Strategies for Effective Group Project-Based Courses

by
Frederick C. Harris, Jr. and Carl G. Looney
Department of Computer Science
University of Nevada, Reno
Reno, NV 89557
fredh@cs.unr.edu looney@cs.unr.edu

Abstract: *The fact that people are more productive when working together in groups has been accepted in industry for a long time, but it has not been integrated into our educational system broadly or effectively. In this paper we describe our strategies for effectively incorporating projects into courses and how we are implementing them in our Department, and some problems and pitfalls we have encountered along the way.*

Keywords: group projects, cooperative learning, engineering education, senior projects

1. Introduction

The traditional classroom approach at most of our institutions is familiar to almost anyone. It involves whole-class instruction, recitation, homework problems, and tests. A lecture with questions is followed by homework assignments. Tests are scheduled to cover the designated material. That is the way we were taught and that is the way we assume that teaching is to be done.

Unfortunately this classroom approach does little to prepare students for a work environment. Companies typically organize employees into project teams with a team leader. The teams meet and discuss their tasks and then break the problem down into parts and interact throughout the design and implementation to arrive at a total solution. Each worker has a value to the company according to one's productivity in such an environment. According to companies that have interviewed on campus and to the feedback from companies where our graduates work, they prefer students who have exposure to group work. Such students have been exposed to cooperating with people from different backgrounds and cultures and to working on tasks that jointly contribute to the overall project.

The fact that people are more productive when working together in groups has been accepted by industry for a long time. However, group learning has not been integrated into our educational system broadly or effectively. The reasons for this include issues that lead to questions by the instructors such as, "How am I supposed to assign individual grades?", "How do I deal with groups and their dynamics?" and "Is it worth the extra work?" The process is more complicated than conducting a class of traditional format in that it requires choosing group size, partitioning the class into groups and then selecting projects. The groups are charged with the responsibilities of planning, analyzing, specifying, designing, programming and documentation. The *raison d'etre* for giving a project-based course is to let the students develop joint problem solving skills in the course subject. Success in the job market is also a goal.

In this paper we first present our experiences and approaches. Then we review the literature for another perspective. We describe strategies that we have used and lessons that we have learned for effectively incorporating group projects into courses. We describe how we are implementing them in certain courses in the Computer Science Department at the University of Nevada, Reno. This implementation covers selected undergraduate courses, graduate courses and graduate/undergraduate courses. We present some problems and pitfalls encountered along the way.

2. A History of Our Group Project-Based Courses

2.1 Our Initial Group Projects.

When the Computer Science Department at the University of Nevada, Reno was formed, the adopted curriculum included a *Senior Projects* course that was modeled after such courses in the College of Engineering and the School of Mines. The description for this course became: *CS 426 Senior Projects - Supervised group or team projects with emphasis on implementation of engineering design*.

Over the years the structure of *Senior Projects* has been modified so that it is now integrated with the semester course *Software Engineering* into a year long sequence. This sequence allows the students to learn proper design methodologies and practice them on the project that they implement the following semester. This integration forces the selection of projects a bit earlier in their academic career and creates a problem with those students who cannot decide what they want to do. Therefore, the instructor of this class has created a list of projects that are appropriate for this level.

The groups were initially selected by the students, and the project was identical for every group. This setup allowed the students to work as a group and easily interact with the instructor since there was only one project for the instructor to keep up with. Task breakdown within the group was left to the members with no interaction from the instructor and there were only intermediate and final presentations to be done in front of the class to demonstrate the project success.

2.2 Group Projects in Graduate and Graduate/Undergraduate Courses.

In addition to the *Senior Projects* course, many of our faculty have incorporated term projects into their courses at the graduate/undergraduate and graduate levels. The format of the projects varies with the course and instructor, but the trend is clearly toward more term projects in graduate courses. Part of this appears due to the need expressed by students and employers for hands-on and cooperative experience rather than the usual questions in limited contexts on homework and tests. Our graduate students are from the MS program in Computer Science and the MS and Ph.D. program in Computer Engineering. There is also an increasing number of graduate students from other sciences and engineering who take classes in Computer Science.

A purely hands-on type of course is not sufficient for graduate students, so the lectures must cover the basic theories, methodologies and derivations of equations during the first half of the semester. When students plan their projects they must contemplate which methods to use and how to integrate the entire project, which exposes many of them to actual (rather than contrived) problem solving situations. It is sometimes a rather traumatic experience for students who have always had a textbook page or section from which they could extract the solution technique for a given textbook problem.

