AC 2007-1495: EFFECTS OF THE TEAM-BASED APPROACH ON INDIVIDUAL LEARNING

Jason Pitts, Oklahoma State University

Patrick Teague, Oklahoma State University

Alan Cheville, Oklahoma State University

Charles Bunting, Oklahoma State University

Sohum Sohoni, Oklahoma State University

Effects of the Team-Based Approach on Individual Learning

1.Introduction

This study is a part of the ES21C project at Oklahoma State University. The goal of ES21C is to prepare OSU electrical engineering students to meet the challenges of engineering in the 21st century. The proposal for the ES21C project gives the following summary:

> "Engineering students learn what we teach them, but often do not become what we intend.[Students learn] the behaviors that let [them] succeed in classes, but these behaviors do not always correlate with success in engineering. Engineering Students for the 21st Century is a reform program for undergraduate engineering that is trying to align the behaviors that are taught in our program with those that help students succeed. To accomplish this we plan to transition from emphasizing acquisition of knowledge to emphasizing student development."[1]

Our research problem relates to the overall ES21C project, and by extension the transition to team-based learning[6], as a "devil's advocate" test. Many previous studies focus on the improvement of students' grades but they neglect the possibility of loss of conceptual and procedural understanding. We attempt to objectively determine the possible negative effects of team-based learning. We also hope to discover if the *pair-programming* model produces the same results as regular team-based classes or if there is a noticeable improvement in student understanding.

The paper describes the overall design of the experiment and outlines the method we propose to use to study the effects of various tiers of team involvement on the students' learning. This will consist of a "three-pronged attack" in which the lecture portion of the course will remain nominally unchanged while the lab section will use differing levels of team interaction. These interaction levels we will vary from each sample group and will attempt to determine the effects of each level and isolate the level that correlates to the highest student conceptual understanding in the non-lab course work.

The paper is also a call for participation in this study. The reason behind this call is that, despite the matching and removal of outliers, it will be difficult to present statistically significant results due to the small sample sizes of the different methods. Increasing participation, thereby increasing sample sizes, will greatly help increase the statistical significance of the results.

The remainder of the paper is organized as follows. Section 2 presents the related work and some background knowledge necessary for understanding the project. Section 3 describes our methods in detail. Section 4 lists the possible outcomes from this study. Section 5 calls for collaboration, inviting other instructors teaching a similar course to adopt our methods and participate in the study.

2.Related Work

This section presents background on the methods adopted by this study, namely *team-based learning* and *pair-programming*. Before we describe the two methods, we discuss different types of knowledge, a topic crucial for understanding the goals of this project.

2.1 Knowledge Types

In educational research, terms such as knowledge, have more than a single meaning. For example, Jeffery, an electrical engineering student, may have the ability to look at a circuit and identify parts as capacitors and resistors, yet he may have no idea how the circuit functions. On the other hand, Maude may be able to examine an RC (resistor and capacitor) circuit diagram and immediately calculate the circuit's response to a given input but have no idea how to build it from parts. A third student, Walter, may be able to look at an RL (resistor and inductor) circuit diagram, and based on his knowledge of inductors and of RC circuits, be able to calculate the response of the RL circuit without ever being taught explicitly how to solve RL circuits. He still may not know how to build the circuit like Jeffery though. All three students have knowledge of electric circuits, but they all lack important knowledge about electric circuits as well.

This leads to the discussion of different types of knowledge. One could say that the students exhibited procedural knowledge and conceptual knowledge, respectively. Hiebert and Lefevre discuss this in "Conceptual and Procedural Knowledge in Mathematics: An Introductory Analysis". They discuss that one can think of conceptual knowledge as a "connected web of knowledge", and a "network in which the linking relationships are as prominent as the discrete pieces of information"[4]. It is important to note that conceptual knowledge is not a piece of information, but rather can be a piece of conceptual knowledge "only if the holder recognizes its relationship to other pieces of information"[4].

The way one gains conceptual knowledge is by developing these links. In the above example, Walter is the only student showing conceptual knowledge. Although Maude can solve a circuit problem, she has been taught how to solve that specific type of problem; this is not conceptual knowledge. Developing conceptual knowledge means the student need not be taught how to solve the inductor problem directly; rather he can link the two existing pieces of knowledge.

