Managing and Assessing Software Engineering Group Projects

Donald R. Schwartz
Millsaps College

While the benefits of assigning group projects are numerous, managing and evaluating them can sometimes become daunting tasks. Among the biggest challenges include determining individual grades for group members and attempting to reflect the "real world" by mixing up the makeup of each group and the tasks to be completed. After trying various approaches and styles for more than a decade, I think that I have developed a useful pedagogy for managing group projects, one that attempts to allow each student to work on different parts of different projects, with a different group of students, over the course of the semester. This approach also gives the students several opportunities to evaluate their own individual work, the work of their teammates, the work of their own group, and the work of another group. These evaluations play a major role in helping me to assign grades to the group as a whole, as well as to each individual member of the group.

Getting Started

On the first day of class, along with the normal paperwork (syllabus, etc.), students complete a self-evaluation questionnaire regarding their current level of familiarity with a wide variety of computer science topics, along with the Computer Science courses they have completed and the courses they are currently taking. This is very useful since students come into the course at the sophomore, junior and senior level, with a wide range of abilities. (The only actual prerequisite for the course is our CS2 course.) The questionnaire can be given orally and students simply note the item number and give their responses. (See Figure 1.)

Most items on the survey are very self-explanatory. The topics listed under item 8 are purposely very general. They also include such "gimmes" as WWW and word processing, so that every student can put a high number for at least a few topics. This is especially helpful for the sophomores, for whom this may be their first upper-level course. (I added item 11 when I became Chair of the department, since I'm often contacted by companies in our region looking for part-time workers.) The self-assessment numbers prove very useful when determining which students actually implement which projects. This is discussed later in the paper.

On the first day of class, we also discuss the types of projects we might tackle during the semester. I briefly describe past projects such as a video poker game, a C-to-Pascal program converter, a Pascal-to-C converter, as well as a variety of websites for online class registration, student advising, a dorm-room scheduler, admissions, and various board and/or card games. Students must come to the second class meeting with suggestions for three semester projects they might like to complete. Students must submit written descriptions of the projects, along with explanations of why the projects would be appropriate for an upper-level course. Students briefly describe the projects to the rest of the class. As a class, we narrow down the possibilities to a manageable number. (Of course, I reserve the right to include my suggested projects into this final list. Throughout the year, I solicit project ideas from across campus and include those I

feel are appropriate.) On the third class day, I have the students come in with their top 3 or 4 "favorite" projects from the list. I tally the results and let the students know what projects we'll tackle. (I still reserve the right to include my "favorite" project to this final list, but have actually discovered that the students usually pick good projects, so I rarely have to "force" my choice into the final mix.)

1. Name and LoginID
2. List all CS courses taken (and the grade you earned)
3. List all CS courses you are taking this semester
4. List all programming languages you have used
5. List the CS topics you like the most
6. How many people in class can you name?
7. What software applications are you comfortable with?
8. Rate your expertise (0-10)
    1. Programming
    2. GUI
    3. OS
    4. Database
    5. Software Engineering
    6. Networking
    7. Architecture/Hardware
    8. WWW
    9. Hypertext
    10. Algorithms
    11. Oral communication
    12. Written communication
    13. Word Processing
    14. Spreadsheets
    15. Scheduling Software
9. What is your favorite part about groupwork?
10. What is your least favorite part about groupwork?
11. Do you have a job (either on-campus or off)? If yes, describe it. If no, do you want one?

Figure 1 – Self-Assessment Form

During the first week, I have the students enter their schedules into the Outlook calendar tool, including classes, meetings and work schedules. (Meetings and work schedules are to be listed as "tentative" events.) I also enter my information into Outlook. As the various groups determine their meeting times, I have them "invite" me to each meeting via the "Plan a Meeting" feature of Outlook. This enters their meetings onto my calendar (as a tentative event). I tell them that I will try to attend at least one meeting per group, as long as they schedule meetings that "fit" into my (very busy) schedule. I remind students that the best time for me to try to meet with them is during my office hours, so they should try to plan around those. I do not let students schedule meetings in my office, but do try to sneak out and touch base with students during office hours. (I consider meeting with student groups an appropriate use of a portion of my allocated office hours.) If I can't sneak away, I let the students know they can come by my office to let me know what's going on. Students often like to meet late in the evening – I let them know on the first day that I will rarely, if ever, be able to meet with them "after dark". I do try to attend at least a couple of meetings per project during the semester. I find that the students are really appreciative when I meet with them, even if it's just for a few minutes. They are able to ask a few questions and update me on their progress. It also helps maintain a good level of rapport, which is usually reflected in student evaluations!