Examples of courses that have been conducted with group term projects include *Modeling and Simulation*, *Fuzzy Systems*, *Expert Systems*, *Genetic Algorithms* and *Neural Networks*. The mixed graduate/undergraduate courses include *Image Processing*, *Pattern Recognition*, *Object Oriented Programming*, *Artificial Intelligence*, *Operating Systems*, *Compilers* and *Internet Programming*. Courses that specifically have not used this format include courses where essential theory must be mastered and demonstrated on tests, such as *Theory of Computing* and *Analysis of Algorithms*.

The graduate courses and graduate /undergraduate courses that we have presented with group term projects have had some surprisingly good results, but some problems have occasionally arisen. A positive result is that groups sometimes find new methods in the literature that work well or modify a given method for a particular application. An undesirable result, however, is that a group may have trouble getting the project underway or may not be making much progress at a point dangerously late into the semester. In mixed courses, it is possible that one or more groups may contain both graduate and undergraduate students. The instructor should remain aware of the progress in these cases or prohibit such groups from being formed, although our experience has shown that the top undergraduates perform better work than many graduate students.

2.3 Group Projects in Undergraduate Courses.

There are several things that we have learned while integrating group projects into undergraduate courses that are different from graduate and graduate/undergraduate courses. These relate to the setup and initialization of the project and the groups

Typically projects in these courses are smaller than the projects in the graduate courses in both scope and time requirements. Therefore, these projects must be more structured, removing a lot of the preliminary decision making out of the students' hands and placing it into the instructor's.

For example, in our *CS I* and *CS II* courses, there is a 3 week project at the end of the semester that is designed to tie all the topics together and help the students have a better grasp of all the material. The large lecture format makes it infeasible for the instructor to help students select individual project topics. Therefore, groups are all given the same project. In *CS I* this project is done by individuals in order to get them used to Computer Science projects, while in *CS II* the project is a small group project and the students are allowed to choose groups of 2 or 3 within their own lab section.

In Programming Languages the students are given more flexibility in the project selection, as long as that project covers some topics from an instructor provided list This class has also integrated more accountability than the graduate courses through the use of bi-weekly milestone reports as well as individual reports at the end of the project regarding the amount of work contributed by other team members. Since this is announced ahead of time, the students are more cognizant of the amount of time they put into their project and how their effort is perceived by the other students.

Other undergraduate course instructors have assigned group roles to assist training the students to work in groups. For a group of 3 the possible roles are a Team Leader, Team Administrator, and Team Critic. The Team Leader is responsible for planning meetings, directing discussion, and directing the research. The Team Administrator is responsible for ensuring that timetables are met, typing the reports, and handing in the completed project. The Team Critic is responsible for editing the reports, critiquing the logic and mathematics, and ensuring the reasonableness of the conclusions.

Other instructors have brought in professors from the psychology department to discuss group dynamics and problems that can develop. The instructors have then had the students use this information to designate the group tasks for each member as well as deal with problems within the group.

3. Problems of Conducting Group Project-Based Courses

3.1 Some Crucial Issues.

Major issues in conducting project-based courses, from both the instructor's and the student's side, are:

1. Selecting the project group size
2. Selecting the project groups
3. Selecting the project topics
4. Planning the project tasks, personnel and methods
5. Coordinating the tasks (monitoring the project)
6. Writing and testing program modules
7. Documentation and bibliography for tasks
8. Writing the project final report
9. Making presentations of the results to the class

3.2 Administering Group Projects and Monitoring the Issues

A group contains two or more students. However, a group of 5 or more students is almost certain to have problems because a term project is not large enough nor long enough in time to use all of the students effectively,

especially when they are not experienced at working in groups. Most of a project is done in the last few weeks of the semester. As the group size increases, the complexity of the planning and interaction increases. A group project is a valuable experience in teaching students to work together, but that is secondary to understanding all of the project content.

Our experience shows that for a moderately sized class of 15 to 30 a group size of 3 is near optimal for full student participation. Because the class size may not be divisible by 3, we may assign 2 or 4 to one or more groups. For large classes, the group size can be increased to 4. If a class is too large and the number of groups is too large with 4 students per group, then the instructor should forego the group project format because it will be impossible to do justice to its administration and grading of final reports.

