Establishing Individual Accountability for Learning in an Exam-less, Group Project Course

Paul Stanton, Duane Fairfax
Department of Computer Science
United States Military Academy
West Point, NY

Abstract

We work toward a process model to establish individual accountability for learning in an exam-less, group-project course using an evolutionary approach and *vertical slicing.* We allow students to evolve into self-directed, cooperative learners over the duration of the three courses that comprise our Computer Science Engineering Sequence and then evaluate their learning using a group project. The first course is strictly individual effort, the second course combines group and individual work, and the sequence culminates with an exam-less, group project course. Students benefit from opportunities to learn how to succeed in a group setting over time, while we increasingly adjust the allocation of evaluation points from individual to group effort. In our final course, we employ cooperative learning using vertical slicing where students select a component of the system whose development they are responsible for throughout each phase of the engineered solution. This vertical slicing allows us to develop homework assignments that can be completed individually but then combined into the group's collective effort after submission for grading. The homework assignments are not superfluous, but rather contribute directly to the completion of the team's project. In the end, all of the students are exposed to the breadth of the problem domain, they benefit from the cooperative learning environment, and the instructors have an opportunity to grade individual effort. We effectively enforce individual accountability without detracting from the group's progress toward completing the project or introducing unnecessary examinations.

I. Introduction

Over 600 studies in the past 90 years have been dedicated to validating the assertion that students learn better when working together in small groups[1]. Whether referred to as cooperative learning, collaborative learning, or simply group work; the results of the research are consistent: students retain information longer, students perform better during evaluations, and students appear more satisfied with the course material[2]. Based on these results, cooperative learning has long been accepted as a pedagogical approach. However, instructors must establish the appropriate conditions for learning in a group setting for the technique to be successful. Most undergraduate students have limited experience working in an environment where their success depends on others. Thrusting students into a group setting where their grade is predominantly comprised of group work can result in negative experiences that detract from learning. Easing students into group work and allowing them to evolve as cooperative learners reduces the potential for negative consequences while simultaneously increasing the opportunities for them to learn from each other. We promote this evolution by gradually increasing the group requirements and the associated evaluation points across our sequence of three courses.

As the group requirements increase, we are careful to continually enforce individual accountability. In order for the faculty to maximize the benefits derived from a cooperative learning environment we must ensure that each individual student learns all of the course objectives while working in teams. This is where we have employed *vertical slicing* of project material to force individuals to complete a *slice* on their own that can be evaluated and graded independently, but then incorporated into the group effort. Using this technique the students select a component of the system whose development they are responsible for throughout each phase of the engineered solution. The teammates work on the individual components separately but in parallel. After submission for grading the team members incorporate the components into a functioning system.

In the process, they explain their products to the rest of the team, they discuss their problems, and they develop solutions together. Each student contributes to the team, team members are able to capitalize on each other's strengths, and the students learn from each other. Instructors, meanwhile, are able to evaluate individual performance.

Several traditional methods for establishing individual accountability have been researched and employed successfully: using past performance metrics to assign students to groups, employing peer assessments, and individually administered exams. Using these techniques in concert has proven most beneficial, but not all courses can benefit from the combined effects of these traditional approaches. Specifically, there are many project-based courses that do not incorporate exams into their evaluation methodology, leaving a significant void in the ability to enforce individual learning. Some courses are designed to teach concepts that are not testable during an examination period, but rather require a significantly longer time frame for proper evaluation. Semester-long, project-based courses fall into this category. The dilemma then becomes finding an alternative method for ensuring that individual group members actually learn the material without simply taking credit for their teammates' efforts. The goal of this paper is to demonstrate an applied approach in a cooperative learning environment whereby individual learning is enforced through student development in a group setting and evaluation of individual effort within a slice of the group assignment.

The remainder of this paper outlines our approach as follows. Section II describes the Computer Science Engineering Sequence. Section III describes the tenets of cooperative learning, potential problems, and traditional approaches to enforcing individual accountability. Section IV discusses vertical slicing. Section V discusses our initial results of the proposed approach. We end with a summary and conclusion in Section VI. Our methodology for allowing students to evolve as cooperative learners in our Computer Science three course sequence is discussed throughout the paper.

