Lab Partners: If They're Good Enough for the Natural Sciences, Why Aren't They Good Enough for Us?

Laurie Williams and Lucas Layman
North Carolina State University
{lawilli3, lmlayma2}@ncsu.edu

Abstract

Despite many professed benefits of collaboration, some computer science educators feel students need to master work individually, particularly in the courses early in the curriculum that feed into software engineering courses. In the natural sciences, however, students almost always work with one or more partners in the laboratory. What can computer science educators learn about collaborative lab settings from our natural science counterparts? We conducted a survey of science and computer science educators to compare views and use of collaboration in their classes. The positive and negative aspects of collaboration, as reported by the natural science educators, are strikingly similar to those of computer science educators. These results suggest that computer science educators should be more open to the use of collaborative labs, as is done in the natural sciences, for the overall benefit to students.

1. Introduction

"In the first course, students need some time to absorb the ideas themselves."

"My inclination is to allow more group work starting in the second course."

"We want to be sure that each student writes enough code him/herself to learn the introductory concepts."

"I am against pair-programming in introductory courses, where students need to develop strong programming skills themselves."

– four anonymous survey respondents

As the quotes above illustrate, some computer science educators believe that students should work alone in introductory computer science classes and that any collaboration should occur later in the curriculum [10]. However, by making students work alone, the incorrect myth that software engineers work alone all the time is perpetuated [10]. Millennial students, those belonging to the generation born after 1982, are denied their desire of working with others is denied [5-7]. Furthermore, the opportunity for gaining the general benefits associated with peer-lead, active learning [8] is lost.

In the natural sciences, such as chemistry, physics and biology, students almost always work with one or more partners in the laboratories beginning with introductory classes. Why would collaboration be beneficial for science students but be detrimental to computer science students? We conducted a survey of science and computer science educators to compare the views of the educators to see what computer science can learn from our natural science counterparts who have decades of experience with collaborative lab settings.

The rest of this paper is organized as follows. Section 2 provides information on background and related work on collaboration and collaborative pedagogies. Section 3 describes the research method while Section 4 describes our research results. We summarize in Section 5.

2. Related Work

In this section, we summarize research results related to the use of collaborative pedagogy and about the benefits of collaboration when approaching creative tasks.

2.1. Collaborative pedagogy

The term “collaborative pedagogy” takes on many definitions in educational literature. Drawing on Dillenbourg’s [2] broad definition of “collaborative learning,” we refer to collaborative pedagogy as a pedagogy that emphasizes “situations in which two or more people learn or attempt to learn something together.” Group exercises in the classroom, community lab projects, and pair programming are all considered part of a collaborative pedagogy. Collaborative learning has been central to science curriculums in biology, chemistry and physics for many years and has been increasingly adopted in mathematics, technology, and engineering as well. Collaborative labs help students to learn through experience, leverage the perceptions of their classmates, and form their own opinions on the utility of science through social constructivism [4]. A meta-analysis of 383 published studies on small-group learning (2-10 students) by Springer, et al. confirms the general sentiment that collaboration has consistently had positive effects on student achievement, self esteem, and attitude toward learning [8].

2.2. Collaboration on creative tasks

Laboratory exercises in the natural sciences often follow prescribed steps leading to a predictable outcome. Conversely, laboratory exercises in computer science often involve students using creativity to determine solutions to problems. Is collaboration beneficial in a creative setting? Creativity emerges in student's interactions with the environment, with each other, and with the knowledge artifacts created with others [3]. In a healthy group, students will discuss different perspectives on a problem and will need to elaborate on their individual ideas and thought processes to others, which can help the students learn to think from different perspectives [1].

3. Research Methodology

The research began with informal, independent conversations between the first author and two natural science professors (one in physics, one in chemistry) at North Carolina State University. The first author asked for permission to videotape the other professors’ laboratories, which prompted the professors to ask about the purpose of the taping. The first author explained that computer science was just beginning to allow collaboration in its laboratories, and that we wanted to see what we could learn from the natural sciences. Without prompting, each science professor noted that there were problems associated with lab partners. Specifically, they noted the following problems:

- very good students know they can do the job well by themselves and so they do not understand or realize the benefits of collaborative group work;
- poor students have a better chance to "cash in" on someone else's work when working in pairs; and
- students do not like to work with partners that know significantly less than them.

The first author observed the similarity between the concerns of the natural science professors and those of computer science professors who discourage collaboration.

Subsequently, the authors videotaped (1) a solo programming lab; (2) a paired programming lab; (3) a physics lab; and (4) a chemistry lab. The purpose of the videotaping was for an instructional video on collaboration for computer science educators, not for qualitative analysis of student interaction. However, in the course of the videotaping and the subsequent viewing, stark differences between the solo programming and the other three collaborative lab settings were observed. Students in solo labs were generally pensive, serious, frustrated and dependant upon the teaching assistant. Students in a collaborative lab setting were generally productive, happy, and independent. Quite by accident, one student was videotaped in both a solo programming lab and in a collaborative physics lab. Note the difference in the facial features and apparent demeanor of the student in Figure 1.