The other type of knowledge Hiebert and Lefevre discuss is procedural knowledge. They state that it is made of two parts. The first part is "composed of the formal language or symbol representation"[4]. It is familiarity with the symbols and syntax of the problems. In the above example, Jeffery shows the first part of procedural knowledge of circuits. He recognized the parts (symbols) by their appearance. This is independent of whether he understands what they do.

Hiebert and Lefevre say that the second part of procedural knowledge is the "rules for completing … tasks." This is a set of "step-by-step instructions that prescribe how to complete tasks"[4]. In the example, Maude exhibits this second part of procedural knowledge. She knows how to solve the RC problem because she had been taught the steps. It is also important that she does the steps in order or she would not do the calculation correctly.

2.2 Team-based Learning

Many institutions have used team-based learning methods. It seems that there is an overwhelmingly positive attitude towards this method of instruction. Nearly all literature about team-based implementations of classes shows a highly successful method with most students showing increased learning and very few students exhibiting negative opinions about the method. This is not to say that all team-based implementations have been and will be successful; however, the great experiment that is team-based learning has shown little negative aspects to date.

One example of a team-based implementation in a programming class at the College of Computing and Information Sciences (GCCIS) at Rochester Institute of Technology (RIT) shows such a positive response[8]. This is important because most team-based studies have not been in programming classes. The drawback to this particular example is the small sample size of eighteen students.

The class was an upper division java programming course. The grade results of the pre/post test showed a small increase in overall grades. The only major difference is that one student failed the non-team course and none failed in the team course. An important piece of information gathered from the course was from an evaluation done by a member of the GCCIS Industrial Advisory Board who observed the course. His impressions of the class were very positive[8]:

- "Each of the teams seemed to genuinely have good chemistry…it seemed like the students like each other and respected the idea of respective strengths and leveraging skills."
- "They were competitive in a healthy good natured way…and I think they were interested in each team's interpretation of the task."
- "I like the enthusiasm of the teams. They were proud of their work and that is very healthy."
- "I saw nothing negative…and frankly am quite impressed by the level of teamwork demonstrated in the class…projects in the real world are built by work teams who learn to rely on one another…these guys demonstrated that in spades…if they walk out with a grasp of how teams need to be productive and interoperate then they will have learned a great deal more than simple programming."

At the end of the course, the students "were asked to rate how effective the overall course had been in achieving various student learning goals." They rated the goals in terms of effectiveness as such:

84% - Working together as a team
92% - Self-Directed Learner
100% - To be able to share ideas with members of your team
84% - Team-based Learning method of learning
92% - Favorable experience with the team

This study presents important observations from a third party. It helps to show that the data is not just the instructor's positive opinion on his own class. The students' opinions about the course show positive views about teamwork and communication, and considering this is the most important aspect team-based learning, next to the curriculum, it is good to see these high ratings[8].

Another study in team-based learning came from the Dept. of Electrical and Computer Engineering at Univ. of Missouri Rolla. The course was graduate level "Information Theory and Coding Theory." As with the previous study, this study suffered from a small sample size, however it did note that the number of students from one semester to the next was the same, and the ethnic composition of the classes were similar[7].

The data for this course was based on the instructor ratings from the students. The change in average scores in the instructor ratings for the entire department were compared with the change in average score for the course in question from the semester before team-based learning was applied and the semester in which team-based learning was applied. The average scores for the non-team-based classes increased by 0.2 (on a scale of 0-4). The team-based class increased 0.7. This is a 17.5% improvement compared with a 5% improvement in the non-team-based classes. The instructor also noted that the written statements from the students about the course were very positive[7].

Another study, called "Why Some Groups Fail," analyzes what causes students to dislike group work. The conclusions of the paper detail what not to do when forming groups, grading, giving feedback, designing the group activities. For example: do not allow students to form their own groups, do not make groups too small or too large (4 being a widely accepted ideal size), do not make group performance a minimal part of the grade, do not have a system that lacks peer evaluation, etc[3].

This is the only study we have come across that shows some of the negative aspects that can arise from group work. However, it only focuses on how to form groups and how to avoid some of the pitfalls in designing a course using team-based learning. In the available literature on team-based learning, no study has tried to analyze if students lose important knowledge in the process of successfully using teamwork as a learning tool. This is a key component of our project. We ask the question, "For a successful group, whether measured by grades or evaluation, do we still fail to impart vital conceptual knowledge to certain members of the group or team?" We propose to determine whether team-based learning will fail to facilitate a student learning the material due to the manner in which the teams or groups divide the work for the project(s).