II. Computer Science Engineering Sequence

The goal of the three courses in our Computer Science Engineering sequence (CSES) is to ensure that students majoring in disciplines other than computer science develop a foundation for effectively employing computer science fundamentals and Information Technology to solve real world problems. One of the desired outcomes for the CSES is for students to learn to effectively function as part of a diverse team to accomplish a common goal.

CS300, Introduction to the Fundamentals of Programming, is the first course in the sequence. It is strictly an individual effort with traditional homework assignments and exams. As the second course in the sequence, CS350, Database Design & Implementation, employs an evaluation strategy that combines group and individual effort. There are three exams and three individual homework assignments that account for 65% of the course work. The remaining 35% is allocated to a semester long term project which requires students to identify, analyze, design and implement a database application. Students are closely coached through the complexities of team dynamics to develop as cohesion and to effectively manage the project to completion. IS450, Principles of Distributed Application Engineering, is the final course in the sequence. The entire course is structured around the development of a working system in teams. Students complete a series of individual assignments worth 42% of the course grade which are incorporated into the team product. The team's collective intermediate submissions and the final system account for the remaining 58% of the course points.

	CS300	CS350	IS450
Individual	100%	65%	42%
Team	0%	35%	58%

Allocation of Points Across the CSES

III. Description of Cooperative Learning

Cooperative learning occurs when students work together in small groups to accomplish a collective task[3]. Research has shown that, when employed properly, cooperative learning can result in improved conventional academic achievement such as performance on standardized tests. Of equal, or more importance, the research also reports that a well constructed cooperative learning environment can contribute to developing conceptual skills needed for problems requiring critical thought. It can also improve social and leadership skills gained through group member interaction.

We wanted to capitalize on these potential benefits by having groups of 3-4 students work together to first construct a database in our CS350 Databases course and then to build a web-based distributed application in our IS450 Distributed Applications Engineering course.
These benefits, however, are not automatically achieved, but rather instructors must place considerable thought into how to implement the cooperative technique. There are several key elements that must be present in order for students to learn in a cooperative environment[4]:

1. Positive interdependence. Students within a group must be forced to rely on one another to be successful on their project or homework. The scope of the work must be such that it is impossible for the team to do well (finish the work and receive a good grade) without considerable contributions from each member.

2. Individual Accountability. Instructors and team members must have a method of holding each person accountable for his or her contribution. Moreover, each student must learn all of the course objectives; learning only a subset is not sufficient.

3. Face-to-face interaction. Some work can, and should, be separated out and completed in parallel, but students must still be forced to interact directly with one another. The nature of the tasks for the work should allow for a division of labor, but they must also require a degree of integration that can only be accomplished collectively.

4. Appropriate use of collaborative skills. Students must learn how to interact with others and develop leadership, decision-making, communication, and conflict-resolution skills that will be required by students upon graduation.

5. Group processing. The team has to approach the overall work from a group perspective. The members must establish mutual goals, a collective timeline, and group policies to keep the team focused. Additionally, they must periodically assess their collective performance and make adjustments as necessary.

A. Problems Encountered without Individual Accountability

To establish an environment that includes all of these components is ambitious considering the population of students. Most undergraduate students will have been raised in a classroom environment that is very individually focused where they have only been responsible for themselves. They will have limited experience in working with, helping, and relying on others. Inevitably there will be some resistance to a cooperative learning strategy from a subset of the student population. Strong students may resent having to "pull weaker students along," team members may have considerably divergent goals (one member may be striving for an A in the course while others may be content earning a C), a subset of the group members may be overly assertive to the point that they dominate all group activity, or select individuals may not put forth the degree of effort expected by the team. Identifying and understanding the potential problems in a cooperative learning environment is a critical step in determining how to minimize the impact that they may have on individual accountability. It is the instructor's job to develop an environment that prevents or minimizes the effects of these potential problems. This responsibility begins with team creation and persists throughout the course via continual assessment and evaluation.

