Collaboration did not 'help' and why that might be a good thing

Dr. Tony Lowe
Jarvis College of Computing and Digitial Media
DePaul University
Chicago, IL, USA
https://orcid.org/0000-0002-5510-5235

***Abstract*— This research-to-practice paper explores the role of collaboration in the classroom. Many computing educators create opportunities for collaboration as a strategy to help students in computing courses. Collaboration improves learning, motivation, engagement, and retention, yet academic dishonesty concerns often lead instructors to set policies that limit students from working with others. Course designers must balance allowing formative collaborations and restricting the students who 'over collaborate' seeking grades over learning. Understanding the value and cognitive mechanisms underlying collaboration juxtaposed with the risks to academic integrity may help educators create a comfortable equilibrium within their courses.**

The shift to virtual learning during the pandemic created a pseudo-natural experiment on the effects of collaboration. Students 'lost' many interactions we take for granted when moving from a classroom to a solitary screen. To restore some semblance of community, the author created virtual workgroups (a.k.a. pods) to replicate the circumstantial bonds of an in-person setting. The course design encouraged students to collaborate yet required them to report their collaborations, knowing that the grading included heavy scrutiny for original work. Returning to campus provided the opportunity to compare the various collaboration approaches with those formally introduced and their outcomes.

A combination of survey data, status reports, and technology provides insights into how students view collaboration, their associated behaviors, and collaboration's relationship to learning. Most students see value in and note few barriers to collaboration, but these opinions did not lead to differences in behaviors nor consistently improved outcomes. The lack of meaningful results may seem disappointing given the efforts to create a highly open and flexible collaborative learning environment. An alternative view suggests that collaboration helped but not in a way that disrupted individual learning. By adding appropriate checks and balances, courses can support open collaboration and maintain academic integrity.

Keywords—Collaborative Learning, Computing Education, Vygotsky, Academic Integrity

I. Mixed Views on Collaboration

Research and classroom practice on collaboration in computing education do not always align. Many authors write about the benefits of students collaborating. Students who collaborate are more motivated [1], less likely to quit [2], may learn more [3], [4], and are learning an essential skill that many top workplaces desire [5]. Yet educators worry about students collaborating *too much*, fearing collaboration becoming plagiarism [6]. Educators must balance measurements of individual learning versus effective learning aids when creating policies and curricula. Tradition and habit may lead educators to sacrifice how much an individual student could potentially learn because they distrust that 'too much collaboration' could overrepresent an individual's knowledge. In short, measuring *individual* learning often supersede vital pathways to improved learning (and inclusive access).

The pandemic changed many aspects of teaching, but more than any might have reshaped collaboration. Online education removed students from their traditional networks and the opportunities for 'chance pairings' – two students sitting next to one another, meeting in a computer lab, or other ways students expand their study networks. The post-mortem feedback after a capstone team assignment made this loss apparent. Students raved about the opportunity to work with others and lamented that the project was not earlier in the term. They wished they could have better leveraged their teammates throughout the course. Collaboration was so critical for some students that creating opportunities to work with others may be essential to creating inclusive computing courses.

I '*knew*' the value of collaboration from the research, yet it did not immediately enter my course design. As a corporate trainer and online instructor, insular students were the norm. Students either excelled as independent workers or asked for help when needed. My experiences with 'intentional collaboration' as a student were mixed. Being a CS minor, team projects tended to mean that my peers assigned me tasks tangential to the course's main content. I learned significantly more about C's curses library than I ever did about system programming or game theory (but received strong grades)! Completing coursework in Engineering Education challenged my slight anti-teamwork bias through empirical and theoretical accounts of collaborative learning. Collaboration seemed helpful, but the theory and research did not give practical insights into *how* most students collaborate (or fail to).

Students can benefit from many forms of collaboration, but understanding how may help educators harness it. My version of classroom collaboration, and the mixed and potentially depressing results, may further the discussion of how to promote collaboration and learning. As a spoiler, students who collaborated in my courses did not show any particular improvement in their grades. Their attitudes toward collaboration did not match their behavior, and getting them to collaborate was difficult even when they wanted to. Despite this, I hope to inspire instructors to allow more collaboration and worry less about its impact, positive or negative.

II. Why Collaboration Helps

Educational research literature has many accounts of how collaboration increases learning directly and indirectly. Rather than rehashing their findings, practitioners may find it more applicable to see the benefits of collaboration from a theoretical standpoint. Seeing collaboration through the lens of cognition may explain why these benefits exist and why not all students see the same benefits. From this viewpoint, we may see that collaboration is not merely a "nice to have" option for students but an essential skill for becoming and excelling as a computing professional.

Lev Vygotsky believed that social interactions help shape developing skills and conceptual understanding. Vygotsky [7] suggested that our interactions with others inform our understanding of certain aspects of the world (i.e., social constructionism). The language we use and the 'corrections' we receive from others help develop the internal logic that Vygotsky called inner speech.

