Collabor8

Making postsecondary CS education a welcoming and collaborative space

The Collabor8 Team

Trisha K.

Daniel R.

Juliette W.

Angela C.

“When starting out, being able to form connections and feeling a sense of belonging is crucial in staying involved.”

— *Sery*

Junior CS Major and CS intro class TA at OSU,

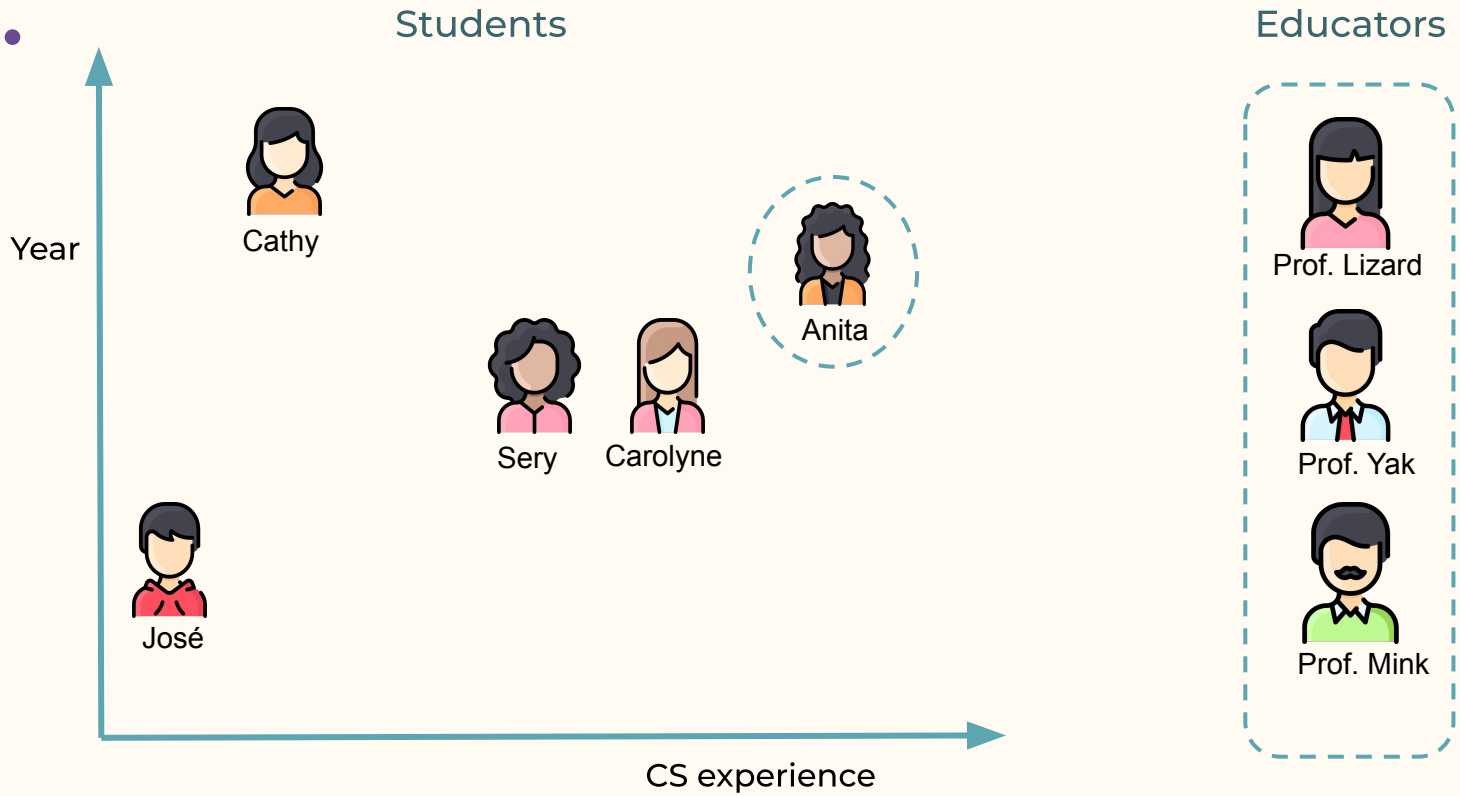
Initial POV

WE MET Sery, a Junior CS major and TA at OSU

WE WERE AMAZED TO REALIZE that her strong feeling of **community** in her intro class was crucial in deciding to do CS, yet she worked **independently** in later classes