B. Traditional Methods of Enforcing Individual Accountability

1. *Creating Groups*

The first step in promoting individual accountability in an environment suitable for cooperative learning is to build the teams in a productive manner. Placing students in effective teams requires considerable forethought in order to account for each of the elements of a successful cooperative learning endeavor. The research is consistent in recommending instructor-picked, heterogeneous groups of 3-5 students[5]. If left to select team members by themselves, students would not necessarily keep their individual learning as their primary goal. Stronger students may gravitate to one another leaving the weaker students to flounder, or students may overly weight the significance of friendships and social acquaintances. In our sequence, instructors examine the students' background and interest to determine the best possible combinations during team formation.

We leverage the structure of our course sequence to improve our ability to place students in groups in our final course. CS350 and IS450 use a directed approach to place students into project teams. In CS350, the students are placed within a team by the instructor without any input from students. In contrast, the directed approach is slightly relaxed in IS450 as students are afforded the opportunity to submit a preference of team members for consideration in the team formation process. As more mature cooperative learners, the students are given this opportunity because a majority of them will have experienced the pitfalls of team dynamics and have experience determining what characteristics a compatible team member should possess. The evolution process takes the students through a maturation process that develops them into life long learners capable of functioning effectively as a project team member. Additionally, the CS350 course director can provide invaluable insight to the IS450 course director having evaluated the project teams in the Database Design course. The insight from the CS350 course director is more realistic than a statistical analysis because it is predicated upon anecdotal evidence that can help determine the compatibility of students in order to avoid structuring teams that would potentially be dysfunctional.

Moreover, a student's academic interest and past performance are equally important as team member compatibility. The project teams are developed with a slight variance in academic backgrounds to provide opportunities for stronger students to help weaker students without combining students with divergent goals. This allows the students to benefit from the experience of explaining difficult concepts to others without creating an environment where some students do all of the work. We also structure a proper heterogeneous mix that accounts for other types of diversity such as differences in academic interests, gender, and ethnicity. This provides opportunities for students to apply their strengths toward the group effort and therefore increase the positive interdependence that promotes a sense of satisfaction and motivation to continue learning. Lastly, the actual size of the group has a critical impact on individual accountability. The right sized team can maximize collaborative effort while minimizing potential problems. If the group is too small, individuals can easily dominate group sessions, or there may be insufficient diversity of insight or skills to enhance learning. On the other hand, if the group is too large then select members can easily avoid working, some quieter members may simply be ignored, or there may be insufficient work to keep all members occupied. We build our teams with 3-5 students in accordance with recommendations from the research[5].

2. *Using Peer Assessments*

After initially assigning students to groups, instructors must then provide direction, guidance, and continual observation of the teams' progress. As part of the guidance, instructors should ensure that each group has a method for policing itself in the form peer assessments. In order for students to embrace the cooperative learning environment, they must feel that there is a method of ensuring fairness in grading. Nothing will demoralize students faster than for a non-contributing student to receive a high grade based solely on the other group members' efforts. Research shows that students derive a much greater sense of satisfaction and higher test scores from groups that have the ability to provide a peer assessment that is factored into grade calculation[6]. The peer assessment should reflect the degree of contribution each team member makes toward the collective effort, but determining what the measure of the contribution should be requires thought. Intuitively one would consider the amount of actual work accomplished, but this may unfairly benefit a stronger student and create an unintended sense of competition within the group. In place of accounting strictly for academic ability, instructors may use a concept of team citizenship that measures an individual's cooperation with the team and the willingness to help others[7]. This promotes teamwork that will foster a constructive cooperative setting and benefit learning.

