Experience with Peer Learning in an Introductory Computer Science Course*

Craig E. Wills
David Finkel

Computer Science Department
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609

Running Head: Peer Learning

*This work was partially sponsored by a grant from the Worcester Polytechnic Institute Center for Curricular Innovation and Educational Development.

Abstract

A problem in teaching large introductory computer science courses is to overcome the impersonality of the large lecture class and to provide more personal attention to individual students. We used peer learning experiences to increase student interaction in a large introductory class and to instill in students the need to take responsibility for their learning and that of those around them. In support of this approach we have introduced the use of upper-level undergraduate students to help facilitate student group interaction and developed software to minimize the administrative overhead of handling many groups.

The paper describes our work and reports on results, which show that responsibility for learning was transferred from the instructor to the students. Peer learning was most accepted by students who reported having trouble with course while the approach had mixed support among students who did well in the course. Our overall assessment based on two offerings of the course with emphasis on group projects indicates it is beneficial for the students and faculty involved.

1 Introduction

A problem in teaching large introductory computer science courses is to overcome the impersonality of the large lecture class and to provide more personal attention to individual students. The traditional lecture format, while being an efficient medium for transmission of factual knowledge, is poorly suited to developing higher-level cognitive skills and increasing student motivation [14]. In addition, we believe it reduces the quality of the educational experience for the student, which in an introductory class for majors can have negative ramifications for future classes. It is easier for students to become isolated in the course without an obvious support mechanism, so students who do not understand material can "fall through the cracks."

To combat these problems, our work has involved changes in how we teach one of our introductory courses CS2005, Techniques of Programming. Our approach is to use peer learning experiences to instill in students the need to take responsibility for their learning and for the learning of those around them. Recent work has shown that educational quality for students and productivity for faculty can be enhanced through use of peer-learning environments where students do not just learn and faculty do not just teach [6, 8]. This course also seeks to add more exposure to a breadth of upper-level computer science topics through projects and closed labs.

The novel aspects of our work are to apply group learning in a large introductory computer science class setting and to expect more direct responsibility on the part of students for their learning. In support of these goals we have introduced the use of upper-level undergraduate students to help facilitate student group interaction. In addition, we have developed software to minimize the administrative overhead of handling many groups and for students to electronically record group learning activities.

This paper details our experience with two offerings of this course emphasizing peer learning. It expands on work presented in [15]. The paper begins with background on the school and course, proceeding to explain the approach used and related work. The middle portion of the paper describes specifics of the course structure, the group programming projects, assessment and software that was developed for use with the course. The paper goes on to present results from our work based on feedback from faculty and students involved with the course. It concludes with directions for future work and a summary of our findings.

2 Background

Worcester Polytechnic Institute is a private university with approximately 2800 undergraduate students and 200 computer science majors. The academic calendar at WPI consists of four 7-week terms with students typically taking three courses each term. Each course typically meets four times each week with an additional meeting for courses with laboratories.

CS2005 is the recommended gateway course for students wishing to take upper-level computer science courses and is taken after an introductory programming class. It is intended to accept students who have had a single term of computer programming and to develop skills in the design, implementation and analysis of basic data structures—similar to the traditional CS2 course [10]. The course currently uses the C programming language with programming and laboratory assignments done on workstations running a version of the Unix[1] Operating System. The audience for this course is diverse—including not only computer science majors, but large numbers of non-majors, particularly from departments such as Electrical and Computer Engineering, Math and Management. The course is taught twice yearly. Tables 1 and 2 show the diversity of student population for two offerings of the course. There were 128 students in the Fall offering and 130 students in the Spring offering, with a larger proportion of Computer Science majors in the Fall.

[1] Unix is a registered trademark of Unix Systems Laboratory

3 Approach

The approach we have taken is to motivate students to accept more responsibility for themselves through use of peer learning. Students take part in small group activities in class and in programming projects outside of class. This approach is appealing for its potential to draw students together and help them learn from each other. Rather than have students sit and absorb, they have a responsibility to themselves and to others for their learning. The goals in using this approach are:

- to increase the number of contacts among students in the learning process, which we believe will increase the accountability of students for their work and others,
- that student interaction will increase and become a larger part of the student learning process than traditional student/instructor, student/TA interaction, and
- to employ support mechanisms that will promote this course organization without overwhelming faculty resources.

