understands limitations of their code and can discuss them.

2 **+36.0**

Accomplished level (A - B)
Understanding of the code suffers from **a couple of important flaws**.

- Student is very comfortable with most parts of the code, but has minor difficulties explaining a couple other parts. Typically needs to often refer back to the code.
- Student understands well the high-level design of the code, but has minor

Q2.3 Communication
24.0 points

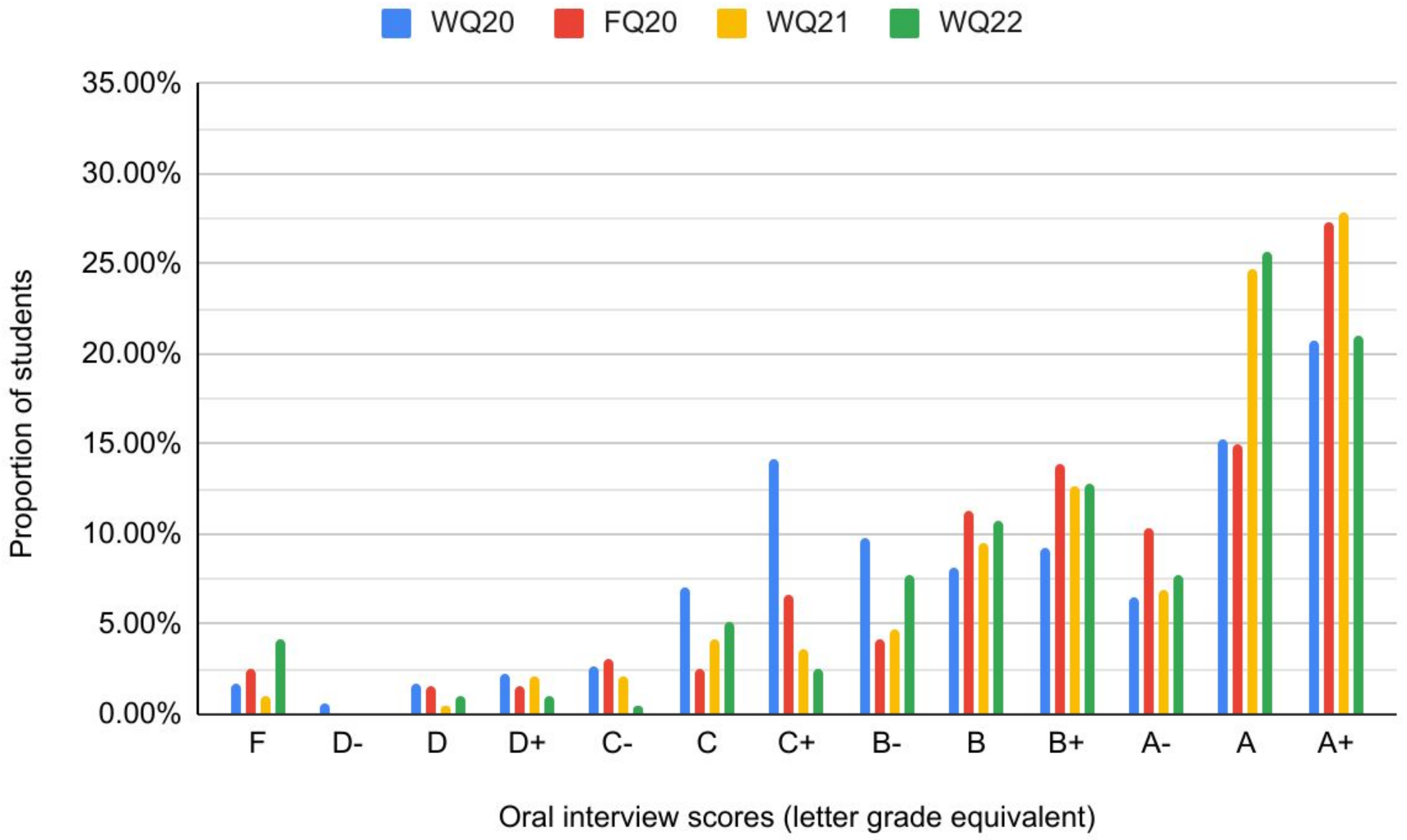
1 **+24.0**

Exemplary level (A+ - A)
Communication is **excellent**.