We give the students notice early in the course that they must form into groups of a certain size. The group members could be assigned at random, which is perceived as fair. However, we have found that many students know someone with whom they prefer to work. Others may not have a preference or may not know any other students in the class. At midterm we ask and record the groups that have self-organized. This usually accounts for at least half of the class. For the remaining students we suggest groups based on the unannounced principle of how close they sit to others who are not yet assigned (a clustering effect). We allow any groups a period of several days to change and make adjustments. If a problem arises later as to a group composition, then the instructor must take quick action. We note that a husband and wife should not be together in a group with one or more other students.

The instructor, based on the level of the course, may present a list of project topics. It has worked well for us when the instructor also lets the groups either choose from the list or find their own topics. It occurs often that a group will be reminded of a desirable topic by one or more topics from the given list. Our instructors reserve the right to approve or modify a topic so it fits the desired level of difficulty. Once a group and the instructor have an agreement on a topic, the project can begin. It is a useful policy to permit the group to change the topic up to an announced deadline date because during the planning stage the group may discover something not anticipated.

The term group project format should be used only if the area is a specialty of the instructor: if the instructor is covering a course for the department but is not active in that area, then this format is not as effective because the topics must contain insight, be useful and applicable and be accompanied with usable references. We also suggest a bibliography and provide data files in many cases (data has been difficult to obtain in some cases, but the *World Wide Web* on the Internet is making this easier). The issues of planning, coordinating, programming and testing, documenting and referencing, writing the final report and making presentations are the responsibilities of the students.

We provide a handout sheet detailing the format of the final report. Some faculty limit the report to 10 letter size pages, not including the program listing that is to be attached. The report is to provide an introduction to the problem (the *what*), discuss how it is handled currently and provide references (the *who*), describe and justify the approach to be taken (the *how* and *why*), set up the computer runs and describe the data and any preprocessing of it, and make analyses and conclusions based on the results.

Some groups attack the project with vigor while others appear to flounder. It is crucial that the instructor monitor the groups to see what is happening. Certain ones will never get started without some pushing. We require that each group hand in a proposal of two pages or more that describes briefly what the problem is, how they will attack it, the data source and the expected results. This forces the groups to meet and write something down. Once this happens, the projects usually take on a life of their own and begin to develop.

3.3 Some Problems.

Some of the major problems that we have encountered are:

1. A group can not agree on a topic or else can not agree on how to implement it
2. A group keeps waiting to get started until it is too late to complete it by the deadline
3. A group has personnel problems and cannot work together
4. One or more members in a group is not contributing fairly

5. A group decides late in the process that it needs a different topic
6. One or more important references for a method cannot be found
7. A group cannot find the appropriate data
8. A group's program will not work properly
9. One or more students in a group are too advanced and experienced for the others

3.4 Dealing With Problems.

Conducting project-based courses can be a difficult experience for instructors who have not had experience with this approach. Unanticipated problems often arise. When a group has difficulty selecting or proceeding on a topic, we suggest extra topics and possibly suggest some methods for the students to examine. The instructor can also provide or suggest a source of data that could be used. The instructor, however, should not become involved in working on any project or solving any basic problems for the students because this would be unfair and would deny the students the complete experience of cooperative problem-solving. The project proposals usually get a group started, but the instructor should question the groups occasionally to see that they are progressing. Students can give a false impression that everything is moving along satisfactorily when it is not.

There is also the situation where the students are at an impasse but will not ask for help. To avoid this in graduate courses, some of the faculty use several class periods after midterm for consulting with the groups: no lecture is given and the instructor sits at a table in the classroom to consult with whomever walks into the room (attendance is not required and only consulting is done during these periods - students not receiving consulting are expected to be working on the projects). The reason that consulting may not work for undergraduates is that a university's accreditation requires a specific number of class meetings during a semester and this may not qualify as a class meeting. Group consulting has worked surprisingly well to get the groups over some of the initial problems. It does not tell the students how to do the project, but is an exchange of give-and-take technical discussions that let the students discover a way to overcome the problems. Discussions of group members with the instructor usually clear up misconceptions and fill in missing information. The students feel more confident and have their own good ideas for proceeding when they leave the consultation. Some groups come back again and certain groups come back at every opportunity.