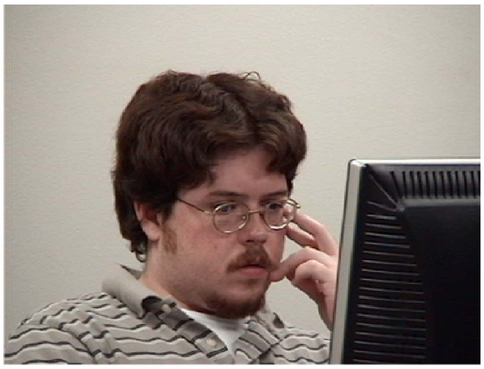

Figure 1: Same student in solo programming lab (left) and collaborative physics lab (right)

The discussions with the science professors and the observations of the students during the videotaping prompted the authors to conduct two anonymous online surveys – one for natural science educators and one for computer science educators. Both surveys were conducted using the SurveyMonkey survey tool (www.surveymonkey.com). The purpose of the surveys was to compare the views of the educators to see what computer science can learn from our natural science counterparts who have decades of experience with collaborative lab settings. Some demographics were collected from the respondents to generally determine the external validity of the sample among educators in each discipline.

The survey for natural science educators was 10 questions long. To obtain possible respondents for the survey, the second author visited the biology, chemistry, and physics department web pages of the first, middle, and last five institutions on U.S. News and World Report's annual rankings of top overall and top public universities[1]. We then gathered the email addresses for one or two educators who taught introductory courses in each of the programs, resulting in a total of 176 educators from 26 institutions who were invited by email by the first author to take the survey. After two weeks, a reminder email was sent to the educators. The survey was answered by 48 science educators. Only 25 of the respondents provided information we could use to determine their discipline: 10 from chemistry, 8 from physics, and 7 from biology.

The survey for computer science educators was 20 questions long. The computer science survey contained ten questions nearly identical to the science survey, and ten additional questions. The survey was advertised on the SIGCSE (www.sigcse.org) mailing list and noted that anyone who answered the survey would be provided the results of the survey. The survey was at least partially answered by 148 computer science educators, 84% of these were professors (essentially evenly split between assistant, associate, and full professors), 14%

[1] *America's best colleges*, 2006, U.S. News and World Report, Washington, D.C.

were lecturers or graduate student instructors, and 2% were high school teachers. Of these, 65% taught in four-year colleges or universities, 31% in research universities, 2% in community colleges, and 2% in high school. Finally, the educators were currently teaching the following courses: CS1 (50%), CS2 (33%), data structures (30%), software engineering (31%), and graphics (5%). After answering the demographics question, 47 of the educators stopped answering the questions so the reported results involve responses from 101 educators.

4. Results

In this section, we summarize the results of the survey. Some of the computer science questions referred to collaboration in the general sense. Others referred specifically to pair programming, a structured approach to collaboration whereby two programmers work together at one computer, collaborating on the same algorithm, code, or test [11]. In our discussion, we will make it clear which of the two the question related to.

4.1. Rationale for Allowing Collaboration

In this subsection, we discuss the results of one question on the science survey and two questions on the computer science survey related to the educator's rationale for allowing collaboration. Presented first are the results of a multiple choice question on the rationale for using lab partners in science or pair programming in computer science. This question was answered by all 48 science educators and 64 computer science respondents who allow pair programming in their classes. The results are shown in Table 1. As shown, the lack of equipment and space is more of an issue in the sciences than in computer science. Almost all educators agreed that students learned better in collaborative settings. Computer science educators were more likely to say that students needed less help from the lab instructor. The difference may be that science educators are less likely to have a comparison point because they had never conducted a lab in which everyone worked alone so they do not know how much help solo students would need. Conversely, many computer science educators have experienced feeling overwhelmed with student questions.

Table 1: Reasons for Allowing Collaboration

Rationale		**Percentage**
Lack of enough equipment.	Sciences	53.1%
	CS	9.4%
Lack of enough space for each student to work alone.	Sciences	30.6%
	CS	6.2%
A student is more likely to adhere to safety procedures if he/she has as partner.	Sciences	14.3%
	CS	N/A
A student is more likely to follow lab procedures if he/she has a partner.	Sciences	18.4%
	CS	39.1%
Students generally answer each others' questions/help each other so are less likely to need as much help from the lab instructor.	Sciences	44.9%
	CS	71.9%
Students learn better when discussing lab activities and findings with others.	Sciences	89.8%
	CS	92.2%
Helps with retention of students in computer science	Sciences	N/A
		39.1%

Additionally, the computer science educators were asked an open-ended question about their rationale behind allowing general collaboration in their classes. These results are compared with the additional commentary provided by the science educators. In many ways, the qualitative responses supported the reasoning presented in Table 1, and additional themes emerged. Some representative verbatim comments appear below with an indication of whether these themes were mentioned by computer science educators, science educators, or both.