Project Management

In my Software Engineering course (as well as in most other Software Engineering courses), we have 6 major milestones [1]:

- **Requirements**, which describe what the customer wants,
- **Specifications**, which constitute the "contract" for the product to be produced,
- **Preliminary Design**, in which the task is broken up into modules, and the interaction among the modules is defined,
- **Detailed Design**, in which pseudocode for each module is developed,
- **Implementation**, in which the actual coding is done, and
- **Testing**, during which the finished product is evaluated for accuracy and completion.

Each of these has an associated document which the student groups must submit.

There are many possible approaches to use when it comes to assigning groups for the Requirements Document.[2,3] I happen to leave it to the luck of the draw. I come into class with a deck of cards. I have the students determine the size of the teams for each project, based on the description of each project. We then assign card values to each project and have students pick a card to determine their project. The students then get together and schedule their first out-of-class meeting to get started on the project.

As a milestone is completed, each group has a few minutes to present its work to the class and to describe what issues in the next phase the next team might find the most challenging. Each group suggests a staffing level for the next phase. After every group has completed its presentation, we make sure the suggested staffing levels "work" (that is, every student will be assigned to a project) and then pull out the cards. However, the cards aren't always "right". I try to make sure that no more than one student remains on any given project for more than 2 consecutive milestones. I also try to make sure that each student works on each project at least once, so that every student gets exposed to every project. As the semester goes on, I also try to make sure that no two students have worked together on the same team more than once or twice, so that students work with new partners on new projects throughout the term. By the time we come to the detailed design phase, I often have to put aside the cards and start assigning students so that the new-project-new-teammates paradigm is satisfied. About the only time I "predetermine" staffing is during the implementation phase. I try to make sure that implementation teams contain relatively equal distributions of students. (Every group should have the same number of sophomores, the same number of seniors, the same number of database "experts", web "experts", etc.) Of course, this is not always easy (or possible), but I find using the results of the survey given on the first day of class (Figure 1) to be very useful here. Even if we have, say, a project that is heavily database-driven, another that is web-based, and a third that requires a lot of graphics, I do not "load up" the database project with just students who have had those courses. I think that it is very important that students pick up new skills while completing the project and remind the students that this is one of my most important goals. Just because a student hasn't had a database course does not mean she doesn't have to learn enough SQL to contribute to the implementation. Since this is an upper-level course, this is not an unreasonable goal.

Project Assessment

At each milestone, the students must submit two copies of the actual document, one to hand in to me to be graded and the other (without names or other identifying information) to be given to the group working on the next phase of the project. Each group must also submit a written evaluation of the previous document (which we call the "incoming document"). This evaluation must describe what was good/useful about the incoming document, what was wrong or missing, what extra work needed to be done by the current team in order to complete the current document, and what made their job easier and/or harder. Each team understands that their current document must be completely finished regardless of the state of the incoming document. Their evaluation of the incoming document is used by me when assigning a grade to the incoming document. In addition, the current group must indicate a suggested staffing level for the next phase of the project (the number of students, the needed area(s) of expertise (or familiarity), etc.). We use these suggestions in order to determine the number of students per group for the next phase.

On the days on which documents are due, students hand in their documents and evaluations. Each student then completes a form in which each describes the experience of completing the assignment and the group dynamics that came into play. (See Figure 2.) This form is completed privately – group members are not allowed to sit near each other while this is done. This allows students to be completely candid with their responses. I remind the students that this is their opportunity to give me official feedback about how the group work was completed. Since we employ group work in *every* Computer Science course, the students have been exposed to this evaluation form since their CS1 course. I tell them that this is their "official" feedback forum – I do not consider verbal feedback to be as useful, since I would have no documentation for things said in my office. (Of course, students are welcome and encouraged to come to my office to discuss group issues as they arise. If an issue is extremely volatile, I will consider stepping in to correct the situation. However, I always try to allow the students to try to handle issues as a group—a very important skill to be honed. I do remind the students to be sure to include these issues in their evaluations.)