> "Inner speech is not the interior aspect of external speech – it is a function in itself. It still remains speech, i.e., thought connected with words. But while in external speech thought is embodied in words, in inner speech words die as they bring forth thought. Inner speech is to a large extent thinking in pure meaning. It is a dynamic, shifting, unstable thing, fluttering between word and thought, the two more or less stable, more or less firmly delineated components of verbal thought." (p. 249)

Vygotsky describes inner speech as something that begins with words and meanings but transforms into a deeper conceptual understanding.

> "[A] concept is more than the sum of certain associative bonds formed by memory, more than a mere mental habit; it is a complex and genuine act of thought that cannot be taught by drilling" (p. 149)

Inner speech is more than definitions and grammar but a profound connection of facts and experiences. The knowledge of inner speech transcends words– a common phenomenon among experts. Experts often feel certainty in their choices or actions that they can't entirely explain (or easily teach) to novices. Vygotsky's work guides educators to blend direct instruction and experience to promote conceptual understanding.

Vygotsky contributed two lasting concepts to modern educators: the more knowledgeable other (MKO) and the zone of proximal development (ZPD). The MKO is most commonly a person who knows more than the learner about a subject. The 'best' MKO is a trained professional skilled in eliciting development, though any MKO is sometimes better than none. The ZPD represents some bit of work that a person can complete independently versus more challenging work that they could accomplish under the guidance of an MKO. For example, after Y2K, I was charged with retraining many of the COBOL programmers that saved us from a crisis. Most of my students had decades of programming experience and could tackle simple problems after basic instructions on Java syntax. Their prior programming knowledge plus Java facts allowed for the creation of simple programs – the upper limit of their independent ZPD. They could not yet build code for a web application, as this lived outside of their realm of experience in COBOL. Following my lead as their MKO, most could construct dynamic web pages, but not all. The ZPD indicates the boundaries of what a person can do independently and with collaboration. Web development was viable for those with enough prior programming background and Java knowledge. Even an MKO could not overcome gaps in prior knowledge for other students, and they failed to mimic my work. The ZPD delineates a learner's potential growth *and* reflects the limitations of their current abilities.

Educators can utilize the ZPD and MKO to promote and assess learning. Most traditional views of 'copying' are negative. When students merely copy work, they are not really learning, right? Vygotsky seems to disagree. He suggests successful mimicry is unlikely unless a person has some prerequisite skill. More so, mimicry is essential to expanding an individual's ZPD. Some students inevitably copy as a means toward a good grade, but such copying is likely clumsy and observable as it takes underlying skill to mimic meaningfully. Vygotsky suggests that some students desperately need a collaborator to follow as part of their growth, and cognitive science can help us understand why.

Vygotsky tells us why collaboration is helpful, but dual process theory can further explain collaboration's support. The ZPD presents a useful theoretical construct but lacks measurable concepts that might indicate progress. If only Vygotsky lived long enough to create a *ZPDometer* that would customize assignments to challenge each student, but not too much! Instead, educators must discover their learners' inner speech maturity within their subject area and tune assignments accordingly. Vygotsky gave little guidance for measuring or even the exact nature of inner speech. Combining Vygotsky's theory with the cognitive model within dual process theory proposes exciting implications for more precisely defining the ZPD and measuring growth. Dual process theory describes cognition as containing two distinct mechanisms. System 2 is the conscious and rational logic we most associate with human cognition. System 1 describes the tacit knowledge and implicit skill that mirrors the less describable aspects of Vygotsky's inner speech. Vygotsky's conceptual understanding seems to be a mixture of learning across these two systems, explicit and implicit. Including System 1 within our "equation of collaboration" helps illuminate why working with others is critical to expanding a learner's ZPD.

System 1 provides an excellent explanation of the variance within novices' ZPD. All of my former COBOL students had years, if not decades, of programming experience and the same access to the new information about Java. If knowledge is the primary factor in setting the ZPD, then any resource (e.g., book, web page) would suffice as an aid, but Vygotsky noted that development often requires an MKO. A knowledgeable person can quickly provide facts, but the ZPD is not merely about a quick recall. If access to knowledge were the main limitation, novices could take their time and eventually mimic any task, but Vygotsky hinted that mimicry is unlikely for tasks outside of their ZPD.

> "The child acquires certain habits and skills in a given area before he learns to apply them consciously and deliberately." [8, p. 101]

Vygotsky pointed out that unconscious skill (that of System 1) is a prerequisite to the skills of conscious application (System 2). Dual process theory seems to describe the mechanics that Vygotsky captured within the ZPD. A learner can accomplish new tasks using System 2, but only up to a point. New information fills short-term memory, so System 2 can no longer attend to all required details. At this point, it becomes impossible to mimic complex tasks. A learner's ZPD advances when System 1 matures enough to automate tasks and reduce the burden on System 2.

