AC 2007-2924: BENEFITS AND STRUGGLES OF USING LARGE TEAM PROJECTS IN CAPSTONE COURSES

Troy Harding, Kansas State University-Salina

Benefits and Struggles of Using Large Team Projects
in Capstone Courses

Abstract

Computer System Technology graduates should have strong conceptual and practical knowledge as well as being able to work collaboratively at all levels of software development. One way to bring this all together is by using a capstone course involving a major semester-long team project.

This paper will describe and compare the projects used in our capstone courses over the last two years. It will also present and discuss the impact of having multiple teams all working on their own team project as opposed to having the whole class work together on the same large project. The paper will discuss scheduling and implementation problems, evaluation procedures, peer collaboration in problem solving and conflict resolution, students' attitude toward using projects as an instructional tool, and instructor reflections on the process. Suggestions for improvement from both an instructor and a student perspective will be included.

Introduction

The benefits of employing collaboration in the classroom are well known[1]. In addition, advisory board members, prospective employers, and industrial partners have been expressing the desire to hire computer systems technology graduates that are well rounded in all aspects of our profession. Graduates should possess strong conceptual and practical knowledge as well as be able to work collaboratively at all levels of software development: from problem solving to design, from development to implementation and maintenance. Such an obvious request is not easily implemented.

In an attempt to implement this request, two methods of forming student teams were explored in the capstone course for web development technology. The capstone course in which the projects were completed is for advanced students and focuses solely on students designing and implementing a large project using development teams.

Collaborative Structure

In one section of the capstone course twelve students were grouped into three teams of four students each. The class as a whole was assigned a project with the idea that different aspects of the project would be divided among the different groups. This will be referred to as the "cooperative-teams" approach.

In another section, the students were also divided into teams of four. However this time each team completed their own version of an assigned project. For the most part the teams worked independently of each other. The only time the teams worked together was to collaborate on such things as conducting user surveys and client interviews in the early stages of the project planning. This will be referred to as the "independent-teams" approach.

The web development capstone courses used a modified version of a Systems Development Life Cycle as described by Jonathan Lazar in his book, *User-Centered Web Development Lifecycle*[2]. The Lazar Model defines seven stages in project development:

1. Define the mission and the user population of your web site
2. Collect user requirements
3. Create conceptual design
4. Create physical design
5. Usability testing
6. Implement and market
7. Evaluate and improve

In both the cooperative-teams approach and the independent-teams approach the teams in the class cooperated on the first two stages listed above. There were several practical reasons as to why this level of cooperation was necessary even for the independent-teams approach. The main reason has to do with efficiency. Real projects with real clients are used whenever possible. The benefits of doing so have been documented[3,4,5]. However, in order to reduce the amount of time the client and potential users spend in answering questions it was necessary for the student teams to work together in the gathering of the client and user requirements. Using the independent-teams approach, once the requirements were gathered each team determined on their own how the information would be used in the design of the website. In contrast, the section using the cooperative-teams approach split the responsibilities of creating one website among the three teams.

The following table summarizes the collaborative structure of the two approaches:

Cooperative-teams Approach	Independent-teams Approach
• 3 teams of 4 students • All teams worked together to collect client and user requirements • One project produced for entire class • Project tasks divided between teams • Teams met regularly and cooperated with each other during design stages	• 3 teams of 4 students • All teams worked together to collect client and user requirements • Each team produced their own version of the project • Teams worked independently of each other during design stages

Students Perspective

At the end of the semester students were asked to fill out a survey and write a reflection about their experiences.

The survey used a Likert type scale where students would rate various components of performance quality as Excellent (4), Very Good (3), Satisfactory (2), Marginal (1), Terrible (0).

Following are the average responses for each statement:

Statement	Cooperative-Teams Class Average	Independent-Teams Class Average
1. Group dynamics within your team:	2.9	3.1
2. Group dynamics between teams in the class:	2.1	2.8
3. Quality of work produced by your team:	2.7	3.0
4. Quality of work produced by the class as a whole:	2.5	3.1

Because there are only 12 students in each of these classes the sample size is quite small. However, the survey responses along with student comments indicate that there are some significant differences in the student perspective. As might be expected, questions 1 and 2 on the survey indicated a much lower score for group dynamics between teams in the cooperative-teams class as compared to group dynamics within the team. Several student comments from the cooperative-teams class indicate a certain level of frustration with working with other teams. Following are some examples of comments about the other teams:

"They are obstructive and greedy with the class project and assignments."

"Team 1 has dominated this project. They have made myself and my team feel isolated from the project."

"I feel like the other teams as a whole did not want to share responsibilities."

"We could be better at communicating/keeping each other aware of what the other is doing."

It should also be pointed out that there were several students who rated group dynamics between teams as "Very Good", but the majority of the students rated this area as "Marginal" or "Satisfactory". In addition, there did not seem to be any correlation between question 2 answers and which team the student was a member of. There were members of each team that rated this high and members of each team that rated it low.