IT WOULD BE GAME-CHANGING TO create a community and encourage **mutually beneficial collaboration** in (intro) CS classes

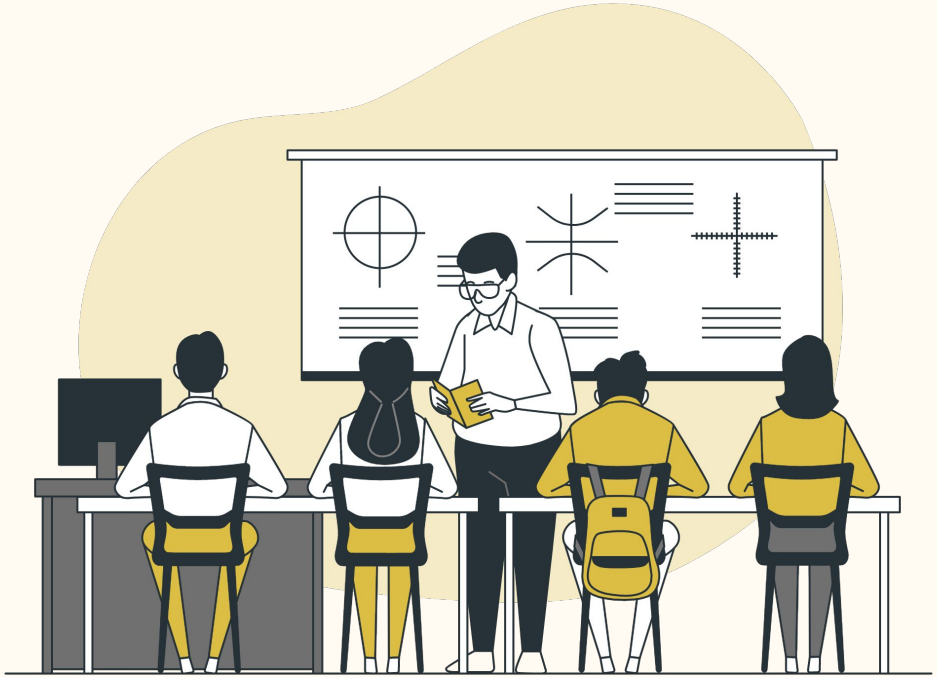
Additional Interviewees



“People here are really focused on the more computational type skills than they are their soft skills.”

– Anita

CS senior at Purdue, 3 internships



“This game theory didn't work: students do not think rationally.”

— *Prof. Yak*

ECE professor at a public university with 16 years of teaching experience

“If you can talk your way through that with someone and teach someone else how to do it... then that really solidifies your understanding of the material.”

– Prof. Lizard

CS lecturer at a private university

“At some level, you have to learn the basics for yourself to make sure that you're facile to contribute to a team.”

— *Prof. Mink*

Professor and Associate Chair for Education in CS at a private university

Collaboration is key to CS education regarding learning effectiveness, resources, and preparation for careers.

Students value collaboration that promotes **independent** growth and fosters **relationship** building.

POV 1

WE MET Cathy, a grad student at Stanford who just began taking CS classes last year with no previous CS background

WE WERE AMAZED TO REALIZE that she was most stressed by **not being able to start** because of tool trouble