Novice programmers provide an excellent example of mimicry and the ZPD in action. The COBOL programmers (far from novices) who mimicked code for web applications worked well within their ZPD. The combination of prior coding experience (System 1) and Java instruction (initially System 2) meant they could follow demonstrations. But what about those that could not? Were they poor programmers? Perhaps, but unlikely, they had been working and paid well for many years. Did they merely not memorize enough Java syntax and semantics? Possibly, but they did not have to as they simply needed to type in the demonstrated code. A simpler explanation is their eyes could not detect the nuance of the syntax, and their hands could not quickly correct mistakes. They were unfamiliar with the Java syntax and error messages in the new language with their untrained System 1. It was not a matter of memorization but of habitation. Those who struggled wrote less Java code at that point. As their MKO, I could compensate for conceptual understanding about web applications but could not overcome deficiencies in tacit Java knowledge.

Combining dual process theory with Vygotsky's ZPD helps explain why collaboration is extremely helpful for many students or not consistently effective for others. The interaction is ineffective when a lesson is outside a learner's ZPD. Advanced learners may chafe at the slow pace of lessons when System 1 provides instant answers rendering System 2 unneeded. Lessons beyond a learner's System 1 competency fail when their eyes and hands cannot keep up. Students who seek collaborations may be compensating for conceptual understanding. Those who could benefit but avoid them may see them as futile as they often remain unhelpful. Peers can only act as an MKO when they have overlapping ZPDs. Advanced students cannot always help struggling peers when their System 1 automates skills with which their peers need support. Such pairings may feel like they are speaking different languages. To some degree, they are as advanced System 1 skill allow System 2 to forget early struggles. Peers may benefit more from similarly struggling students. As the saying goes, sometimes two heads – and their undertrained System 1's - are better than one.

Sometimes imperfect collaborations are important because of the nature of advancing a ZPD. When educators resist or even forbid student mimicry, they limit opportunities to mature a learner's System 1. A 'strong' System 1 is often the result of prior educational opportunities and a lack of other extracurricular stressors (e.g., having less time to study due to work/family care responsibilities). Banning collaboration may more significantly impact already struggling/stressed students. Promoting collaboration and the chance of mimicking may give the best opportunity for growth and narrowing achievement gaps. Allowing unchecked collaboration risks the exact opposite effect, where students receive the illusion of learning yet circumvent the practice that System 1 vitally needs.

III. Methods for Fruitful Collaborations

The rest of this theory-to-practice discussion focuses on collaboration behaviors within a new course structure based on these theoretical insights and student feedback. While there is data, it is preliminary and not carefully as controlled as a formal study would require. This discussion aims to show how theory informs practice, garner feedback, and inspire future research.

A. Overall Course Design

The general course design drastically rethinks every aspect of pedagogy, assessment, and policies, including collaboration opportunities. Understanding these changes is essential as it sets the context for how and why students may choose to collaborate or not. The course structure described here applied to courses spanning introductory programming (CS1), data structures (CS2), systems programming, object-oriented programming, and a web applications course - generally skills-based introductory courses. An underlying learning objective within each course is developing 'System 1 competency' through repetition and broad exposure to applications of the concepts. Every course included an optional team project as a 'capstone', offering collaboration opportunities and applying the concepts within an open-ended problem-solving activity.

Since grades are a strong motivator for many students, the grading system rewards successful effort as a core of the earned grade. Students earn top grades by accumulating points within various exercises focused on each course concept. Submissions earn points only when 'correct', generally without partial credit, but with opportunities for points far exceeding expectations. For example, in a data structures course, an A requires 1000 points, 850 for a B, and 700 for a C, with more than 1500 points available across various exercises. To show full competency, students must earn at least 100 points across Trees exercises, 60 points on Graphs, etc. An accumulating points system gives students the agency to focus on preferred topics and limit feared ones, but not entirely. It promotes repetitive practice over last-minute cramming. Letting students choose assignments rewards a focus on their areas of need at any level of complexity. Most importantly, it gives every equal opportunity at top grades.

Another policy decision that influences collaboration (and poor decisions around academic honesty) is due dates. My experience and the research [9] suggested that many students benefit from flexible due dates. My courses recommend a work schedule, but the final due date for most assignments is the last day of the course. Like the grading system, flexible deadlines encourage students to work tasks to completion rather than accepting triaging points for a deadline. Struggling students can linger on foundational concepts instead of moving on (often unprepared) to new topics. Open deadlines wreak havoc on structured collaborations, as students within a rigid group still tend to work at different paces.