The peer assessments can be used to adjust individual grades from the group's assigned grade. The degree by which an individual's grade should be adjusted is calculated based on the peer assessments provided by all teammates. Several studies suggest varying methods for integrating the peer assessments into an individual's overall grade, but they consistently propose limiting the scope of the adjustment to one letter grade above or below the group's grade. As additional guidance, the instructors should provide the students with multiple opportunities to conduct peer assessments over the course of the semester so that an individual can overcome an initial negative rating. Since negative ratings may cause unwanted conflict between group members, the assessments should be conducted anonymously. This also increases the likelihood that individuals will provide an honest assessment.

Both CS350 and IS450 employ a peer rating scheme. Each course is structured with a series of In-Progress Review (IPR) briefings that coincide with interim submission requirements. As part of the submission, students provide feedback on the degree of effort that each team member has provided. These periodic peer ratings provide the instructor with additional insight into the team dynamics. The instructor gains a perspective on how the group interacts while they are away from the classroom and can then help focus and guide the team. The instructor can also address anomalous ratings and help the group resolve problems with specific individuals who refuse to cooperate.

Peer assessments can be a valuable tool, but instructors must keep in mind that students will likely have very little experience working constructively in groups and they won't know how to best employ the tool; instructors must provide considerable guidance. Students will probably lack sufficient social and leadership skills required for conflict management, so they will need help determining a plan for their group interaction. A recommendation is to have the teams develop both a team policy document and an expectations document at the beginning of the course that provide a foundation for handling future problems[5]. CS350 coaches students into this process.

The team policies address the administrative rules for the team's operations: roles with associated responsibilities, procedures for group meetings, methods for submitting assignments, and strategies for dealing with uncooperative team members. The expectations document permits the team to collectively determine a realistic goal for the course. The documents then help students determine an appropriate peer rating by judging whether an individual has complied with the operations and lived up to the expectations. They also provide a guide for holding each other accountable prior to when a potential conflict arises.

Peer evaluations will assist an instructor in determining if all individuals are contributing to the group effort, but they can be misleading. Group members may find it socially difficult to provide an accurate assessment of their peers (even in an anonymous setting) resulting in peer evaluations that provide a false representation of the individual effort. Also, while the peer assessments help to ensure that everyone is contributing toward the group goals, this does not necessarily mean that each student understands each objective for the course. Instructors need additional tools to enforce individual accountability.

3. *Using Group Roles*

Students must assume some responsibility for their own learning, but teachers must steer their group interaction in a positive direction. The teacher should mentor the teams to ensure that they are properly distributing the work instead of isolating tasks to particular individuals. Keeping in mind that many students will naturally gravitate toward a "divide and conquer" approach, instructors can dictate that team members assume particular roles during portions of the course and that the roles they rotate periodically. This forces students to be exposed to the breadth of the problem that the group is trying to solve. Each individual should assume each of the following roles or some suitable variation during the course of the assignment: coordinator (organizes tasks and assigns responsibilities), checker (monitors the team's solution for correctness, completeness and accuracy), recorder (writes the solution), and skeptic (plays devil's advocate to ensure various perspectives are considered in determining the final solution) (Johnson, 1999). These administrative responsibilities are in addition to performing work toward the actual solution, but they are essential to reinforce the use of collaborative skills.

Instructors can also mandate that different group members present the solution. Presenting the group's work implies that the briefer has a thorough understanding of the entire solution and not just the part that he or she worked on directly. To develop this understanding requires considerable face-to-face collaboration within the group where each member explains the portion that he or she worked on. In CS350, the instructor assigns roles to the groups and coaches the students into understanding the responsibilities of each person. There are several ungraded opportunities for students to explore the roles before being evaluated. In IS450, students are expected to have matured as cooperative learners. The instructor does not assign roles, but rather requires that the students select one team member to conduct one of the required briefings in its entirety. There are four briefings such that each student must brief at least once. The entire team's grade is dependent on the success of the individual briefer. All team members have a vested interest in ensuring that the briefer is completely knowledgeable of the entire system. Experience has shown that, the coaching employed in CS350 results in the students naturally assigning roles within the team and selecting the "coordinator" role for the briefer.