Procrastination is more rare, but it can and does happen. Generally a student will complain that another group member has not done a certain task and that the group cannot make progress until that student completes the work. Such a group needs to be monitored very closely during the next few days to see if some intervention is needed. We have, on occasion, moved a student to another group or required the student to work alone. The latter denies that student the benefit of cooperative learning and usually means a lower grade. To help prevent such situations from arising, we announce at the beginning of the course that each student must sign and date a *disclaimer* to be attached to the project final report that tells what tasks that student did. When the students know that this is a requirement beforehand, it minimizes such problems.

The instructor should not undertake term group projects without the commitment to break up groups and make reassignments when needed and to have substantial discussions with students to get them started on the projects and keep going.

4. The Literature Covering Cooperative Learning

During our first experiences with project-based courses, we sought help from other professors and from the literature. It is amazing how much educational literature there is on the topic of group-based projects, typically called *cooperative learning* and how little it is known by those of us who practice teaching as a profession in other academic fields. It is with this in mind that we will review a sampling of the literature at this point.

Cooperative learning approaches seek to replace independent seat work with cooperative learning activities. Cooperative learning involves having the students work in small groups and allows students to work with some of their peers. This interaction allows them to receive information and feedback from their peers in addition to that

from the teacher and the curriculum materials. It is this additional source of learning that helps the students to internalize the material at hand.

The first topic the literature covers is termed *task structures*. There are three major categories of task structures: individual, cooperative, or competitive. Traditional independent seat work falls into the category of an individual task structure. Cooperative task structures require students to work cooperatively in order to meet the task requirements. Assignments that require the students to assist one another in learning, or work together to produce some sort of product, involve cooperative task structures. Competitive task structures require the students to compete (whether individually or in teams) in order to meet the goals. Contests, debates, and other games fall into this category.

The next topic in the literature is *individual or group goals*. Within a group, individuals may cooperate in working toward individual or group goals. When working towards individual goals the students in a group may discuss how to respond to questions, check each other's work, provide feedback to other ideas and approaches or work on what we would consider basic tutorials with one another. When they are working toward group goals, the members work together to produce a single product that results from pooled resources and shared labor.

The last major topic in the literature is *task specialization*. There are two types of task specialization. A project can be divided into sub-tasks where each sub-task is solved by different individuals within the group, or each member can work simultaneously on the same task (providing multiple solutions).

Summaries of cooperative learning are available from [2], [4], [5], and [7]. In [7] Slavin reviewed 41 studies and showed that 26 found significantly greater learning in classes using cooperative methods and only one found significantly more learning in the control group. All of the surveys pointed out that cooperative learning methods that ensure accountability of individual group members to their group mates produce higher achievement than methods in which it is possible for one or two students to do the work while the others take a more passive role (found to be true in our experience). The findings on task specialization were dependent on the subject matter, which we would expect. Finally, there is no evidence that group competition offers any advantage over cooperative learning methods.

There have also been several studies on the effects of group compositions toward learning. Some of these results can be found in [3], [8], [9], and [10]. When looking at these studies, three main conclusions emerge:

1. Whether or not students master the material depends not only on their entry level achievement, but the nature of the experience in their group. Students who learned to explain the material usually had very good experiences in their group because they could act as peer tutors to those in their group. Those that learned to ask questions when they did not understand also did well. But the performance of those who could do neither suffered as result of being in a group.

2. The quality of student interaction can be enhanced through training. Those groups that had training spent more time on tasks than those without training. Those who understood something about group dynamics and what their responsibilities were performed very well, and those that did not floundered.

3. Certain combinations of students work better than others. We all expect that, but the question then is which ones work better than others. The results can be summed up as follows.

- A heterogeneous group can have success, but it can also have some problems. The above average students tutor or train the low students, which is good for both categories, but the average students in the group sit around and don’t learn as much.

- In a homogeneous group the amount of success also varies. If the group is composed of all average members, then the group worked very well because they could all study and fill in each other's knowledge weaknesses during discussions. If the group is made

up of all above average students they have a tendency not to help one another since they assumed they all knew the material already. However, when the groups were made up of all lower level students, they all worked diligently but then became frustrated because no one in the group had enough knowledge to answer their questions.

Most of these studies showed that it was important to train the students how to work together in groups so that they could learn how to ask questions, how to answer questions, and thereby make their experience a profitable one.