- *"I am teaching them to function in the world. The world functions collaboratively."* [both]
- *"Students are happier. It is more fun! My students are smiling and sometimes laughing… Increases morale by providing a powerful anecdote to frustration and isolation."* [both]
- *"It builds teamwork and communication skills, which employers unanimously say they want."* [computer science]
- *"Sometimes, labs actually need two pairs of hands, two sets of eyes (e.g. someone needs to use a timer while another person is putting a system in motion)."* [science]

4.2. Advantages and Disadvantages of Collaboration for Students

We asked all educators their perception of the pros and cons of collaboration from a student perspective. This question was answered by the 63 computer science respondents who allow pair programming in their classes and all 48 science educators. The results are shown in Table 2. The most prominent disadvantage was that one student would sometimes do all of the work. This issue and some suggestions for addressing this problem will be discussed in Section 4.3.

Table 2: Perceived Student Disadvantages of Collaboration

Disadvantage		Percentage
Have to deal with difficult people	Sciences	39.5%
	CS	63.5%
Sometimes a partner will try to have the student do all the work	Sciences	69.8%
	CS	73.0%
Have to explain things to their partner	Sciences	23.3%
	CS	52.4%

The educators were able to provide qualitative explanations to their answers and any additional disadvantages. Some representative verbatim comments appear below with an indication of whether these themes were mentioned by computer science educators, science educators, or both.

- *"…having to deal with difficult people for example, having to explain things to a partner are not disadvantages…Teaching something to someone else is a very good way to learn."* [both]
- *"I try to have small groups of comparable abilities and backgrounds work together in lab."* [both]
- *"Scheduling difficulties causes stress."* [computer science]
- *"Very strong students often have problems pairing. They often prefer to program alone… I make them pair because learning to get along with others is part of the course."* [computer science]
- *"Members of a team receive part of the grade based on the team's joint report and part on their particular role. A student sometimes resents having his/her own*

grade depressed when a major role is assigned to a team member who isn't trying as hard." [science]

Our research [12] and that of psychologist Vygotsky [9] support the educators' observations that students work most effectively and compatibly with partners of similar abilities.

The responses to the educator's perception of the benefits of collaboration to students are presented in Table 3. Essentially, the majority of educators supported that these four advantages were realized by students.

Table 3: Perceived Student Advantages of Collaboration

Advantage		Percentage
Learn more effectively	Sciences	87.5%
	CS	74.6%
Get to know others in class	Sciences	68.8%
	CS	76.2%
Develop collaboration skills	Sciences	79.2%
	CS	77.8%
Learn how to deal with different types of people	Sciences	70.8%
	CS	54.0%

Again, the educators were able to provide qualitative explanations to their answers and any additional advantages:

- *"After I explain a concept more than once, sometimes a fellow student can help break down a barrier to learning that I have not identified. Learning is more important than who accomplishes it."* [both]
- *"They are more comfortable and eventually more confident."* [computer science]
- *"Students often confide in me that they feel like everyone else understands the material. When they work in pairs, they see that others are struggling as well, which increases morale."* [computer science]

4.3. Advantages and Disadvantages of Collaboration for Teaching Staff

Computer science educators can be resistant to instituting pair programming. Three common educator resistances to pair programming are presented in Table 4. The educators were asked to rate, on a five-point Likert scale ranging from "no problem" to "major headache," how severe each problem has been in their classes. This question was answered by 60 computer science respondents who allow pair programming in their classes and all 48 science educators.

The single biggest problem identified by the educators is one person of the pair doing all the work. In our experiences [11], classroom management tactics can be used to reduce these instances of "freeloading."

- Collect peer evaluations after each lab and/or assignment to gather information on the contributions of each student. We have created the open source PairEval[2] tool which can be used to manage the collection of peer evaluations for instructor review.
- Promptly speak with students whose contribution has en evaluated poorly on the peer evaluation.