When students can choose what they are working on, teaching to a fixed timetable makes little sense. Traditional lecture schedules slow down some students while others come unprepared for a deep discussion of new ideas. A flipped classroom model, hosting most "lecture materials" and demonstrations on YouTube, allows just-in-time knowledge. Some students experience friction without their expected weekly lecture, but a flipped model reinforces collaboration over individual work. A flipped classroom frees time for group activities and collaborative work, following the business wisdom that "this meeting could have been an email." Including collaboration in a course design does not require a flipped classroom but shows the extent an instructor might consider changes to their teaching in light of new theory.

While other course design elements are interesting in promoting the tacit knowledge and implicit skill of System 1, these are the elements that most impact fruitful collaboration. The accumulating point system and flexible deadlines give students the time to collaborate widely. Unlimited work time gives fewer reasons to copy work when motivated to learn. Ensuring that students are putting in the "correct type" of work is critical to not just allowing but encouraging collaboration. Fortunately, measuring the desired work habits aligns well with promoting System 1 development and, with technical support, fits in easily with computing exercises.

B. Ensuring Academic Integrity Within Collaboration

Instructors ensure academic honesty by checking that homework is not copied and measuring summative knowledge with exams. "Copied homework" is a complicated concept in a highly collaborative computing classroom. What defines copying versus multiple parties contributing to and submitting a coded solution? An existing multi-section course I once taught allowed students to collaborate so long as they never looked at each other code - a straightforward but impractical rule, as it is rather difficult to help another debug without seeing their work. Another policy one of my students ran afoul of allowed collaboration with people in the class, but not their programmer parent – eliminating their best sources of an MKO during virtual learning. Most systems give students rules to follow but place the onus on the instructors to catch cheating and do little to improve self-directed learning skills.

One backstop for catching poor study habits is the summative exam. A well-designed exam can be an excellent assessment of System 1. Timed exams often focus on the automation System 1 provides but also cater to students with a naturally gifted System 2. Students who shortcut their homework are less likely to complete tests in the allotted time, even if they cram the required facts. Unfortunately, the feedback only comes late in the term and does nothing to either train System 1 or correct miseducation about how we learn. Even when tests are well constructed, they indirectly reward (or punish) the study habits that create the required types of knowledge. They may be effective at catching undesirable behaviors but do little to teach the desirable ones.

An alternative approach to academic honesty is to teach students how to study and measure (and reward) work that proves they are adhering to the desired behaviors. Instead of limiting collaboration opportunities, I ask students to:

1. State who they worked with *(and they do!)*
2. Submit original work
   (or at least work with others at the same time)
3. Show your work *(with tool support to do so)*

Students must report their collaborations via a cover sheet attached to each submission. For example, complete solutions exist online for most common data structures, so students can lean into these resources rather than recreating these concepts from scratch. They must report where they found the code (e.g., the book, GitHub), and their code should be a literal copy and paste (more on this later). Most assignment submission tools include plagiarism detection with an associated similarity score. I allow submission similar to peer work (even highly) so long as they worked synchronously and stated their collaborators. These first two rules demonstrate a trust that students *want to learn* and place the onus on students to do what is right while recognizing that such collaborations are common in professional practice.

Trust should come with verification. In courses that need to promote "System 1 learning", the best measure is the work leading up to the final product, not the final code. System 1 needs to see the mistakes and fixes involved in coding. It requires the practice of typing, running, and seeing the output of code. Encouraging students to persist through the messy slog of coding and debugging supports the "correct type" of learning required to advance the ZPD much more than any successful submission. "Seeing" a student's working process and giving feedback on that process is significantly more difficult! Measuring productive work is imprecise, as some hard work is unproductive, and not all students require hard work to advance. Over time I developed two approaches to measuring effort.

One approach that companies use in evaluating candidates is the behavioral interview. Like the classroom, knowledge and skills tests do not always tell employers how their prospects behave in day-to-day and open-ended circumstances. My students must complete at least one 15-minute behavioral interview where we talk about their working process (e.g., their most challenging assignment and how they debug). Behavioral interviews give instructors a summary of how students work and learn. They are "authentic" assessments because they are a standard gateway to employment, often left unpracticed until it comes time to get a job. Yet, like tests, they are summative, do little to give real-time feedback, and are time-consuming.

Code-heavy classes can monitor student programming behaviors by including technology. I created a simple "ledger file" that tracks students as they work. My ledger captures their code and timestamp when students run their assignment code. Even if students work with existing code or others, my ledger shows when they add code progressively, not merely copying complete solutions. As Vygotsky suggested, mimicking requires at least some level of basic skill. Students have attempted to fool the ledger but are generally unable to add lines progressively (if their code does not compile, it cannot run!). It may be possible to "fool the ledger," but doing so likely takes more time and skill than simply doing the work.