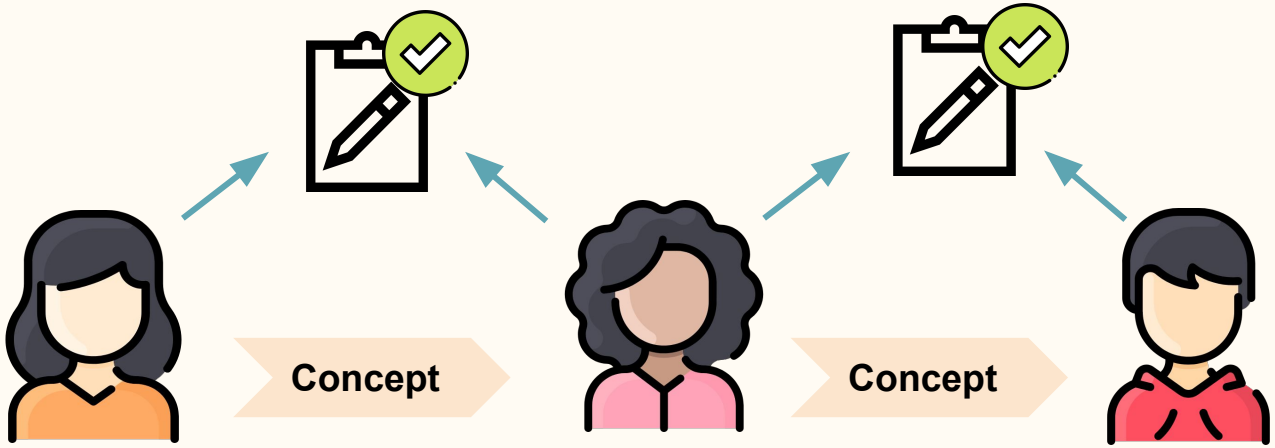
IT WOULD BE GAME-CHANGING TO make students feel more **supported** when starting out and eliminate any **roadblocks** before further learning

How might we make every student feel comfortable about basic knowledge and confident to move on to further learning?

Solution: A dynamic student-editable FAQ/class wiki

Prototype 1

Peer Teaching



Assumptions:

- **Teaching** helps one **understand** a concept
- Correct understanding of a concept **won't get lost** in translation
- Novices are uniquely situated to teach other novices

Prototype 1 Peer Teaching

Teaching is a nerve-wracking experience

*"I was kinda **nervous** because this concept is brand new."*

"I felt nervous teaching another person."

More active learning; very rewarding

*"He taught the concept the **exact same way** I taught him!"*

*"I was more **focused** when learning, knowing that I'll have to teach it."*

*"The more questions my 'student' asks, the more **fulfillment** I get."*

*"I'd love to have a **study group** like this!"*

POV 2

WE MET Carolyne, a Junior CS major at Yale

WE WERE AMAZED TO REALIZE she likes to work by herself to avoid feeling like an **impostor** even if it means spending much more time and effort on assignments

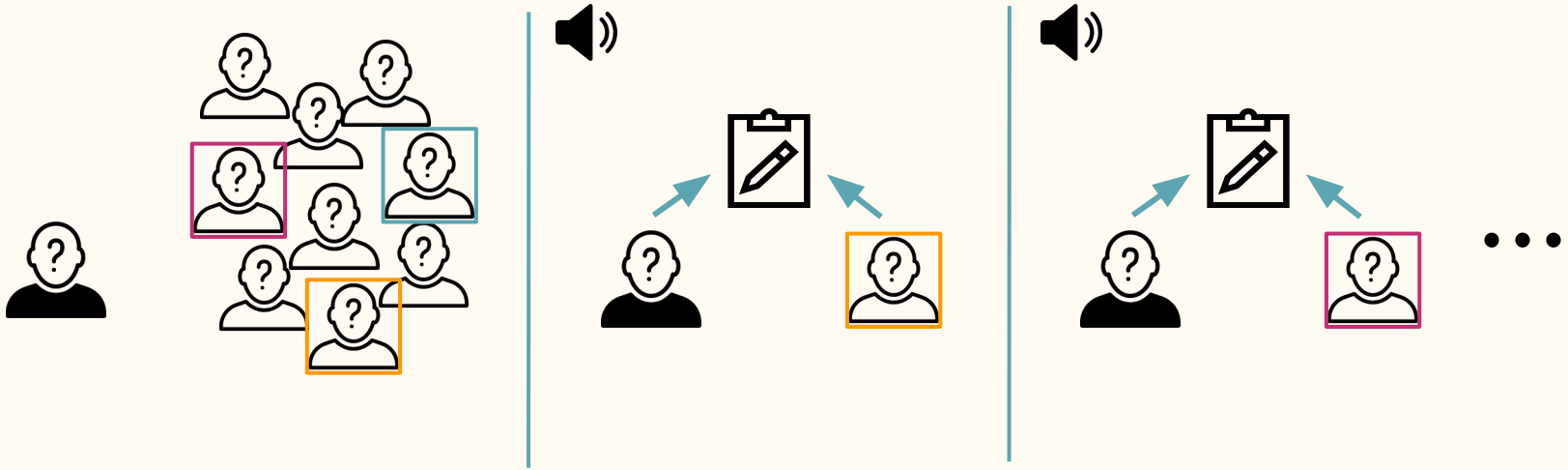
IT WOULD BE GAME-CHANGING TO create collaborative spaces that don't foster **impostor syndrome**