- Content:
  - Accurate, thorough, and directly on point.
  - Central idea/purpose vividly stated and supported.
  - Well-structured sentences in effective sequence.
- Verbal delivery:
  - Free of errors in grammar.
  - Word choice helps with clarity.
  - Varied and dynamic, enhanced by speech rate, volume, and tone.
- Non-verbal delivery:
  - Eye contact.
  - Good posture and appropriate attire.
  - Helpful gestures to support presentation.

UCDAVIS
COMPUTER SCIENCE

Statistics



UCDAVIS
COMPUTER SCIENCE

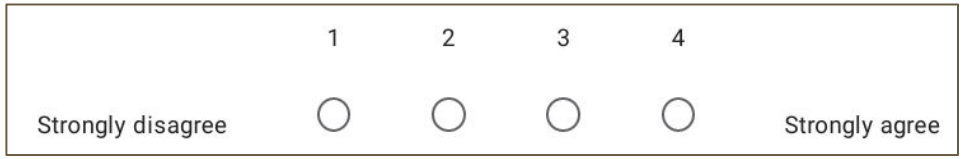
Self and peer evaluations

UCDAVIS
COMPUTER SCIENCE

Evaluation form

- Completed by each student, after project deadline
- Open for about a week
- Two “facets” of group work
  - 1. Engagement in the project
    - Qualitative self evaluation
    - Qualitative peer evaluation
  - 2. Contribution to the project
    - Quantitative contribution

Engagement in project



- Organization
  - *You had a role in the clerical organization of your group. For example, you helped define the terms of your collaboration: how/when/where you should meet, how you should work together, etc.*
- Communication
  - *You had a role in the communication of your group. For example, you helped maintain a constant communication with your partner throughout the project.*
- Cooperation
  - *You were willing to listen and respect the ideas of your partner, and discuss the work distribution. For example, you would not try to always impose your way of doing things.*
- Attitude
  - *You showed positive and enthusiastic attitude, and it was pleasant to work with you.*
- Contribution of ideas
  - *You contributed ideas to the project in terms of how to tackle the assignment, structure the code, build certain algorithms, etc.*
- Contribution of code
  - *You participated in the programming aspect of the project.*

UCDAVIS
COMPUTER SCIENCE

Contribution in project

How would you quantify your and your partner's respective contribution to the project?

- 0% – 100% [0.0] ⇒ *Your partner did (almost) everything while you didn't do (almost) anything*
- 25% – 75% [0.25] ⇒ *Your partner contributed substantially more than you*
- 50% – 50% [0.50] ⇒ *You and your partner contributed (almost) equally*
- 75% – 25% [0.75] ⇒ *You contributed substantially more than your partner*
- 100% – 0% [1.0] ⇒ *You did (almost) everything while your partner didn't do (almost) anything*

UCDAVIS
COMPUTER SCIENCE

Deviation scores

- For each facet, data points are averaged across all the projects, for both the self and peer evaluations
- A “deviation score” is computer for each facet, by averaging the variations between the self and peer evaluations

	Organization	Communication	Cooperation	Attitude	Contribution of ideas	Contribution of code	Engagement average	Engagement deviation	Project contribution	Contribution deviation
P1 - Self	4	4	4	4	3	3	3.66		0.5	
P1 - Peer	4	4	4	4	4	4	4	-0.33	0.5	0
P2 - Self	4	4	4	4	4	4	4		0.75	
P2 - Peer	3	2	2	2	3	4	2.66	1.33	0.75	0
P3 - Self	4	4	4	4	4	4	4		0.5	
P3 - Peer	4	4	4	4	4	4	4	0	0.75	-0.25
Averages							**3.72**	**0.33**	**0.625**	**-0.0833**

- A positive deviation means the student may have **inflated** their self evaluations compared to their partner(s) reported
- A negative deviation is the opposite