While my system of ensuring academic honesty is likely imperfect, it balances learning and integrity. It emphasizes effort over results—students who want to learn earn rewards for their time. It encourages the mimicry and feedback from others that Vygotsky states are essential for conceptual development and System 1 needs. This academic honesty structure seeks to embrace rather than restrict collaboration as a learning strategy and vital skill but also ensures that students are earning their grades.

C. *Evolving Collaboration Strategies*

My course design always allowed for collaboration, but I never realized how much encouragement it takes to get students to do it. My early teaching experiences mainly were with adult learners, who generally have little hesitancy to ask questions if needed. During my first course with traditional-aged undergrads, I received fewer weekly questions from my 117 in-person students than from my six online adult students, including during live class times! A follow-up survey told me that most students preferred asking peers from other sections despite having assigned teams within the course. They were happy to collaborate on team projects but less for individual work. This feedback and my unsatisfying teaming experiences led me to create teams only as an optional summary assignment but not as part of the core learning, yet that feedback was precisely the opposite.

Students raved about their time working with others on projects, and many lamented the lack of earlier teaming opportunities earlier in the course. Following their feedback, I wanted to provide collaboration without losing individual accountability on each assignment, thus the 'Pod' system. Pods are an assigned group of peers encouraged to use each other (a.k.a. pod mates) as collaborators. Pods ideally replicated the ad hoc connections created by physical proximity and happenstance relationships developed across campus but unavailable during virtual learning. Assigning pods ensured that no student was left out, and each student had a group that was not overly homogenous. Pod formation strategies can seemingly benefit from teaming strategies.

Forming strong teams is an imprecise art, but research offers some suggestions. The makers of a widely used teaming tool, CATME, suggested several options for team formation [10], including compatible scheduling and reducing how often teams isolate underrepresented individuals. Seeding teams with an identified MKO improved teammate outcomes in a highly project-driven course [11] and would seemingly offer support on individual assignments. The first Pods distributed students ranked by self-reported prior experience and confidence in programming, drafted students in order, and limited the pod size to four students. Each Pod likely included a student from each quartile and, ideally, a range of overlapping ZPDs. A final sweep balanced pods to avoid isolating (when known) women and international students [12].

The 'best' mix of teams varied drastically across the terms and the return to traditional classes. Creating pods based on self-assessed skills demonstrated little success. Few of these pods reported fruitful collaborations possible due to incompatible ZPDs, working on different assignments (the flexible schedule), or unknown social pressures. Anticipating this, I planned to rotate pod assignments periodically. Adjusting the pods for preferred schedules (e.g., working in/out of class times, synchronously/asynchronously) likewise did little to encourage collaboration. The most successful pods formed after a few weeks of work based on the number of points earned. Pods that include 'fast-paced' students saw some benefits on the really tricky problems but did not report much beyond this. The strongest advocates of pods were the routinely hard-working but 'middling' students. These students worked for hours and bounced ideas off of each other, most having done little collaboration with the initial pods. They may not gain the full benefits of a trained MKO, but they augmented each other's Systems 1 and 2 and reported motivational benefits. Groups of lower-scoring students did not seem to benefit much from Pods. That grouping would require further investigation to pull apart the gaps in learning, the role of a struggling peer as MKO, and other factors (e.g., lack of engagement, external commitments).

The return to campus has removed most enthusiasm for the Pod system. The Pods became optional to include only interested students, thus possibly improving participation. Without an explicit assignment for Pods, it seems students struggle to ask peers questions. Pods continue as an experiment in promoting collaboration and mainly suggest that it is not an easy task. Even giving bonus points for working collaboratively (including with me) shows no real improvement. Students who benefit from collaborating do so, but those numbers are small in my experiences.

D. *Data on Collaboration*

A secondary benefit of my alternative curriculum is enhanced sources of potentially valuable research data it gathers. The cover sheet has students indicate if they collaborated on an assignment. Since all assignments are due at the end of the course, students complete weekly status reports on their progress. The status reports ask students to report their time on various tasks, including time spent collaborating with peers. Given the spotty participation among the Pods, I added a survey on collaboration that gave a token amount of completion points. Combined with the final points earned, each source provides corroborating or conflicting data sources on students' attitudes towards collaboration, actual behaviors, and course outcomes.

IV. Results and Discussion

A. *Attitudes Towards Collaboration*

In a year, I went from appreciating the need for collaboration (but skeptical of team-driven coursework) to being a full-on facilitator of collaborations. Some of my hardest-working students praised the opportunities to collaborate, so I wanted to give them the best experience possible and possibly transform other struggling students. Yet getting students to collaborate was difficult. For every student who raved about their Pod, another would mention how theirs did little collaborating. I turned my one-on-one feedback into a survey to see how my students viewed collaborations.