Much previous work supports this approach. In computer science education, collaboration is typically introduced in upper-level software engineering courses, and experience has been reported at this level [5, 11]. In addition to our work, other work at the introductory level has been recently reported [12]. In the Biology Department at WPI, intensive collaboration has been introduced into the first-year biology sequence, with the experience creating many positive results [4]. The results from this experience, and the increasing importance of collaboration on our campus and in industry [2], has led us to introduce this technique at an early stage in our curriculum. The decision is partially confirmed by an evaluation of the computer science Major Qualifying Projects (MQPs), capstone senior projects, on our campus. One of the results of the evaluation was that projects done by a student team were generally better than those done by students working by themselves [7].

The issues are how to bring collaborative learning to the classroom without overwhelming faculty in administrative overhead, and so that students not only accept it, but "buy in" to it. We have taken a two-pronged approach: 1) use peer learning assistants (PLAs) to help facilitate the peer learning process; and 2) use software not only as part of teaching the course, but to lessen the overhead of administering such a course, thus making the course viable for an individual faculty member to teach. This approach affects all facets of the course, which are detailed in the following sections.

4 Course Structure

The weekly course structure is four 50-minute class meetings with all students and one 50-minute laboratory of about 20 students. The class meetings are led by a faculty member. The labs are led by graduate teaching assistants (TAs) for the course. Peer learning assistants are upper-level undergraduate students who facilitate group work among students and assist with exercises in lab.

4.1 In-Class Activities

The class meetings involve lecture material by the instructor mixed with informal active learning activities and discussion [1]. A key component of the in-class activities is organized group work, with at least one of these activities each non-exam week.

A common activity is a "group quiz" where students group themselves based on seating proximity. The quiz may require the students to consider different design possibilities or apply concepts learned in class to a different problem. While the students work and discuss, the faculty member circulates through the classroom answering questions and facilitating discussion. After the groups have worked on the exercise for most of the meeting time, the class spends time discussing problems encountered. The groups then have time to review their work.

Groups are responsible for making sure each member understands the work that is done. One member records the group membership and results, turning in the work at the end of the class. This approach allows the faculty member to easily review all, or a sampling, of the work submitted, as well as give credit to all students participating in the activity. Correctness of the results is not as important as promotion of the active learning process in this activity.

In-class group exercises allow students to participate in active learning activities in a comfortable environment while maintaining direct involvement with the instructor. We believe these type of exercises are important for students to understand the value of collaborative learning in a setting that involves the instructor. They also allow the faculty member to better understand where students are experiencing difficulties.

4.2 Laboratory

The course includes a weekly closed laboratory where each student has access to a workstation. The labs are led by a TA and assisted by the PLAs. Students learn about and obtain experience with software tools useful for debugging and building software projects. The labs are also used to study a particular topic, such as the comparative timing of different sorting procedures. Learning rather than evaluation is stressed in the labs so students are encouraged to work together and with the TA and PLAs.

Lab periods are also used for peer learning activities related to the group programming projects. Student groups have time to meet with each other and their PLA to discuss the current assignment and how their group is proceeding on the assignment.

A typical set of seven labs for the course is given below including topics and tools introduced in each.

Week 1 Students register themselves online in the class database. Students review Unix, editors and compilers if they are not familiar with their use.

Week 2 Students meet in their programming project groups for the first time. This lab is led by a PLA who leads students through team building exercises including selection of a group name and a code review.

Week 3 Students learn about the Unix tool *Make* [3] for building multiple software modules together.

Week 4 Students review problems with pointers by debugging a program using a debugger.

Week 5 Students learn about a number of scripting tools available in Unix and how these tools are combined in scripts. They learn that tasks can be accomplished by other means than creating a new program.

Week 6 Students experience the use of recursion through a simple recursive fill algorithm used in graphics. The shading for each "pixel" indicates the level of recursion. The program is implemented with X windows.

Week 7 Students compare the performance of different sorting algorithms on various sizes of input. Performance is plotted using the *gnuplot* plotting tool.

5 Group Programming Projects

The most significant peer learning activities are group projects done outside of class, which focus on larger design and programming problems. There are three such assignments during the course of the term following an initial assignment done on an individual basis that leads into the group work. The projects are done in pre-assigned groups of 4-5 students. Students in these groups share a common lab time with compatibility of out-of-class schedules. Each group is led by a PLA to help facilitate group activity. All students on the project receive the same grade unless individual assessments indicate otherwise.

The first project in the course takes one week and is designed to help students review material from their first programming course (in C or Pascal) and to help students "get up to speed" with the language used in the course. A typical assignment used for this task relates to mathematics and involves representation and manipulation of polynomials. Students create an array implementation and implement symbolic integration and differentiation, along with evaluation of the polynomials. This project begins as an individual assignment, but during this time the students are formed into groups. As the first group activity they perform a code review of each other's work to obtain peer feedback on what they have done and to learn from examining what others have done. They also get to know each other and choose a group name.