UCDAVIS COMPUTER SCIENCE

Grading equation

- Average for each facet is adjusted with corresponding deviation
  - Positive deviations directly subtracted from average
    - Penalize students who inflated their evaluations
  - Negative deviations divided by two and added to average
    - Slight boost to students who underestimated their engagement/contributions

	Engagement average	Engagement deviation	Project contribution	Contribution deviation
Averages	3.72	0.33	0.625	-0.0833
Adjusted scores		**3.72 - 0.33 = 3.39 3.39/4*100 = 84.74%**		**0.625 + (0.0833/2) = 0.66665 0.66665/0.5*100 = 133.33%**

- Both facet scores averaged to give final group work score
  - 40% engagement + 60% contribution
  - Group work score not capped to 100%
    - Reward students who provided more than their fair share of work across most projects

Engagement (40%)	Contribution (60%)	Final score (100% –uncapped)
84.74%	133.33%	**113.90%**

UCDAVIS COMPUTER SCIENCE

Statistics

	Engagement Score Averages			Project Contribution Score Averages			Final Score Average
Quarter	**Score (out of 4)**	**Deviation**	**Adjusted Score**	**Score (out of 1)**	**Deviation**	**Adjusted Score**	**Grade**
WQ20	3.61	0.05	88.01	0.49	0.03	89.90	89.15
FQ20	3.60	0.04	87.95	0.48	0.02	89.69	88.99
WQ21	3.62	0.03	88.71	0.48	0.01	94.02	91.90
WQ22	3.62	-0.02	89.06	0.48	0.01	92.37	91.05
Average	3.61	0.03	88.43	0.48	0.02	91.50	90.27