The survey looked at several factors of why students collaborate (or not), what they view as advantages, barriers, and plans for collaborating in the future. The survey results cover students in CS1 and CS2 courses. The following show aggregated responses since there were no discernable differences between courses or terms. The first question asked why students collaborate with the responses shown in Table I.

TABLE I. HOW DO YOU FEEL ABOUT COLLABORATING WITH OTHERS ON SCHOOL ASSIGNMENTS?

Response	**%**
I work with others when I need help	37
I only work with others when I must	14
It is a critical part of my study habits	11
I prefer to work alone	6

Few students held the extreme view of either avoiding collaborating or viewing it as a critical success strategy. Most seem to collaborate as needed. The next question (Table II) reinforced this attitude as less than a quarter actively sought to collaborate more and very few less.

TABLE II. WHEN IT COMES TO YOUR COURSE WORK, WOULD YOU PREFER TO

Response	**%**
Collaborate about the same	40
Collaborate more	23
Collaborate less	4

The primary collaboration benefits seem equally pragmatic. The majority reportedly learn more when collaborating, and nearly half (yet less than half!) see collaboration as a critical job skill. The grouping who saw these specific advantages (↑ in Table III) scored statistically significant more points than their peers, but this was not true of all classes.

TABLE III. WHAT ARE THE ADVANTAGES YOU SEE IN COLLABORATION?

Response	%
Other people help me to understand the ideas better↑	61
Gets the work done faster	44
I don’t want to fall behind others	28
I like helping others to learn↑	40
It is an important skill needed in my profession↑	47
More likely to get the right answer	38

Most students did not believe that collaboration led to faster or more accurate answers. This attitude may imply they see learning as a priority over merely earning points in less time. Yet the percentages of people who see the benefits exceeded the number who participated voluntarily in a Pod. Nearly 69% of students saw at least one advantage, but this did not lead to an ‘easy’ option to collaborate more. Something seems to stand in the way of students engaging more in an activity they see as beneficial.

A similar number of students reported at least one barrier to collaboration shown in Table IV. Despite admitting to at least one barrier, most participated in either a Pod or the final project for the class, particularly in CS2, where nearly every student engaged with the project. It might be worth exploring the intersection of collaboration beliefs and where the educational typically accepts or ‘forbids’ collaboration.

TABLE IV. WHAT HOLDS YOU BACK FROM COLLABORATING MORE WITH OTHERS?

Response	**%**
I am afraid others will judge me	30
I don’t have anything to contribute to helping others↓	14
I don’t know anyone to communicate with	28
I feel shy↓	32
I don’t know how to collaborate very well	8
I don’t like it	10
I struggle to communicate with others	22
It is tough to find a time that works	29

For the most part, seeing any specific barrier to collaboration did not indicate any difference in final grades. In a few of the sections across the courses (but not most), students who “felt shy” or “did not have much to contribute” earned statistically significantly lower points than their peers (↓ in Table IV).

B. Collaboration Attitudes and Behaviors

Since people’s attitudes and actions do not always align, it is helpful to triangulate student behaviors and beliefs. Using the weekly status reports, 50-95% of students reported time collaborating across fifteen terms, ranging from three to over twenty hours. Students collaborated the most in CS2, partially because of the course design (an extended team project) but possibly because of additional experience in CS.

Attitudes on collaboration seemed to have little impact on how often students collaborated. A few terms saw statistically significant differences for individual questions on what they felt and how they reported time spent collaborating (e.g., “students who felt they had little to contribute” or” no one to work with” collaborated less). The results also demonstrated the opposite expectations (e.g., “students who don’t like it” or “struggle to communicate” collaborated more). Since these trends did not appear across all data sets, it seems that attitudes and behaviors about collaboration did not correlate overall.

Students generally reported more collaboration in 20-21 than in 21-22 for the percentage who reported collaboration and the hours spent. The reasons for this drop are not immediately apparent. On the one hand, returning to the classroom should give more opportunities for collaboration. On the other hand, the Pod system may have compensated. Tighter schedules might explain if the difference was merely the time spent. For example, long commutes and extracurricular activities might result in fewer available hours as students squeeze in time between commitments. Another possible explanation could be students’ general fatigue/malaise due to the pandemic [13].

C. Collaboration Behaviors and Grades

Vygotsky suggests that collaborations help people learn, so in theory, people who collaborate more should demonstrate that learning through better grades. One caveat that any study should consider is how well aligned is the grading system to learning. Grades are often buffered by 'easy' points or tempered by penalties unrelated to understanding (e.g., late penalties). One of the goals of the accumulating point system is aligning grading with a different definition of learning. A point reflects the effort spent pursuing a correct answer, not merely a correct one. Students can earn 'easy' points (e.g., status reports), but the majority demand they put in time and effort, as demonstrated by their ledger. "Correct" assignments that fail to show reasonable work in the ledger receive no points. This point system measures learning differently and may be highly sensitive to collaboration (of the right or wrong kind).