The first item on the feedback form allows me to track the actual amount of time spent working as a group. Hopefully, the students within a particular group have similar estimates for this item! If numbers do not coincide, this raises a flag as to the amount of time a particular student met with the group.

The second item is probably the hardest item for the students to complete, especially early in the semester. I remind students that if everyone did the same amount of work with the same level of success, then all students should get a 50. However, if some students contributed more than others, those students should receive a rating above 50. Of course, this means that those rating points must be taken away from other student(s). (In the past, I used the word "score" instead of "rating" and an average of 75 instead of 50. I found that students equated the word "score" with "grade" and often became upset that they were giving everyone what they interpreted to be a 'C' on the project if all students did the same amount of work. No matter how much I insisted that these were *not* grades and that 75 did *not* mean a 'C', the students clung to this perception. I considered phrasing this question in terms of a monetary bonus given to the group, to be distributed among the members as indicated by the student, but found that this approach made it

harder for me to easily and quickly evaluate the results since different groups often had a different number of members. I have found that using the word "rating" and an average of 50 works fine – it's far enough removed from the concept of "grade" and 'C' to be palatable to the students.) I insist that students include themselves when assigning ratings, since the ability to assess their own work is very important. Of course, my biggest initial concern was that a student, regardless of how much actual work she had done, would give herself the highest rating. However, I have found students are actually very honest when comparing their work and the work of their teammates.

Your Name:

Group responsibilities and amount of time spent meeting outside of class:

List **all** group members (including yourself). Give **each** member a rating.
Ratings MUST average 50!

Who (if anyone) acted as group leader?

Was the workload evenly distributed among group members? If not, describe.

Who did the most work?

Who did the least work?

Were the group dynamics good? (Did y'all work well together?) Describe.

Did the group achieve its goals and responsibilities? Describe.

Describe what YOU PERSONALLY did during this phase. What did you have to teach yourself to complete this phase?

Figure 2 – Group Evaluation

The third item indicates who, if anyone, took the lead on the project. I do not assign a group leader. I have found that a leader will emerge if one is needed. If a majority of the teammates name the same person as the leader, then that student gets "credit" for being the leader. It is not unusual for students within a group to indicate that no one was the leader.

Items 4-6 allow the students to identify those members who did not pull their weight and those members who picked up the slack. The first of these items often reflects how the workload was initially distributed. Sometimes the initial estimates of how much work is required for a particular subtask are wrong, but this might not be discovered early enough to re-distribute the

load. Items 5 and 6 allow each student to explicitly identify students who did more or less work than others. Ideally, these items correspond with the ratings given to each student in item 2.

Item 7 has the students describe the group dynamics. It is usually the case that the members of the group got along well; however, if issues arose, this is where the students reflect and respond.

Item 8 allows the students to explicitly identify which portions, if any, of the assignment were not completed. An explanation of why some things were not finished is also given. Explanations are sometimes as silly as "We ran out of time" or as serious as "Jimmy and Susie never showed up for any meetings". (The most extreme explanation I've encountered happened when a student disappeared for over a week. The student's body was recovered in a nearby river about a week later.) This item is especially important for the "next" group, so that the missing part(s) can be handled. (I have each group describe what, if anything, is incomplete when they present their projects to the class.) It also makes the testing phase much more pleasant when the testers know which portions of the implementation are missing or might crash.

I added item 9 so that students could identify the specific portions of the task for which they were personally responsible. It allows students to discuss the work they completed and take credit/blame for the good/bad parts of the project. It is also my goal that students should have to teach themselves new skills when completing an assignment. This usually occurs during the design and implementation phases.