The second activity is given a little over a week into the course after the students have been exposed to stacks and queues. This project is given as a group assignment. The purpose of the project is for students to learn about data abstraction and to complete a project that is broken into pieces. One such assignment used in the course adds breadth to their knowledge of computer science by exploring the problem of job scheduling in operating systems. The students create a driver program to simulate the arrival of jobs into their system with specific start times and duration. In addition, each student creates routines to manage these jobs based on a different job scheduling policy. Each policy uses either a queue or stack to maintain the jobs.

Details of the assignment are covered in a peer group meeting outside of class and laboratory time. If possible, this meeting includes the PLA. Students discuss the overall organization of the project and how it can be divided for work among the group members. With the help of the PLA, students take responsibility for pieces of the project based on individual styles and interests.

After distributing work among individual group members, the students work on their portions and communicate (often electronically) with each other and their PLA on progress and problems. Group work is facilitated with a electronic mail alias for each group and a group-specific directory where group members can do their work and share it with the group. The students have access to each other, the PLA, teaching assistants and faculty for individual or group assistance. The students' work culminates by building the individual modules together into a working whole and testing them on various data. Students turn in their group contributions electronically along with an individual evaluation of each group member's contribution to the whole.

The next major project in one of the course offerings was to build an airline reservations system. This project uses more of the techniques the students are learning (linked lists, hashing and searching) and involves aspects of databases and heuristics for storing information and routing passengers. As before, the groups break up the project among the group members who must work together to build the entire system. The project is significant in scope and tests students' abilities to design, code and organize it.

The final group project involves trees and recursion. In this project, the students are given a data structure representation for a tree and create a set of routines, most of them recursive, to build and manipulate it.

5.1 Project Lifecycle

Figure 1 summarizes the lifecycle of a group project and shows the roles of the instructor, TA and PLA. The instructor of the course is the overall manager for the activities in the course. He or she creates the group assignment and is ultimately responsible for questions or problems. As shown, the design for the project is done by the entire group and broken into parts for individual work. Once individual pieces are coded and tested, the parts are built together and tested. The PLA helps facilitate group work and is available for technical questions during all phases of the project.

Once the project is complete, the students use a software tool *gturnin* to electronically turn in source code and tests of the completed group project. Group members also discuss how the project went; after which each member uses the software tool *turnin* to submit his/her assessment of how the group worked on the project along with individual assessments on how each member contributed to the project. The completed group projects are graded by the TAs.

6 Assessment

Group projects are graded as a group with this grade used as a base for individual grades. Two approaches were used for assessing individual contributions by members. The first was a weighting approach where members received a weight out of 100% for their contribution. For example, if there were four group members and each contributed equally then each received a weight of 25%. Individual grades were assigned by dividing the assigned weight by the expected weight for individual contribution and using this value as a multiplier on the group grade. This approach was more complex than necessary for translating weights into multipliers and caused problems by penalizing other members of the group for more work taken on by one member without group consent (all weights had to sum to 100).

The other approach was to use multipliers directly for individual group members. Individual multipliers were multiplied by group grades to yield individual grades. Thus a situation where all group members contributed equally was indicated by all multipliers of 1.0. Multipliers of greater than 1.0 were allowed, but generally ignored unless the group as whole had done poorly, but one or two individuals had done more work. In practice, 90% or so of the individual project grades were the same as the group project grade.

In terms of group accountability for individual learning, there were three exams given in the course emphasizing important material. Particular questions on the exams related directly to group activities and were noted as such. Each student group was then given a group score for the exam based on individual group members' performances on the group questions. Groups whose individual members did well received bonus points. This aspect of the course emphasized the accountability of students to not just complete the work, but for all group members to understand it.

7 Software

Not only did we use software to aid in teaching the course, but a key aspect to reducing the administrative overhead was the development of software to help in managing the course. Although this software did not relate directly to the student learning process, it was necessary to maintain records on grades and group activities. We had some prior experience with electronic submission and grading of projects. We built on this experience to make the administration of group work manageable for large classes. Without such software we believe the administration of such a large, collaborative-based class would be difficult. Software was also useful to promote student responsibility for recording learning activities [13]. The software we developed is compatible with Unix environments. Its major components are described as follows.

- The most important piece of software was for electronic registration. Rather than have students fill out paper forms that must be sorted manually, we have developed a program that allows all information about a student to be gathered online and stored in a database. Because of the nature of the course, students have computer accounts from which they execute the program the first day or two of class. We gather basic student information along with scheduling information and student background coming into the course.