[2] http://agile.csc.ncsu.edu/paireval/

Table 4: Educator Problems

Concern		No problem				Major headache
Students assigned to work together simply cannot get along. You need to split the partners up and reassign to new partners.	Sciences	49%	41%	6%	2%	2%
	CS	35%	45%	15%	3%	2%
One person of the group is doing all the work	Sciences	10%	25%	42%	21%	2%
	CS	7%	23%	25%	33%	12%
Cheating on assignments or collaborating between groups when they aren't supposed to.	Sciences	36%	34%	15%	6%	9%
	CS	58%	25%	12%	5%	0%

- Change partners often to gain multiple perspectives/opinions on the contribution of each student. Most computer science and science educators in our survey indicated that they switched partners often throughout the semester. Switching systematically frees students of undesirable partners and avoids awkward confrontations, as supported by this representative educator comment:

  > *…sometimes people choose bad partners and get stuck in a dysfunctional relationship that they refuse to break … Students are too meek to exercise their right to change partners or drop partners if they are unhappy.*

- Have severe grade implications for non-participation. In our classes, the student's grade can be multiplied by their professed contribution (e.g. the student did 50% of what they should have, the earned grade is multiplied by 0.5).

The results indicate cheating is perceived as somewhat of a problem. However, many computer science educators provided qualitative feedback that pair programming is a structured way of managing and "legalizing" the collaboration that will inevitably happen between students. One computer science educator commented to this point:

> *[Pair programming will] cut down drastically on plagiarism problems… many students will work with others whether it is permitted or not, therefore it is better to formalize the process, using methodologies shown to improve learning.*

Two additional benefits of collaboration expressed by the educators are (1) a reduction in grading; and (2) a reduction in the number of student questions.

4.4. Balance with Individual Work

All educators also emphasized the need to balance collaborative work with individual work in each class. In the sciences, slightly more than half of the educators ask students to prepare a pre-lab and/or to complete the lab report individually. Many computer science educators noted that they used collaboration in closed laboratories but asked students to also complete additional programs and/or written homework assignments individually outside of lab. By having some individual work, the instructor can assess individual performance and the student is motivated to be able to perform on his or her own.

5. Summary

Most likely, every reader of this paper has taken at least one natural science class with an lab and has experienced working with a lab partner. Why have lab partners generally been used for decades in the natural sciences but are not generally accepted for computer science? Often computer science educators have concerns about and perhaps imperfect experiences with collaboration that drive them away from allowing students to work together. The results of our survey suggest that the concerns and imperfect experiences of educators in the natural sciences are very similar to those of computer science educators. However, educators in the natural sciences have determined the benefits of collaboration outweigh these shortcomings, and their students enjoy the benefits of collaboration. Based upon these findings, we recommend that computer science educators integrate collaboration and the notion of "lab partners" in their classes as much as possible to the benefit of the students.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants ITWF 00305917 and BPC 0540523. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Many thanks to the students in the labs who allowed us to videotape and photograph them and to the almost 200 science and computer science instructors for answering the surveys.

7. References

[1] P. C. Blumenfeld, R. W. Marx, E. Soloway, and J. Krajcik, "Learning with Peers: From Small Group Cooperation to Collabroative Communities," *Educational Researcher*, vol. 28, no. 8, 1996, pp. 37-40.

[2] P. Dillenbourg, Ed. "What do you mean by 'collaborative learning?'" in *Collaborative-learning: Cognitive and Computational Approaches*, Oxford: Elsevier, 1999, pp. 1-19.

[3] G. Fischer, "Social Creativity: Turning Barriers into Opportunities for Collaborative Design," 8th Conference on Participatory Design (PDC '04), Toronto, Ontario, CA, 2004, pp. 152-161.

[4] A. Hofstein and V. N. Lunetta, "The Laboratory in Science Education: Foundations for the Twenty-First Century," *Science Education*, vol. 88, no. 1, 2003, pp. 28-54.

[5] D. Oblinger, "Boomers, Gen-Xers, and Millennials: Understanding the New Students," *Educause Review*, vol. 38, no. 4, July/August 2003, pp. 37-47.

[6] D. Oblinger and J. Oblinger (Eds.), *Educating the Net Generation*. Boulder: Educause, 2005.

[7] C. Raines, *Managing Millennials*, 2002, http://www.generationsatwork.com/articles/millenials.htm, accessed February 12, 2006.

[8] L. Springer, M. E. Stanne, and S. S. Donovan, "Effects of Small-Group Learning on Undergraduates in Science, Mathermatics, Engineering, and Technology: A Meta-Analysis," *Review of Educational Research*, vol. 69, no. 1, 1999, pp. 21-51.

[9] L. S. Vygotsky (Eds.), *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press, 1978.

[10] L. Williams, "Debunking the Nerd Stereotype with Pair Programming," *IEEE Computer*, vol. 39, no. 5, May 2006, pp. 83-85.

[11] L. Williams and R. Kessler, *Pair Programming Illuminated*. Reading, Massachusetts: Addison Wesley, 2003.

[12] L. Williams, L. Layman, J. Osborne, and N. Katira, "Examining the Compatibility of Student Pair Programmers," Agile 2006, Minneapolis, MN, 2006, pp. 411-420.