How might we make a space for collaboration where students feel safe?

Solution: Randomized anonymous group work

Prototype 2

Anonymous Collaboration



- Assumption: Anonymity encourages **less stressful** collaboration

Prototype 2 Anonymous Collaboration

Not a huge stress relief for people who don't experience imposter syndrome

*"I'm comfortable w**ith or without** anonymity"*

It's fair, efficient, and helpful for coding interviews

*"I think it's a great way to prepare me to code in front of strangers like in **coding interviews**."*

*"If I'm stuck with the same group all semester, I'd want good teammates. This constantly switching is **fair** for everyone, so I don't care about others' skill levels."*

POV 3

WE MET Prof. Mink, the chair of his institution'sCS department, who has developed and taught the introductory CS course there for the past 14 years

WE WERE AMAZED TO REALIZE that he does not feel pair programming/collaborative coding in the intro CS courses is beneficial to student learning in practice even though he recognizes the **benefits of collaboration** while working in industry in this field

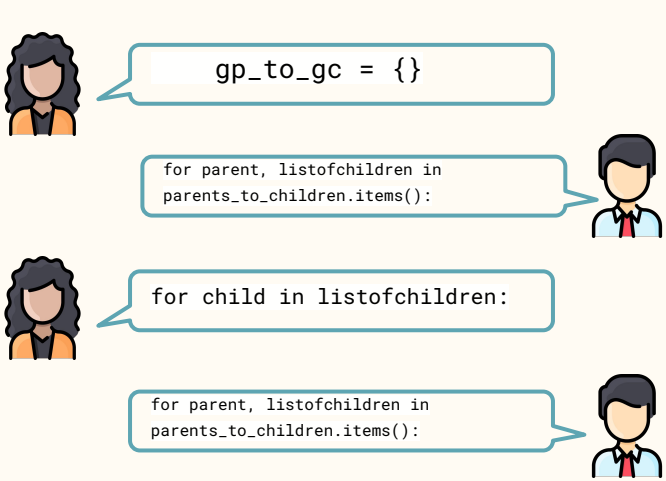
IT WOULD BE GAME-CHANGING TO create a collaborative environment that is in the best interest of student **learning** while also **preparing** students for the collaborative coding that happens in industry / group work.

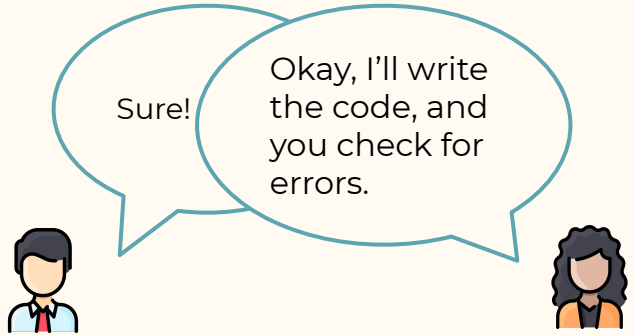
How might we encourage equal contribution from collaborators?