- Once student information is stored another program is used to automatically group students based on priorities given to different parameters. We used the laboratory section and out-of-class schedules as the first two criteria for group programming projects.

- Students turned in projects as a group using *gturnin*. Each group member turns in the piece of the project for which s/he is responsible. The software gathers the pieces together for each the group. This approach increases accountability by recording which group members are turning in work. Students also submited individual assessments of the group performance using *turnin*.

- An electronic survey tool allows us to gather and correlate results. This tool is used by the students at the beginning and end of the course for assessment and evaluation.

8 Results

We have taught two offerings of the course using the described format with an emphasis on group projects. One course had 128 students and the other 130 students. There were approximately 30 project groups formed in each course with 4-5 students in each group. One course used seven PLAs and the other course five PLAs. In each course two TAs were used.

We have feedback from students taking these two courses along with feedback from students in a previous course where individual projects were used. This feedback is in the form of approximately 100 questions administered via the electronic survey tool at the beginning and end of the term (the survey under the old approach was only administered at the end of the course). Student grades were added to the results at the end of the course for correlation purposes. A sampling of these questions along with results are given in Appendix A.

Table 3 summarizes the class offerings from which we have feedback. One professor taught an instance of the course using the old and new format while another professor taught a different instance of the new format course. Student responses from courses with the first professor were volunteer with about half of the students participating in each case. When correlated with course performance these students tended to be better students. The other professor required students to provide response as a small part of the grade and had a much higher response level that better reflected the mix of students in the class. In addition, written comments were gathered from students, PLAs and TAs reflecting their perspectives on the course.

The ProfA-old and ProfA-new evaluations are good for comparison because only the approach differs between the two. The ProfB-new evaluations are good because they give a more complete evaluation of one course. Based on all of this feedback we can make a number of observations, many of which are supported by quantitative results, about our experiences with these courses. In the following, we break these observations down by topic. Overall, 80% of the students expressed satisfaction with the course as it was offered in the ProfB-new instance and 76% in the ProfA-new instance.

8.1 Course Structure

From the perspective of the faculty, the new course structure offers a number of advantages.

- Group projects allow larger and more realistic programming projects to be assigned. The student gains a new perspective when confronted with a problem for which s/he cannot conceptualize all of the details. More realistic projects also provide a better sense of accomplishment when completed.

- Students obtain real software design and implementation experience with group projects. This point cannot be underestimated as the students gain first-hand experience in building software modules together and see the problems that occur with poorly designed interfaces.

- Faculty and TAs are able to spend a more reasonable amount of time answering student questions. Typically in such a large programming course, student questions can overwhelm not only regular office hours, but also cause numerous interruptions throughout each day. With group projects, the students have a ready place to get help with questions from their peers.

- Teaching Assistants had many fewer projects to grade and hence were able to take more care in grading those projects.

Overall, the faculty member was able to concentrate on guiding the students in the learning process rather than being a person that students depended on for answering a large number of questions outside of class. Because the amount of activity required of TAs was also reduced, we were actually able to allot one fewer TA to the class than would normally be required in a traditional version of the class.

8.2 Student Interaction

Interaction among students and faculty/TAs was replaced by more interaction among students—hence increasing students' accountability for their own learning. There was a large amount of interaction among the students as witnessed by group meetings and electronic mail. Students often met before and after class to discuss the current project. The PLAs were actively involved with each group and reported weekly on group progress at a staff meeting.

Because students did not have a choice on how they were grouped, students interacted with others not in their circle of friends coming into the course. This approach opened students up to interaction with students from different majors and years. We noticed that as the term progressed and students became more comfortable working with each other that the student groups tended to wean themselves from the PLAs. By the end of the course each PLA was concentrating effort on only one or two groups which were having technical or interaction problems.

In trying to quantify student interaction, students were asked a number of questions concerning the source of their learning in the class. 44% of the students reported "learning a lot" from talking individually with faculty in the old format while 23% reported so in the new format. Similarly, 61% reported learning a lot from talking individually with the TAs in the old format and 39% in the new format. These numbers correlate with our experience of fewer student/faculty and student/TA contacts.

There was not a corresponding increase in students reporting to have learned a lot from each other. 74% reported doing so in the old format compared to 69% in the new format. This result is counter to what would be expected. A possible explanation is to consider these results are from the ProfA-old and ProfA-new instances of the course in which the students taking the survey tended to be better students. We know from the ProfB-new results that the best students are the least likely to benefit from other students in the course (59% for students receiving an "A" grade versus 77% for all other students). In addition, there was a slight increase from 38% to 41% of students indicating that other students were among the top two sources of learning in the course between the old and new format (see questions 58 and 59 in Appendix A).