4. *Giving Individual Exams*

In a traditional course, individual exams that test all of the objectives provide a proven method for enforcing individual accountability. This prevents the situation where there is "only a group product, demonstration, or performance to be evaluated, [leaving] no mechanism for individual accountability."[8] The pressure of an impending exam will motivate individual students to look at all of the testable course material. The results of the exam will serve as a clear indicator of who understands the material and who does not. We apply this approach in our first and second courses in the sequence. As we ease students into the cooperative environment and allow them to evolve as learners, we begin the sequence with traditional grading strategies.

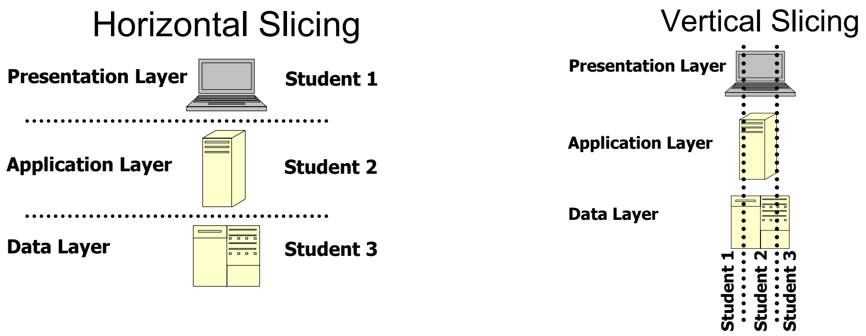
IV. Individual Accountability through Vertical Slicing

We did not want to employ an exam in our final course, however. There are some project-based courses where instructors cannot accurately assess an understanding of the course objectives during a test; the objectives are larger in scope and require students and teams to try, fail, and re-try as part of the learning process. Our distributed applications engineering course required students to develop a web-based application over the course of the entire semester. This required the teams to design a comprehensive system, build a database, develop application logic that runs on a web server, and then implement a web-based client interface. The software engineering lessons, the skills associated with connecting the multiple components, nor the lessons derived from project management could be properly evaluated with an exam. So we elected to set the conditions for establishing individual accountability with a combination of traditional methods and our concept of vertical slicing.

If left on their own, students may be inclined to “divide and conquer” tasks in a horizontal manner where one student works on tasks associated with one course objective or phase while other students work on separate tasks associated with distinct objectives[7]. The problem with this horizontal method is that at the end of the course, each student only understands the concepts associated with a portion of the course material. In the distributed applications course, one of the primary learning objectives is for each student to develop competence in developing all aspects of a distributed application. The students need to understand how to develop the client side graphical user interface, the web server logic, and the database logic. Without supervision, students may try to work within their comfort zone and divide the work such that each student only works on one of these components.

This is the problem that we initially faced in our sequence. While observing the final presentations from two previous semesters it was obvious that individual group members had a very superficial understanding of the entire system if they had one at all. They had divided the work responsibilities in a horizontal fashion and were so isolated in their learning that they had to rotate through the group members during the final presentation to explain the system. The person who had developed the graphical user interface explained the design decisions and associated code, the person who built the database explained how the queries operated, etc. It was disconcerting that no one student could explain how the entire system worked.

Witnessing this outcome, we decided that a more suitable method of dividing the work requirements is to develop a “vertical slice” of the tasks. The components of the system do not change, but rather the way that the students address them. Using vertical slicing, each student must handle a portion of the tasks at each layer; they must develop the breadth of understanding required to meet the course objectives by constructing a portion of the system at each stage. In terms of our distributed application, this entailed an individual student developing tables in the database, writing code for the server-side application layer, and creating the associated web pages for the client layer. We structure the scope of the projects such that students can select a portion of the functionality that they can work on at each layer. For instance, if a student was responsible for adding a person to the system the student would develop the client-side HTML forms, the server-side scripts for handling the data and communicating with the database, and the tables in the database that contain person-related information.