Collaboration did not lead to more points but may have led to better grades. Only one (of five) CS1 class showed a significant relationship between points and time spent collaborating. A few CS2 classes showed a significant relationship, but the group project was also a larger portion of the grade, and more sections did not show any relationship. Figure 1 charts the spread of reported collaboration hours for students earning each grade. Some students who earned higher grades did appear to collaborate more, but not significantly. Some students achieve top grades with little collaboration, and others collaborate more with less success.

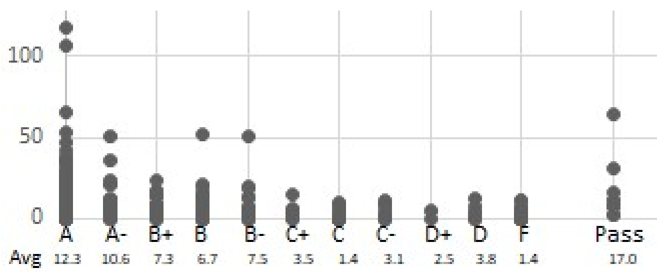

Fig. 1. Collaboration Hours by Grade Earned

V. Making sense of Student Collaborations

A. Distrust the Data

Low-quality data is a plausible explanation for the inconclusive relationship between collaboration and learning. This report describes putting theory into practice, and the data is a byproduct of my "normal course operations", not a systematic study. The survey was not a validated instrument, but a means to fine-tune the Pod system and encourage collaboration. The students may not have understood the questions, the bonus-point-seeking responders may add bias, or the questions may be poorly worded, to name a few concerns. An even more notable concern is the quality of the self-reported collaboration time. The status reports were weekly but optional and with no checks on accuracy. Students who reliably completed status reports also tended to earn better grades, but not universally. It is probably that 'better' students are overrepresented, yet 'should' make it *more* likely to show the desired outcome (collaboration helps grades/learning). The imperfect data certainly would complicate the findings of an empirical research report (which this is not). It does not eliminate our ability to find some meaning in students' reported attitudes and behaviors.

B. Collaboration is an Individual Choice

One mistake educational researchers often make is forgetting the individual learner within the statistics. Pedagogical and assessment advances that move the averages in the desired direction are wonderful, but what about those who did not see the benefits? Collaboration may work well for some but may not help others. Some students struggle to collaborate for non-academic reasons (e.g., schedules, language, or emotional barriers). Vygotsky also explained that learners need collaboration within their ZPD if the goal is development. A 'better' measure of collaboration effectiveness may be the wide distribution seen in Figure 1. Seeing students with high and low levels of collaboration at every grade level means it is not a perfect tool. Merely collaborating does not ensure success, even in a system that strongly promotes it.

Educators can include collaboration so long as individual assessment plans for it. Any assignment that merely asks for correct answers risks not only cheating but sending the wrong message about what it takes to become a professional. Rather than simplifying the materials to the point of acceptable pass rates, the alternative is to provide the time and resources for students to gain the skills required to succeed. My courses tend to reward roughly forty percent of students with an A and another forty pass with a lower grade. Open collaboration, unlimited resubmissions, and long-term due dates do not mean everyone passes. Yet few students fail because they put in the time but do not understand the concepts. Most failures have long stretches where they do not work at all. Vygotsky gives me some confidence that those who pass develop some foundation of competence from which to build.

C. Collaboration and Academic Honesty

The wide array of data gave insight into how individuals collaborate (or not). I caught academic honesty violations each term through plagiarism detection, the ledger, and the assessments. Most 'cheaters' did so late in the term, were rather apparent in their cheating, and admitted the discretion quickly. The ledger and cover sheet mainly serve to teach how to appropriate collaboration. The ledger also functions as a monitor that scales well yet still places trust and accountability on the individual. Organic interactions as students asked questions backed by behavioral interviews provided additional assurances of honest student work. With practice, behavior interviews can tell quite a bit about an individual. It was easy to get a sense of who was applying themselves or not – and I told them so. Interviewing is a notably imprecise skill, but less so with the data the submissions and ledgers provide. It felt apparent when people wrote code versus following another.

D. Collaboration Failed, but Perhaps Not

It was initially disappointing that the data did not confirm the relationship between collaboration and learning, providing an educational silver bullet. Finding any universal aid to education may be impossible given what we are learning about the brain. Dual process theory reminds us that cognition is highly individual, based on a person's experiences as much as their knowledge. When educational goals move beyond retaining facts, experiences may be more important to professional development. Computing professionals must be creative designers and problem-solvers, which seems to transcend simple book lessons. Collaboration may be an essential aid towards such accelerating learning. My students both seemed to appreciate and be reluctant to collaborate. Future researchers may endeavor to unpack these feelings, but they need not limit educators in the meantime if they plan accordingly. By creating proper measures, classrooms can likely remove collaboration limits and allow individuals who need the support to do so in honest and fruitful ways.