Another expected outcome from enhanced student interaction was a better retention rate with fewer students dropping out of the course. Results concerning drop-out rates are shown in Table 4. The table shows the number of students in each course along with the percentage who received an NR grade and of those receiving an NR the percentage that dropped out[2]. The drop-out percentage indicates those students who did not take the final exam or hand in the last assignment and hence were considered to have dropped the course. The results indicate there was a sharp reduction in the drop-out rate, particularly between the ProfA-old and ProfA-new course given at similar times with similar student makeups.

8.3 Group Dynamics

In looking at the dynamics within each group, 79% of the students in ProfB-new and 88% of the students in ProfA-new reported completion of successful projects. These results correlate to our personal observation that overall the groups worked well together. We observed that group problems occurred in about one-fourth of the groups in each course. The primary problems were ill-prepared or unmotivated students who did not do their share of the work in a timely manner.

The most successful groups tended to have strong technical and interpersonal leadership. The "ideal" group had one or two people in the group who not only understood the material well, but also could help others in understanding it. This observation led to having students self-identify their technical competence as a criteria for forming groups in the ProfA-new instance of the course. Unfortunately, there was little disparity between the students' answers and this approach did not lead to significantly different groups than those formed in the first course. Work on how groups are formed needs to be pursued.

The strong interdependence among individual components of the projects caused group problems when less-prepared or unmotivated students did not complete their portion of the project. This was the principal group problem cited by students in their written comments. The fact that not all students were equally motivated or always had the necessary time to spend on the class was not realized before-

[2] Students receive one of four grades for a course: A, B, C or NR. The NR grade represents "no record" and indicates the student receives no credit for the course. An NR grade does not appear on a student's transcript and encompasses the D, F and W grade given at other schools.

hand. However, it is these students who cause the most group problems. Despite these problems we did note an overall reduction in the drop-out rate indicating the group projects help to identify problem students and provide the personal assistance needed to retain them in the course.

One area in which we did not find any significant results was in terms of gender differences. The courses themselves contained predominantly male students (only about 10% female) and none of the results showed any significant differences based on gender.

8.4 Student Evaluation of Groups

We asked the students a number of questions about their experience of working in groups. The principal question asked concerned their agreement or disagreement with the statement that they "liked working in groups to solve the course assignments." 58% of the respondents agreed with this statement in the ProfB-new version and 72% in the ProfA-new version. In the ProfA-old version, using individual assignments, 48% indicated they would have preferred using group projects. Similarly, we asked students if they would have preferred working on assignments by themselves. 42% agreed with this statement in the ProfB-new version, 41% in the ProfA-new version and 82% liked the individual assignments in ProfA-old. Overall these results indicate a stronger student sentiment to retain groups rather than use individual assignments.

We also correlated the question about liking groupwork with other questions to see what kind of students liked this approach. Table 5 shows the correlation between liking groups and how well the student reported his/her group worked together in the ProfB-new course. The correlation coefficient of these two questions is 0.49 indicating a high level of correspondence as would be expected.

Differences also occurred between students perceptions of groups based on course grade. Table 6 shows the correlation of these two results in the ProfB-new course. As shown the group approach was most popular among "B" students, then "C" and lastly "A." These results, along with written comments received from students, indicate that students who have succeeded in a traditional classroom approach are comfortable with that approach and are less interested in change. The top students will generally succeed in any environment, but they are less comfortable in a team environment because they are not only responsible for their own work, but also in helping others. Some student comments felt they were having to work harder for the same grade, although from our perspective this means that responsibility for learning is being transferred from the instructor to the students. The "A" students in the ProfA-new course were relatively more enthusiastic about groups in the survey results, but similar written comments were received.

Table 7 shows the correlation between liking groups and the year of the student for the ProfB-new course. One significant result here is the clear preference for groups by Seniors. This result is not surprising at WPI because the students must complete two significant projects that typically involve groups of students. These projects are done in the Junior and Senior year so Seniors are comfortable working in groups. The other significant result is the non-preference for groups by other students, which in this course were talented high school students who were able to take the course. These students have obviously excelled in traditional courses (they also did well in our course) to take such an advanced course and are less open and have less maturity to enjoy working with others.

To further investigate the correlation of liking group work with different types of students, we examined how students, who reported difficulty in the course, liked groups. Table 8 shows these results for ProfB-new with a correlation coefficient of 0.26 indicating some correlation between these two questions. Similar results were found in the ProfA-new data. Again these results indicate that group work is not as useful for the best students, but was beneficial for students having some difficulty. These results also indicate that this style of teaching is applicable at not only places with generally good students, such as WPI, but may be even more applicable at schools with a broader range of students.