Deciding to employ vertical slicing required us to establish an evaluation strategy for the course where the assigned homework provided value added to the group project rather than being extraneous isolated tasks that detracted from the team effort. By design, the scope of the group projects is such that it requires considerable time and effort from all members of the team in order to complete even the minimal functionality by the end of the semester. Adding additional homework assignments would place an unrealistic burden on the students. The key is to select projects that can be separated into distinct portions for parallel development, but that still require considerable integration at the group level. If selected successfully, students are first forced to work independently to learn new concepts, but then they combine their efforts as a team to reinforce the same concepts collectively. This gives the instructor the ability to evaluate each individual's efforts while still capitalizing on the benefits of cooperative learning.

V. Results

In order to validate our process, we have had 67 students over the past three semesters participate in an end-of-course survey that asks them to reflect on what they have learned and to explain whether or not they felt comfortable building an entire distributed application. We use their answers in conjunction with their final course grades to determine if students are learning in this environment. 60 of the students felt that they could build a system. Of the 7 that admitted that they did not understand all of the course material, 5 assessed their own level of effort in the course at far less than what they felt was required to learn. Of note, all seven of the students who claimed that they did not learn the material received a C- or lower as a course grade. The individually-based homework assignments provided the instructors with the ability to identify the students who were not learning the material and assign them a grade commensurate to their performance.

At the same time, the instructors were able to protect the remaining team members from the impact of a single student's lack of effort. The seven students were all on distinct teams, one of which had a C+ as a group average, but the other six had a B- or better average for the other team members. This is significant because the one student from the team that admittedly did not put forth the effort in the group project did not significantly affect the other team members' grades.

In addition to assessing the ability to build a system, we asked the students to assess their level of contribution to the team effort. 23 students felt that they contributed more than their share, 26 thought that they contributed a proportionate amount, and 18 felt like they had contributed less than their peers. However, of the 23 that felt like they worked harder than their teammates, only six thought that their level of contribution was greater than 10% more than it should have been. This means that within most of the groups, the members provided a relatively similar degree of effort.

VI. Conclusion and Future Work

Many researchers have addressed the benefits of cooperative learning and how to maximize the potential for these benefits by creating the best possible environment. A key component to this environment is ensuring individual accountability in a group setting. The research addresses several methods that instructors may use to construct a suitable environment, but the research is not exhaustive. Specifically, there is no research that focuses on courses that do not lend themselves to comprehensive individual examinations. Without the benefit of an individually based exam, instructors may be left without a comprehensive means of ensuring individual accountability. We have initiated an evolutionary process in conjunction with vertical slicing to address this problem.

We are currently in our third semester of implementing our process. Our future work will contain detailed statistical analysis to assess the effectiveness of the process in greater detail.

References

1. ERIC Development Team (1992) "Cooperative Learning: Increasing College Faculty Instructional Productivity. ERIC Digest," ED347871 1992-02-00.

2. Davis, B, (1993) "Tools for Teaching," John Wiley & Sons, Inc., San Francisco, CA, 1993.

3. Slavin, R. "Cooperative Learning." Review of Educational Research, (1980) 315-342.

4. Johnson, D., Johnson, R., Smith, K., "Cooperative Learning: Increasing College Faculty Instructional Productivity", ASHE-ERIC Higher Education Report No. 4, George Washington University, 1991.

5. Oakley, B., Felder, R., Brent, R., & Elhajj, I. (2004) "Turning Student Groups into Effective Teams," Journal of Student Centered Learning, Volume 2, No. 1, New Forums Press, Inc, 2004.

6. Kaufman, D.; Felder, R. "Accounting for Individual Effort in Cooperative Learning Teams, "J. Engineering Education, (2000), 133-140.

7. Felder, R., Brent, R., "Cooperative Learning in Technical Courses: Procedures, Pitfalls, and Payoffs," ERIC Document Reproduction Service Report ED 377038, (1994).

8. Cooper, J., Prescott, S., Cook, L., Smith, L., Mueck, R., "Cooperative Learning and College Instruction: Effective Use of Student Learning Teams," California State University Foundation, 1990.