Solution: Alternate small parts of a larger process

Prototype 3

Alternate Line Coding vs. Natural Collaboration





Assumptions:

- Forced alternating input results in a more balanced collaboration
- More balanced collaboration leads to more beneficial learning outcomes for both participants

Prototype 3 Alternate Line Coding

Not suitable for more complex problems, doesn't encourage communication

*"It doesn't resemble how it works in the **real world**."*

*It's hard to build **personal connections**.*

A fun, productive activity that fosters a more balanced collaboration

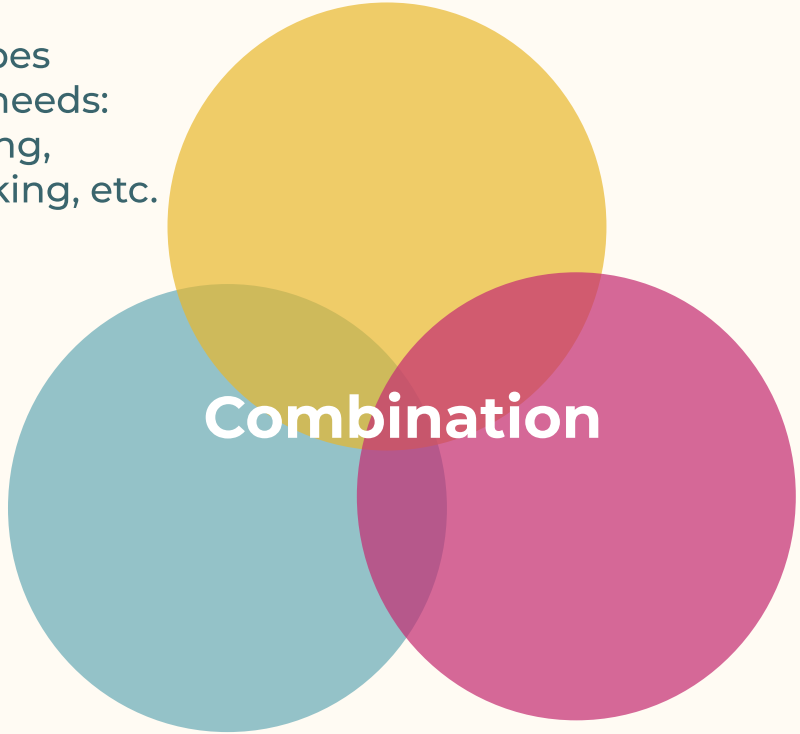
*"It was such a **fun** activity!"*

*"It helps me **explore** other solutions I hadn't thought of."*

*"It's good that **both** of us need to understand the problem."*

Our Solution

The three prototypes address different needs: community building, independent thinking, etc.



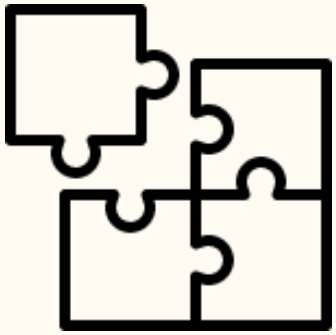
The three solutions complement each other pedagogically

All three prototype tests were generally successful and well-received

Summary

Collaboration and peer-teaching helps students solidify their understanding

Anonymity can encourage student participation



Alternating small parts can lead to mutual understanding

What questions do you have?

Appendix

Additional Needfinding Empathy Map 1

Say:
“We created pair programming as a way to reduce the workload of section leaders”
“If there aren’t tools that force the best practices of pair programming, it is easy to do it badly”
“Most students just divide and conquer on the assignments, which isn’t what we are trying to teach them”
“The 106s are courses where students need to develop code on their own to show individual comprehension of the material’
“It is easier to understand the importance of pair programming / the necessary dynamics when you are an upper level student”
“Online learning makes courses with 500 people even more intimidating”
“The percentage of women section leaders is higher than the percentage of women in the CS major”

Do:
Actively tries to mitigate stereotype threat when teaching
Provides lots of resources for students in his course
Try to lower activation energy for students to get help in the course
Assigns work that can be completed in a reasonable amount of time without a partner
Makes changes to ease the workload of section leaders

Think:
It is important that the gender ratios in the major and in sub cultures of the major are balanced
Students need to work independently on assignments to prove individual knowledge
Discussion sections are the place to get individual and personalized attention
Individual attention is necessary to succeed in 106 courses
Pair programming is necessary in industry
Students feel more isolated from course staff in online learning
Students want a space to collaborate with their peers

Feel:
The need to bring people together so that they don’t feel intimidated in such a large class
The need to continually develop and improve the section leading program to best meet the needs of students
Informed
Justified