VI. Taking this Research to Your Practice

Educators do not need to adopt any or all of the specific elements of the presented course design to use the underlying principles of implicit learning and collaboration. Adopting such a radically different approach can be personally taxing. I dreaded that most of my students would fail my first term using this approach. After completing, so many received top grades that I worried my expectations were too low! I endeavor to find ways to engage procrastinators, probably because of an irrational fear that they may do so because of flexible deadlines. My advice is to adopt and realize principles in a way that supports your values (and allows you to sleep at night).

A first step is challenging your assumptions on cognition, behavior, and their connection. Dual process theory helps to reframe otherwise "irrational" decision-making by students. For example, students can know that collaboration is helpful but will not do so if it is not within their existing habits. I sometimes think they must be trained "not to bother their teachers" with questions as so few are willing to do so, even when they say they will in status reports. The status reports are an excellent mechanism to see these paradoxes and decide where to spend time and effort.

The next step is to try something (anything!) but measure the results. Collaboration was not a priority within my initial course design. It is a response to student feedback that became even more interesting because of its intractability. After many attempts, the only strategy that consistently seems to work is giving Pods direct tasks, and then only for the duration of that task. Left alone, most students seem unlikely to collaborate. Collaboration seems an implicitly learned skill that I forgot when and how I learned yet has been vital in my career in industry and academia. I hope to give experiences that will instill the proper habits in my students.

Our long-term goals should make computing education as permissively collaborative as the computing workplace. The 'best' modern workplace assumes trust and accountability and expects its employees to make good choices [5]. It seems unsurprising that students fail to acquire these habits when coming from educational systems that are fundamentally closed to collaboration. Modeling trust, accountability, integrity, uncertainty, and collaboration as a regular part of classes seems an excellent way to instill these mindsets before graduating. Doing so may require drastic shifts in how we design courses, policies, and how we view learning. Rethinking traditional approaches may be the only way to be inclusive and expand computing education to all.

Acknowledgments

I must thank my students for the honest feedback and conversations we have had over the past years. It can't be easy to tell an instructor you don't like part of their course design, but many brave souls have done so. While I probably will never agree that you need deadlines to complete your work in a course, I do hear your belief. I also want to thank Jacob Furst for supporting innovation in the classroom even when it stretches the norm.

References

[1] L. M. Serrano-Cámara, M. Paredes-Velasco, C. M. Alcover, and J. Á. Velazquez-Iturbide, "An evaluation of students' motivation in computer-supported collaborative learning of programming concepts," *Comput. Human Behav.*, vol. 31, no. 1, pp. 499–508, 2014.

[2] L. Porter and B. Simon, "Retaining nearly one-third more majors with a trio of instructional best practices in CS1," *SIGCSE 2013 - Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 165–170, 2013.

[3] F. J. Rodríguez, K. M. Price, and K. E. Boyer, "Exploring the pair programming process: Characteristics of effective collaboration," in *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 2017, pp. 507–512.

[4] D. Teague and P. Roe, "Collaborative Learning – towards a solution for novice programmers," in *Proceedings of the Tenth Australasian Computing Education Conference in Conferences in Research and Practice in Information Technology - CRPIT*, 2008, pp. 653-669.

[5] E. Schmidt and J. Rosenberg, *How google works*. Grand Central Publishing, 2014.

[6] T. Mason, A. Gavrilovska, and D. A. Joyner, "Collaboration Versus Cheating," pp. 1004–1010, 2019.

[7] L. Vygotsky, *Thought and language*. Cambridge, MA: MIT Press, 1986.

[8] L. Vygotsky, *Thought and language*, Third Pape. Cambridge, MA: M.I.T. Press, 1962.

[9] M. A. Patton, "The Importance of Being Flexible with Assignment Deadlines," *High. Educ. Eur.*, vol. 25, no. 3, pp. 417–423, 2000.

[10] G. D. Ricco, M. W. Ohland, M. L. Loughry, and R. A. Layton, "Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria," *Adv. Eng. Educ.*, vol. 2, no. 1, pp. 1–28, 2010.

[11] T. A. Lowe and S. B. Brophy, "Understanding the impact of strategic team formation in early programming education," in *ASEE Annual Conference Proceedings*, 2017.

[12] M. W. Ohland, "CATME and Team-Maker Tools for Teamwork," 2009.

[13] B. McMurtrie, "A 'Stunning' Level of Student Disconnection," *The Chronicle of Higher Education*, 2022. [Online]. Available: https://www.chronicle.com/article/a-stunning-level-of-student-disconnection?cid=gen_sign_in&cid2=gen_login_refresh. [Accessed: 24-Apr-2022].