2.3 Pair-programming

Pair-Programming, or the XP (eXtreme Programming) method, has been shown to be an incredibly effective method in industry. The basic method is that that two programmers work together on a piece of code on the same workstation or terminal with one acting as the "driver". This "driver" actually types and inputs all the code, while the other programmer watches and offers advice both methodically and in syntax correction. After an amount of time the two programmers switch roles and continue. The idea behind this method is that new viewpoints are exposed that can lead to a better product and since a free exchange of ideas is occurring a synergistic hybrid code can be created that is better than the versions either programmer would have produced separately. Also the very fact that two pairs of eyes are watching translates to an increased chance of spotting errors early and thus less time is spent debugging the code[2].

Research on the effects of pair-programming in a learning environment found that students using the pair-programming method produced better code, quicker than their solo-programming counterparts. However it was found that there was no statistical variation on exam scores between paired students and non-paired students. It should also be noted that a much higher percentage of non-paired students dropped the course midway through the semester, 24.1% compared to 7.6% of paired students.[5]

3 Methods

The experiment to be performed will consist of three stages. In each stage a different method of interaction will be applied to (and used in) the lab section of the course. Each method will be applied to a separate semester, so that students are exposed to only one. These methods are as follows

1. Students will work individually on lab assignments. Each student will be solely responsible for her own work. Lab assignments will be of a nature such that they are challenging but not overly taxing for a single person to accomplish in the given time frame.
2. Students will be assigned teams to work with for the entire semester. Each team will be composed of four (4) members. The team will be evaluated based on their combined performance with individual adjustments based on peer evaluations. Lab assignments will be of such a nature that the team must work together to accomplish the goal, and that no one member could complete individually.
3. Students will be assigned a partner to work with throughout the semester. The pair will be required to work together in lab and will be monitored to ensure they are using the *pair-programming* model. Students will be scored based on the pair's performance. Lab assignments will be tough enough to ensure that one member cannot easily "carry" the pair.

3.1 Benchmarks of Success

The relative success of each method will be determined by a number of benchmarks.

1. Concept and procedure inventory administered once at the beginning of the course and again in the end to determine the students' improvement and retention of the material taught in the course. This inventory will remain unchanged throughout all three semesters.
2. Specific exam questions tailored to determine if the student gained an understanding of the material beyond what could be expected from lecture (i.e. gained through lab work). These questions will change slightly between each semester to ensure students are not able to glean the exam answers from previous students.
3. Students from methods 2 and 3 will fill out peer evaluations and be interviewed to determine which areas each student was involved in.

The primary purpose of this experiment is to determine if not being exposed to a topic in the lab environment affects a students' ability in traditional performance measurements such as exams. As such a major objective in the collection of data is to determine with which areas of a project a particular student dealt.

For method 1 it is obvious that the student is exposed to all areas and concepts of a lab which she attempts to complete. Likewise in method 3, the student-pairs are forced to work together on the same portion of the lab, so they will both be exposed to all areas and concepts of each lab attempted.

Method 2 poses a different problem however; since the group is free to complete the project any way they choose, it is very possible that the students will develop a division of labor and that not every student will be exposed to certain aspects of the code. Indeed as the semester runs its course the students may develop certain specializations and thereby exclusively deal with the particular concept for the entire group. This is why it is vital that the students are made to declare what they worked on in the project so that this data can be figured into each individual student's performance.

3.2 Potential Problems and Solutions

Across the three semesters a number of variables will change that need to be kept in mind. Firstly the method of lab work will be changed. This is the variable that we have most direct control over, and of which we are attempting to track the effect. The scholastic ability of individual students will also fluctuate over the three semesters. In a large enough sample these fluctuations should average out, but for our experiment our samples will consist of around twenty (20) students each semester which is too low to assume statistical balancing. To combat this we will use a form of *post-facto* matching to eliminate these fluctuations from the class average comparisons. This matching will be based on factors such as test scores, GPA, project scores, classification (junior, senior, etc.). We intend to use post-facto matching only if large outliers are detected and will initially match students for comparison based on their exam grades.