A final comparison with liking groups was done based on the learning style of students. Table 9 shows the correlation from ProfB-new for students liking groups and liking problems with only one definite solution. As shown there is a reverse correlation (correlation coefficient of -0.19) between these two questions indicating students preferring a definite solution are less likely to like groups.

8.5 Student Learning

Another key component of our research was to investigate the impact of group work on student learning. To this end we asked students two questions concerning how the use of group work impacted students' perception of their learning in the course. In the ProfB-new course 4% felt they learned much less because of working with others in group projects rather than working on individual projects, 27% felt they learned less, 33% about the same, 34% more and 12% much more. In the ProfA-new course 8% felt they learned much less, 19% less, 35% about the same, 28% more and 10% much more. In both courses students perceive that they learned more using the group projects. This effect was particularly pronounced among the "B" and "C" grade students in ProfB-new, although not so in the ProfA-new course.

For comparison, we asked students the same question except substituting "learning" with "course grade" so that students indicated how they felt their course grade would be because of working with others in group projects rather than working on individual projects. In both courses the students felt their course grades would be a little worse because of group projects (e.g. 7% much lower, 27% lower, 40% about the same, 23% higher and 3% much higher in ProfB-new). Surprisingly, in ProfB-new, this effect was most pronounced in students receiving a grade of "A," indicating the best students felt the group work was penalizing them even though they received the top grade. In ProfA-new, the "A" students responding felt their grade was slightly higher.

One other question concerning peer learning involved the use of informal group activities in class where students worked together on a problem. In ProfB-new, where these activities were used the most, 69% felt these activities were good learning experiences.

9 Future Directions

The results of this work provide a number of opportunities for future work. A field ripe for more work is how to better select the group membership. Our results indicate that groups of students working well together lead to a more satisfying educational experience. We observed that technical and interpersonal leadership is needed for group success. More work towards identifying such leaders at the beginning of the class could be used in forming the groups. Alternately students could be grouped by year or major, but we feel that diversity is important. Some students suggested choosing their own groups, but in such a large class with students not knowing each other well this approach does not appear easy to implement nor equitable to all students.

A key problem faced by groups is how to deal with weaker or less motivated students whose failure to meet group deadlines potentially impacts the success of the entire group. The strong interdependence of group programming projects can cause tension in the group when this situation occurs. One potential solution is allow students to be "fired" from groups, but this may have the negative effect of removing a student from a group who needs help from others the most.

In terms of faculty guidance, there needs to be more emphasis on the correct decomposition of large problems and the testing of individual components. Particularly for the more complex group projects, groups had problems who had a poor design in terms of data abstraction or did testing only after integrating all of the modules.

As an ongoing process, students will be examined on their achievement and preparedness for successive classes. We would also like to expand the use of peer learning to other courses in our introductory computer science curriculum. More complete assessment procedures, using measures such as concept maps [9], could be employed to better assess student understanding of concepts and their relationships.

10 Conclusion

Although there has been much group work at the upper levels of the computer science curriculum, we believe that the large class sizes often deter its use in introductory courses. However, these courses are where this technique can be most useful. We have explored the use of peer learning in such a course. Through the use of peer learning assistants and the creation of software to minimize the administrative overhead we have been able to concentrate on making this course a quality educational experience. Our assessment based on two offerings of the course indicates it is beneficial for the students and faculty involved.

From the faculty standpoint, the use of groups allows larger and more realistic projects to be assigned and provides a (sometimes painful for the students) demonstration of the need for good software design and testing.

Both our observations and comments from the students indicate that responsibility for learning was transferred from the instructor and TAs to the students themselves. Less time was spent by faculty and TAs answering questions, while more time was spent by students interacting. The amount of faculty and TA "interruptions" outside of regular office hours was sharply reduced. Student interaction was facilitated through the use of electronic mail and the sharing of a common directory for group work.

In terms of the impact of this approach on different types of students, we observed the most acceptance by students whose group worked well together and by students who had trouble with the course. The latter observation indicates that peer learning might be even more appropriate at schools with a larger variation in student backgrounds and abilities.

Results among the better students in a class were mixed. There are indications in the results that some students who succeed in traditional environment are less interested in the new environment because they must spend more time helping others. However, it is this increased student/student interaction that was a goal of this work. The results show success, but there is still work to be done in helping all students to buy in to this approach.

The biggest problems developed in groups with weak or less motivated students. The latter group of students was particularly difficult as they were less inclined to seek the help of other group members, but at the same time they could not be counted upon to do their in a timely manner. These situations caused the most group dynamics problems. The formation and dynamics of groups is one problem that needs to be studied more.