Student survey

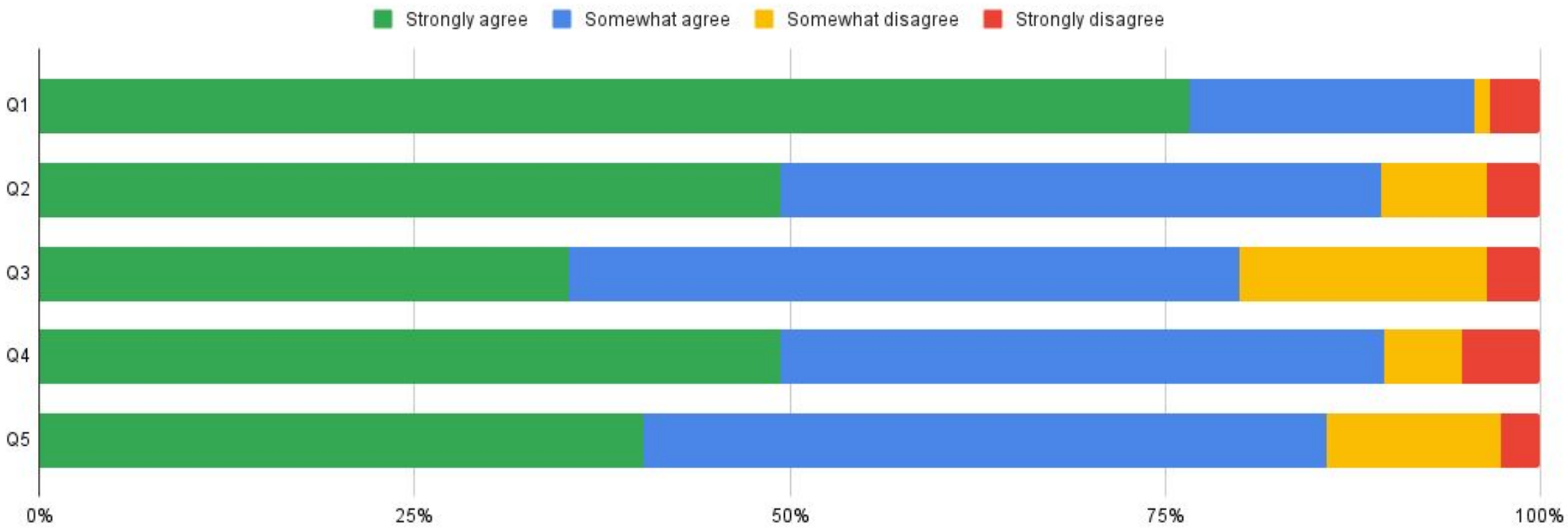
UCDAVIS
COMPUTER SCIENCE

Survey

- Sent to 590 students, received about 15% of responses
- Goal was to measure the agreement of students with our narratives, and the effectiveness of strategy:
  - Q1. I agree with the narrative that group work is unavoidable in CS. *[narrative]*
  - Q2. Oral interviews mimic how you may be held accountable in the workplace (e.g., having to explain work that your entire team will have produced). *[narrative]*
  - Q3. Oral interviews are an effective solution to fairly grading group work. *[effectiveness]*
  - Q4. Group work evaluations mimic how coworkers may provide periodic reviews of themselves and one another in the workplace. *[narrative]*
  - Q5. Group work evaluations are an effective solution to fairly grading group work. [effectiveness]

UCDAVIS
COMPUTER SCIENCE

Results



UCDAVIS
COMPUTER SCIENCE

Conclusion

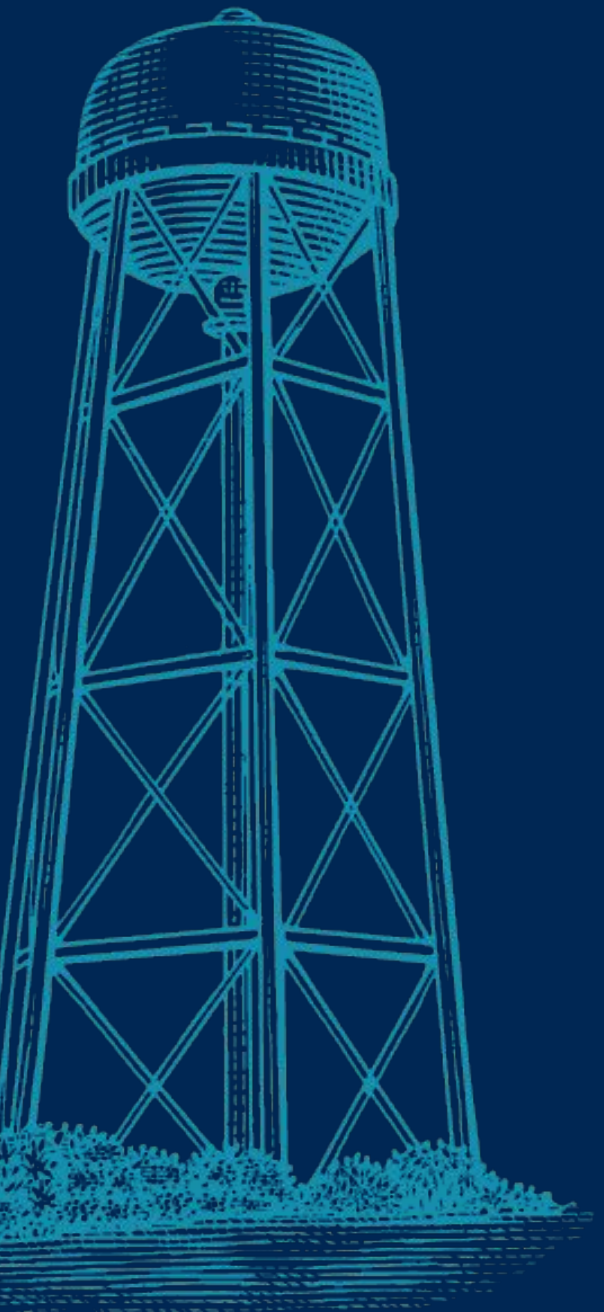
UCDAVIS
COMPUTER SCIENCE

Conclusion / future work

- Students still complain about group work in general ☠️
  - But less so about the fairness of the grading process!
- Good strategy to:
  - Gauge student understanding of their project submissions
  - Have students hold each other accountable
  - Lower the frustrations associated with group work in general

- Potential next step
  - Idea of an “impact score”, based on cumulative deviations
  - Reduce weight of peer evaluations if peer is suspected to be dishonest

UCDAVIS
COMPUTER SCIENCE



Thank you!
Questions?

jporquet@ucdavis.edu

UCDAVIS
COMPUTER SCIENCE