When grading each milestone document, I grade the work based on what was required for the particular phase and assign a group grade. I also incorporate the "next" group's feedback about the previous document into the group grade. This means that grades aren't actually assigned until the end of the subsequent phase. This sometimes bothers a few students early in the semester, but they come to realize that their feedback is actually being used. In fact, each group also gets a grade on its evaluation of the incoming document. I base this grade on the content and thoroughness of the evaluation, along with how closely it matches my own evaluation of the document. I then use the group evaluation forms to determine the "adjustment score" for each student. I have used this score two different ways – either to adjust each student's grade on each milestone report, or as a separate grade to be incorporated into the student's overall project grade. I usually use the score to adjust the grade for each milestone, so that students can readily see how their participation and productivity actually affect their grades. Those who slacked off see their grades tumble, while those who had to carry an extra load see their grades rise because of it. During semesters when most (or all) students seem to be doing their fair share of work, I use the score as an additional "milestone" score.

Students are also graded on how accurate their group evaluations were. If one student claimed everyone did his fair share and gave everyone a rating of 50, while everyone else in the group assigned markedly different ratings, something is indeed amiss.

Calculating evaluation scores is both an art and a science. Item 1 (on Figure 2) carries very little weight – I mainly check to be sure it is complete and corresponds with other group members' estimates. Item 2 carries a lot of weight – on each student's form, I write down all of the ratings each received from their teammates. I also write down the rating the student gave herself, but I

circle it, noting whether it agreed with the ratings given by the other students in the group. Students whose average rating is above 50 earn points, while those below lose points. If a student is identified by a majority of their teammates as the group leader (in item 3), that student's grade gets bumped up. Items 4-6 are used in a similar manner. Students whose work was identified in items 7-9 as adversely affecting the project are further penalized. I then go through the form, tally up the values and use the following scale:

+++ ++ + √+ √ √- - -- ---

The two marks at either end are used only in extreme cases. A √ means that a student did her fair share of the project and therefore needs no adjustment from the group grade. The further to the left from the √ indicates the amount of additional work/effort the student put into the project; the further to the right indicates how little a student did. (For example, a ++ usually means that everyone in the group gave the student a high rating, identified that student as the group leader, and indicated that the student did far more work than others in the group.) Each + or – usually indicates a letter grade adjustment. √+ and √- are used for half-letter-grade adjustments. Of course, the amount of adjustment is up to the instructor – as long as it's consistent, any range can be used. If a student's grade is repeatedly lowered because of evaluations, a teacher-student conference is warranted. Usually, that student will come by on her own to ask why her grades are so low. (If the student does not come by on her own, I'll let her know that she needs to. It's not fair to future groups to have a student consistently under perform.) I do not show the evaluations to the student, but I do use them to convey to the student that her work has been identified as below average by both me and her peers. Often, the student will offer up some sort of explanation, but this usually provides an ideal opportunity to help the student prioritize those issues competing for her time.

Conclusion

I have found that this methodology has proven to be very useful when managing and assessing group projects, not only in Software Engineering, but in all of my classes. Granted, it does require an extra commitment of time on the part of the instructor, especially during the first few attempts to use this style. However, the benefits to be reaped from this approach, as well as the variety of useful experiences the students receive, make it worth the extra effort.

[1] Schach, Stephen R., Classical and Object-Oriented Software Engineering, 6th Edition, WCB/McGraw-Hill, 2005.
[2] Baker, Diane F. and Connie M. Campbell, "When Is There Strength in Numbers? A Study of Undergraduate Task Groups", College Teaching Vol. 53 Num 1, Winter 2005, pp. 14-18.
[3] Steinberg, Daniel and Daniel Palmer, Extreme Software Engineering, A Hands-on Approach, Pearson/Prentice-Hall, 2004.

DONALD R. SCHWARTZ
Don Schwartz is Associate Professor and Chair of the Department of Computer Science at Millsaps College in Jackson, Mississippi. He earned his PhD at the University of Louisiana at Lafayette. He teaches a wide variety of courses and his current research interests encompass the areas of Database, Software Engineering and User-Interface Design. He can be reached via email at *schwadr@millsaps.edu*.