In summary, we view the introduction of peer learning in a large introductory computer science course a success. We will continue to explore ways to use it for creating a better learning experience for faculty and students.

References

[1] Chris C. Bonwell and James A. Eison. Active learning (creating excitement in the classroom). *ASHE-ERIC Higher Education Report 91-1*, 1991.

[2] Rich Cohen and Warren Keuffel. Pull together! *Computer Language*, 8(8):36–44, August 1991.

[3] Stuart I. Feldman. Make—a program for maintaining computer programs. *Software—Practice and Experience*, 9(3):255–265, March 1979.

[4] Leonard Goodwin, Judith E. Miller, and Ronald D. Cheetham. Teaching freshmen to think—does active learning work. *BioScience*, 41(10):719–722, November 1991.

[5] Brad Hartfield, Terry Winograd, and John Bennett. Learning HCI design: Mentoring project groups in a course on human-computer interaction. *SIGCSE Bulletin*, 24(1):246–251, March 1992.

[6] David W. Johnson, Roger T. Johnson, and Karl A. Smith. Cooperative learning (increasing college faculty instructional productivity). *ASHE-ERIC Higher Education Report 91-1*, 1991.

[7] Robert E. Kinicki and Craig E. Wills. 1993 computer science department MQP review. Technical Report WPI-CS-TR-93-5, Worcester Polytechnic Institute, August 1993.

[8] Richard J. Light. The Harvard assessment seminars: Explorations with students and faculty about teaching, learning and student life, First Report 1990.

[9] Joseph D. Novak and D. Bob Gowin. New strategies for evaluation: Concept mapping. In *Learning How to Learn*, chapter 5. Cambridge University Press, 1984.

[10] ACM Curriculum Committee on Computer Science. Curriculum 78: Recommendations for the undergraduate program in computer science. *Communications of the ACM*, 22(3):147–166, March 1979.

[11] Hassan Pournaghshband. The students' problems in courses with team projects. *SIGCSE Bulletin*, 21(1):37–41, February 1989.

[12] Roberta Evans Sabin and Edward P. Sabin. Collaborative learning in an introductory computer science course. *SIGCSE Bulletin*, 26(1):304–308, March 1994.

[13] Steven L. Tanimoto. Beyond the naivety of grades: Educational record keeping for the twenty-first century. Technical Report 92-07-09, Department of Computer Science and Engineering, University of Washington, October 1992.

[14] E. J. Thomas and C. F.Fink. The effects of group size. *Psychological Bulletin*, 60:371–385, 1963.

[15] Craig E. Wills, David Finkel, Michael A. Gennert, and Matthew O. Ward. Peer learning in an introductory computer science course. *SIGCSE Bulletin*, 26(1):309–313, March 1994.

A Sample of Survey Questions

The questions shown were administered in each course (questions 1-47 in the beginning of the course and 48-94 at the end). The answers shown are for 91 respondents in the ProfB-new course. All answers required a numeric response (the surveys were done online) and where appropriate an "average" answer is shown based upon the numbers.

```
4. I prefer problems that have only one definite solution
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1=23  2=46  3=20  4= 2  ave: 2.0

24. I am:  Male--1      Female--2
1=82  2= 9

25. I am:  Computer Science major--1   not a Computer Science. major--2
1=40  2=51

26. I am:  Freshman--1   Sophomore--2   Junior--3   Senior--4   Other--5
1=34  2=22  3=18  4=12  5= 5

48. I learned a lot in this course.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 2  2= 9  3=28  4=52  ave: 3.4

49. I learned a lot from working and discussing topics with other students taking
the class.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 5  2=21  3=47  4=18  ave: 2.9

53. I learned a lot from attending lectures.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 2  2=11  3=43  4=35  ave: 3.2

54. I learned a lot from attending labs.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 4  2=24  3=49  4=14  ave: 2.8

55. I learned a lot from talking individually with the teaching assistant.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 4  2=38  3=33  4=16  ave: 2.7
```

56. I learned a lot from talking individually with our peer learning assistant.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1=17 2=40 3=26 4= 8 ave: 2.3

57. I learned a lot from talking individually with the professor.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 6 2=41 3=35 4= 9 ave: 2.5

58. I learned the most from 1=other students, 2=lecture, 3=lab, 4=textbook, 5=my own work.
1=14 2=31 3= 1 4= 7 5=38

59. I learned the second most from 1=other students, 2=lecture, 3=lab, 4=textbook, 5=my own work.
1=20 2=29 3= 3 4=15 5=24