Another trend that survey questions 3 and 4 seem to suggest is that the cooperative-teams class was less satisfied with the quality of work produced as compared to the independent-teams opinions about their own work.

Faculty Perspective

The instructor teaching the capstone courses experienced challenges and rewards. The biggest problems were (1) managing group dynamics and (2) keeping the project on track so that it could be completed during a single semester.

The instructor tried very hard to anticipate and plan for problems in group dynamics. The groups were chosen by the instructor, who tried to match good students with bad ones by selecting for each group a good leader, a good programmer and a student with good verbal and organization

skills. After selecting groups in this manner, students known to be poor were distributed among them. Grading criteria included a certain percentage devoted to individual work and peer evaluations so that unproductive students did not automatically receive the same grade as others in the group.

Several procedures were employed to manage and handle the groups. First, the groups were required to submit regular progress reports to the instructor. After reading the reports, the instructor would meet with each group, discuss how the project was progressing and give the group suggestions on how to deal with issues of group dynamics. Second, the groups made regular presentations to the client. Third, each member of a group underwent a peer review in which his or her performance was rated by fellow group members. This peer review was factored into the student's grades. Finally, each member wrote a reflection on the project and what they learned during the course.

In spite of these arrangements, groups from both classes suffered the typical problem of unproductive group members. That is, students who chronically missed meetings, rarely completed their assignments, or never contributed ideas.

Regarding keeping the project on track, it was a challenge to make sure that the project did not continue to grow during the semester. Feature expansion is common in Information Technology projects, so the instructor had to be diligent to make sure the students limited project features to what could be done during a sixteen-week semester.

The cooperative-team class seemed to particularly struggle with staying on track. They spent three semester weeks longer than the independent-teams class did on the design of their projects. The instructor eventually had to set a deadline and force them to the next stage of development. This extra time can be attributed to the group dynamics and number of ideas being generated. Even though different parts of the project were distributed among the teams in the class, individuals were really good about brainstorming new ideas for the project as a whole. Invariably these new features or design elements would impact all the components in the system one way or another. This added onto the extra communication overhead of having all twelve students in one class trying to discuss aspects of a single project. The hope was that distributing components of the project among the three teams would have reduced the communication overhead more than it did.

There were several advantages to the independent-teams approach as compared to the cooperative-teams approach. One of the biggest is that the client had a choice of final projects from which to choose and had the benefit of seeing various fresh ideas and different approaches to the problems. The competition between groups was healthy and caused some groups to excel, although at times the competition distracted from the goals of the project.

Overall, the quality of the projects in the independent-teams approach exceeded that of the cooperative-teams' project. On the surface, projects from both classes seemed to be of very good quality. However, closer inspection of the cooperative-teams' project revealed cobbled together code, inconsistent naming, and generally a system more difficult to maintain and upgrade. All of which can be attributed to the communication challenges of this approach.

At first glance it might appear that the independent-teams approach would be the preferred method over the cooperative-teams approach. There is an additional question that needs to be factored in, "What was learned during the course?" If the main goal of the class is to produce a high quality product for the client then the independent-teams approach seems to be the favored method. Generally speaking, however, the student project reflections from cooperative-teams class were much more insightful and full of ideas for improvement than the reflections from the independent-teams students. One could argue that the cooperative-teams class learned more from their struggles and review of their final project.

Conclusions

Using cooperative-teams to do one large class project has the potential to teach students a great deal about teamwork and group dynamics. However, if this isn't managed well the quality of the final project can suffer. This approach might be best suited for large projects where the client is not highly dependent on the final project.

The independent-team approach with each team doing their own version of a project tends to produce higher quality projects and more choices for the client. The tradeoff is that students generally aren't exposed to the same degree of communication overhead and project management challenges. In general, it seems to be a good approach for exposing students to teamwork with the added benefit that there is a greater chance that at least one of the projects will meet the client's needs.

Perhaps with a more structured problem definition and clear distribution of team responsibilities the cooperative-team approach could produce a high quality product while still exposing students to a distributed project environment. It is of this author's opinion that if a suitable project can be found, the cooperative-team approach should not be shied away from. The chance that students may not be completely successful in producing a high quality project is offset by what they will learn going through the process.

Bibliography

1. Johnson, David W., Johnson, Roger T., Smith, Karl A., *Active Learning: Cooperation in the College Classroom*, Interactive Book Company (1991).
2. Lazar, Jonathan, *User-Centered Web Development*, Jones and Bartlett Publishers (2001).
3. MacLean, L.M., Pencek, T., Benefits and difficulties in use of real projects for advanced database applications, Proceedings of the 17th Annual Conference of the International Academy for Information Management, 5 pages.
4. Chamillard, A.T., Braun, K.A. 2002, The software engineering capstone: Structure and tradeoffs, Proceedings of the SIGCSE 2002, 227-231.
5. Harding, Troy, Kinsler, Les, Leite, Pedro, Mertz, Thomas, Business and Education Cooperation through Classroom Projects, 2004 ASEE Annual Conference Proceedings.