5. Conclusions and Future Directions

One approach we learned from the literature that we are planning on trying is called the *Jigsaw Approach*. There are two major versions of this, which are covered in [1] and [6]. In the second version all students in the group start with the same material and a list of tasks. The group members then assign one another to tasks and the students become experts on their own sub-topic. Then members from different groups who are working on the same section meet together in what are called *expert groups* to discuss their task and how to attack it. Once the expert groups are finished the original groups get back together to solve the entire problem. The group grade along with the individual grade is used for each student. We are looking at bringing this into our *Data Structures* course where programming projects would help the students learn more about the material covered and the groups could attack larger and more meaningful projects than an individual would.

Probably the most interesting thing that we have seen from the use of group projects is the effects on outcomes other than achievement. Cooperative learning has promoted friendship and interaction among students, particularly those who differ in sex, race, or ethnicity. It has also had positive effects, and rarely negative, on affective outcomes such as self-esteem, academic self-confidence, attitude toward the class and toward the instructor, and various other measures of empathy.

Finally, cooperative learning methods have achieved impressive results. Slavin [7] is very excited about its possibilities, and so are we. It is important, however, not to view it as a wholesale replacement of whole-class instruction, recitation and independent seat work, but instead as a variation to it that is appropriately interspersed in the beginning courses while the core knowledge is being learned. It is also invaluable as a major tool for application of knowledge in advanced courses. Cooperative learning may be more feasible and valuable in certain classes than others, but all students need to be trained how to share, listen, and integrate the ideas of others, and handle disagreements. We believe that group project-based classes can effectively accomplish these goals.

Bibliography

[1] E. Aronson, N. Blaney, C. Stephan, J. Sikes, and M. Snapp, *The Jigsaw Classroom*. Sage, Beverly Hills, CA, 1978.

[2] D.W. Johnson, R.T. Johnson, and G. Maruyama. "Interdependence and interpersonal attraction among heterogeneous and homogeneous individuals: a theoretical formulation and a meta-analysis of the research." *Review of Educational Research*, 53:5-54, 1983.

[3] P.L. Peterson, L.C. Wilkinson, F. Spinelli, and S.R. Swing. "Merging the process-product and the sociolinguistic paradigms: Research on small-group processes." In P. Peterson, L. Wilkinson, and M. Hallinan (Eds.), *Instructional groups in the classroom: Organization and processes*. New York, 1984, Academic Press.

[4] S. Sharan. "Cooperative learning in small groups: recent methods and effects on achievement, attitudes, and ethnic relations." *Review of Educational Research*, 50:241-271, 1980.

[5] R.E. Slavin. "Cooperative learning." *Review of Educational Research*, 50:315-342, 1980.

[6] R.E. Slavin. *Using student team learning* (rev. ed.) Baltimore: Center for Social Organization of Schools, The Johns Hopkins University, 1980.

[7] R.E. Slavin. *Cooperative Learning*. Longman, New York, 1983.

[8] S.R. Swing and P.L. Peterson. "The relationship of student ability and small-group instruction to student achievement." *American Educational Research Journal*, 19:259-274, 1982

[9] N.M. Webb. "Student interaction and learning in small groups." *Review of Educational Research*, 52:421-445, 1982.

[10] N.M. Webb and L.K. Cullian. "Group interaction and achievement in small groups: stability over time." *American Educational Research Journal*, 20:411-423, 1983.

Frederick C. Harris, Jr.

Fred Harris is an Assistant Professor of Computer Science at the University of Nevada, Reno. He earned a B.S. in Mathematics and an M.S. in Educational Administration from Bob Jones University in Greenville, SC in 1986 and 1988 respectively. He earned an M.S. and a Ph.D. in Computer Science from Clemson University in 1991 and 1994 respectively. His areas of expertise are Parallel and Distributed Computing, Compilers, Operating Systems, and Virtual Reality with mining applications.

Carl G. Looney

Carl Looney is a Professor of Computer Science at the University of Nevada, Reno. He took the B.S. and M.S. in Mathematics from the University of Nevada in Reno in 1964 and 1968 respectively, and he earned a Ph.D. in Mathematics from the University of Iowa in 1972. He has worked in Industry at Hughes, Logicon and others. He has taught a very large array of courses in mathematics, electrical engineering and computer science over a span of 30 years. His current areas of expertise are Fuzzy Systems, Neural Networks, Image Processing, Computational Science, Pattern Recognition, Intelligent Systems, and Machine Learning.