Additional Needfinding Empathy Map 2

Say
“All of our coding assignments were individualistic with a very strict honor code.”
“I felt like the most helpful thing was a lab for reiterating concepts.”
“I got to meet a lot of friends in labs that I got to collaborate with in the future.”
“If you don’t have any experience but you work next to someone who has, they might finish a lot quicker than you, and you would feel stressed out and fall behind.”
“I realized that my experience was far less than other experienced peers.”
“I found those who are more arrogant tend to not be the most patient or willing to help.”
“People here are black and white kind of thinking.”
“None of my CS classes gave me insights about my internship. I learned all that on my own.”
“A lot of schools can do a better job explaining to students what career paths there are available for CS students.”
“Teaching each other is really helpful in industry.”
“Most ppl hate group projects, but they are the most helpful for preparing for the real world.“
“A lot of the girls I know come to CS and get scared so they switched majors.”

Do
Meet students out of class but in the discipline for support.
Prepare individually for job readiness.
Explore careers and applications for computer science individually.
Ultimately work in either software engineering industry or academia.
Gains soft skills from liberal arts minor and other interests.
Avoid arrogant people when seeking clarifications.
Compare skill levels when in a new class.
Teach each other when working in industry.
Learn about tooling.

Think:
Computer science is the hardest discipline.
Higher Education institutions do not focus on teaching soft skills to computer science/engineering students like myself.
Computer science was the best major fit for me.
Working alone is worth avoiding arrogant people.
Tooling education prepared me for industry.
Groupwork was the most helpful learning approach in preparing for work in industry.
My female friends dropped out of computer science because of intimidation.
Fear of honor code violations prevent open collaboration.
Technical qualifications are often overshadowed by an inability to communicate effectively.
General education classes like public speaking are not legitimate at school.
There are lots of opportunities to apply computer science in different ways.

Feel
Independent
Frustrated
Rewarded
Self-motivated
Determined
Empathetic
Intellectually curious
Confident in communication skills
Reassured by industry experience
Underrepresented
Uninformed
Competitive

HMW's for POV 1

- HMW provide individual help to beginners?
- HMW remove the need for tools?
- HMW make students comfortable struggling?
- HMW get students to help each other through the process?
- HMW integrate tool learning process into the regular learning process?
- HMW remove the pressure to understand tools instantly?
- HMW make the tools available and accessible for everybody?
- How might we develop tooling skills separately from the assignments?
- How might we increase efficiency of tool use earlier on in the quarter?
- How might we eliminate setup time?
- How might we better document challenges that students faced when getting their setup to work?
- How might we create a better metric for evaluating healthy struggle versus unintuitive or unuseful effort to problem solve?
- How might we create more instantaneous support for setup challenges?

HMW's for POV 2

- HMW increase students' self confidence?
- HMW level the playing field for students of varying abilities in collaborative environments?
- HMW raise awareness of other students' challenges?
- HMW create anonymous collaboration opportunities?
- HMW make students feel more comfortable sharing their vulnerabilities?
- HMW create smaller communities in the CS department?
- HMW match students with similar skill levels?
- HMW make failing okay?
- HMW eliminate arrogance in collaboration?
- HMW get students to empathize with each other?
- HMW eliminate the desire to compare yourself to your peers?
- HMW make intro CS like a morgue (everyone's equal)?

HMW's for POV 3

- How might we make pair programming support learning in intro CS classes?
- HMW stimulate industry coding in intro CS pair-programming?
- HMW create non-assignment spaces for collaboration?
- HMW create a space where students can collaborate, but also learn individually?
- HMW prevent the divide-and-conquer approach of pair programming?
- HMW design assignments that require group work?
- HMW we expedite building an individual student's foundation?
- HMW demonstrate individual learning while collaborating?
- HMW appease an instructor's goal to have all students know all the material in pair programming?
- HMW learning CS like a tennis game (takes two people to play, but both need to be fully competent)?
- HMW make students more excited about individual work?
- HMW encourage all people to contribute to all parts of the process?
- HMW find the right people to collaborate with?
- HMW we make group work beneficial to learning but not harmful to performance evaluation?