62. I found some difficulty with this course
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1=14 2=16 3=49 4=12 ave: 2.6

69. I like the idea of working on projects with other students
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 5 2=25 3=42 4=19 ave: 2.8

77. I liked working in groups to solve the course assignments.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1=12 2=26 3=37 4=16 ave: 2.6

78. I would have preferred working on the course assignments by myself.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1=12 2=41 3=24 4=14 ave: 2.4

79. I learned a lot in this course about how to work as a team member.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 2 2=25 3=52 4=12 ave: 2.8

85. My project group worked well together.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1=17 2=24 3=36 4=14 ave: 2.5

86. Overall, my project group completed successful projects.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)

1= 3 2=16 3=41 4=31 ave: 3.1

87. The PLA for our group was helpful.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 9 2=19 3=49 4=14 ave: 2.7

88. I feel I learned (1=much less, 2=less, 3=about the same, 4=more, 5=much more) because of working with others in group projects rather than working on individual projects.
1= 4 2=15 3=30 4=31 5=11 ave: 3.3

89. I feel my course grade will be (1=much lower, 2=lower, 3=about the same, 4=higher, 5=much higher) because of working with others in group projects rather than working on individual projects.
1= 6 2=25 3=36 4=21 5= 3 ave: 2.9

90. The in class group quizzes are good learning experiences.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 3 2=25 3=52 4=11 ave: 2.8

93. I am satisfied with the course as it was offered.
(1=strongly disagree, 2=disagree, 3=agree, 4=strongly agree)
1= 0 2=18 3=48 4=25 ave: 3.1

94. Grade actually received in the course (added later) (1=A, 2=B, 3=C, 4=NR)
1=27 2=35 3=27 4= 2 ave: 2.0

Table 1: Percentage of Students by Year and Major (Fall offering)

Major	Fr	So	Jr	Sr	Other	Total
CS Major	32	5	4	1	2	44
non-CS Major	4	22	17	11	3	56
Total	36	27	21	11	4	100

Table 2: Percentage of Students by Year and Major (Spring Offering)

Major	Fr	So	Jr	Sr	Other	Total
CS Major	15	6	5	1	4	30
non-CS Major	16	19	10	21	5	70
Total	31	24	15	22	9	100

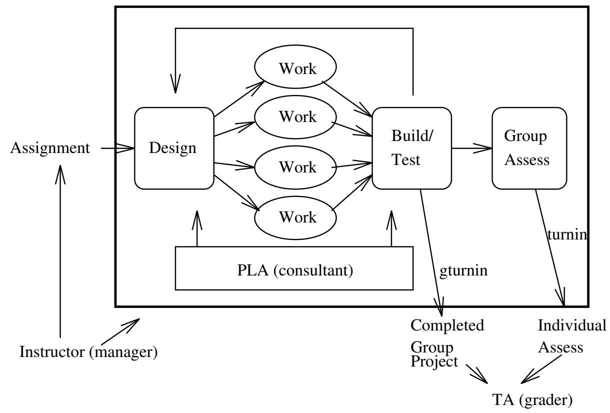

Figure 1: Project Lifecycle for a Group Project

Table 3: Class Offerings and Student Response

		Student Response		
Offering	When Offered	Type	Rate	Mix
ProfA-old	Spring	volunteer	50%	tends towards better students
ProfB-new	Fall	required	90%	comparable to entire class
ProfA-new	Spring	volunteer	50%	tends towards better students

Table 4: Drop-Out Rates

Offering	Students	Overall NR pct.	NR via Drop-out pct.
ProfA-old	150	17	16
ProfB-new	128	14	12
ProfA-new	130	10	7

Table 5: Project Group Worked Well Together Correlated with Liking Group Work (pct.)

	Worked Well Together	
	No	Yes
Disliked Groups	31	11
Liked Groups	14	44

Table 6: Course Grade Correlated with Liking Group Work (pct.)

	Course Grade			
	A	B	C	NR
Disliked Groups	48	37	41	50
Liked Groups	52	63	59	50

Table 7: Year Correlated with Liking Group Work (pct.)

	Year				
	Fr	So	Jr	Sr	Other
Disliked Groups	41	41	50	17	83
Liked Groups	59	59	50	83	20

Table 8: Found Some Difficulty with the Course Correlated with Liking Group Work (pct.)

	Found Course Difficult	
	No	Yes
Dislike Groups	20	22
Like Groups	13	45

Table 9: Prefer Problems with Only One Definite Solution Correlated with Liking Group Work (pct.)

	Prefer One Definite Solution	
	No	Yes
Dislike Groups	27	14
Like Groups	48	10