By administering the concept battery once at the beginning of the semester we can determine those students who come into the class with a certain understanding of the concepts dealt with in the class. This knowledge will allow us to set these students aside when determining the effectiveness of our lab methods since the method was not responsible for teaching the student. An example: Hitomi learned about analog-to-digital conversions (ADC) in her Digital Logic class the previous semester. Once assigned to her team it was decided she would work on a code section involving hardware interrupts and not the section dealing with ADC. If on the exam she then performed well on questions involving ADC it would be erroneous to conclude that she gleaned this knowledge solely from lecture and that not participating on the ADC lab section did not affect her grade.

To minimize possible variations in student experience we will be keeping certain factors constant. The instructor and teaching assistant will remain unchanged to negate any difference in teaching styles that may affect the students' performance. Also, throughout all three semesters we will focus on the same family of microprocessors and use the same simulator for all lab work.

4. Possible Outcomes

We anticipate a number of outcomes that the data collected from this process may lead us to. The first is simply inconclusive results. We are dealing with small samples and if the differences between them are not very pronounced it may be difficult to determine which of these styles allow the student to gain a thorough understanding of the class material. Second the results may show that exposure (or lack of) to material in the lab environment does not affect a student's performance on her exams. It should be kept in mind that this lack of correlation could also be linked to the particular skill or concept and could not be generalized across the whole class. Next, it may be seen that the move to team performance actually alters the way students learn the material negatively or positively and whether or not the pair-programming model amplifies or reverses those changes.

5. A Call for Participation

This study, like most others in engineering education research suffers from a small sample size in its current form. Despite our best efforts to keep as many environmental factors constant through the three implementations, the results are bound to lack statistical significance. Since the instructor, the TA and the students are all human subjects with unpredictable behavioral characteristics (as opposed to chemical reagents or microprocessors), we present this paper as a call for participation, more than an experiment with a certain factual conclusion.

We ask the research question, "For a successful group, whether measured by grades or evaluation, do we still fail to impart vital conceptual knowledge to certain members of the group or team?" We propose to determine whether team-based learning will fail to facilitate a student learning the material due to the manner in which the teams or groups divide the work for the project(s).

The methods outlined in the previous sections provide an experimental framework to answer the research question. If implemented by a number of educators at various institutions, the experiment will have the sample size and the variation within the sample set to provide statistically significant outcomes. We ask for your participation in this study.

The authors acknowledge support from the National Science Foundation through award NSF0530588. Mr. Pitts and Mr. Teague would like to acknowledge support from a National Science Foundation REU award (NSF0631565) which enabled them to conduct the research presented in this paper. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Bibliography

1. Cheville, Alan, et al. Engineering Students for the 21st Century. 2006. 3 Dec 2006. <http://es21c.okstate.edu/project/links/summary.htm>.

2. Cockburn, Alistair and Laurie Williams. "The Costs and Benefits of Pair Programming." 2001. 21 June 2006. <http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF>.

3. Feichtner, Susan and Elaine Davis. "Why Some Groups Fail: A Survey of Students' Experiences with Learning Groups". J. Organ. Behav. Teaching Soc., vol. 9, pp. 58-73, 1984.

4. Hiebert, James and Patricia Lefevre. "Conceptual and Procedural Knowledge in Mathematics: An Introductory Analysis." Conceptual and Procedural Knowledge: The Case of Mathematics. Ed. James Hiebert. Hillsdale, NJ: Lawrence Erlbaum Associates, 1986. 1-27.

5. McDowell, Charlie, et al. "The Effects of Pair-Programming on Performance in an Introductory Programming Course." 2002. 21 June 2006. <http://www.cse.ucsc.edu/~charlie/pubs/sigcse2002.pdf>.

6. Michaelsen, L.K., Watson, W.R., & Schraeder, C.B. (1982). Team-based learning: A potential solution to the problems of large classes. *Exchange: The Organizational Behavior Teaching Journal7*(1): 13-22.

7. Weeks, William. "Team-Based Learning in: 'El Eng 447: Information Theory and Coding Theory.'" 2002. Team-based Learning. 12 January 2007. <http://atlas.services.ou.edu/idp/teamlearning/docs/Course by Wm Weeks.pdf>.

8. Whittington, Keith. "Team-Based Learning in: 'Applications Programming.'" Team-based Learning. 12 January 2007. <http://atlas.services.ou.edu/idp/teamlearning/docs/Whittington TBL Programming Course.pdf>.