

Individual's Evaluation in Courses With Team Projects

Hassan Pournaghshband
Department of Software Engineering
Southern Polytechnic State University

Shahriar Movafaghi
Information Technology Department
Southern New Hampshire University

Abstract

Team projects have become an essential part of many software engineering courses. These projects mirror the typical work situation a student will find after graduation. Since the difficulties and frustrations of working in a team cannot be appreciated unless experienced first hand, team work has been a required part of courses of this nature. Different types of problems regarding team projects have been examined by other researchers. In this paper we discuss the instructor's problems regarding evaluation of teams and also individual members of a team, and present an approach that has proved to be a "fair" approach for the team as well as individual team members.

1. Introduction

In today's expansive technology world, most software engineering projects are developed by a group of software specialists, and not very many projects are completed by individuals.

The fact that most software engineers work in teams makes it necessary for a software engineering student to experience the team work and familiarize him/herself with potential problems. This team project experience could help students learn how to successfully complete a team work, and thus more prepare them for entering business or industry after their graduation. However, team projects present difficulties for both students and the instructor. As far as the instructor is concerned, the job becomes complicated for them mostly because it is difficult to evaluate the individual students in each team. Although this problem has been examined by Chrisman [1] and others, but there are still issues that need to be further investigated to better evaluate students and assign a fair grade to each individual member of the team. Regarding students, team work becomes more complicated for them because they encounter some serious unexpected problems during the project life cycle

that may not have to deal with them otherwise (i.e., if working individually.) The students' problems in courses with team projects have been carefully examined and discussed by Pournaghshband [2]. The author discusses nine of the most serious problems encountered by his students in two different courses requiring team projects. These problems in order of their severity are: poor communications, poor leadership, failure to compromise, procrastination problem, integration testing problem, lack of cooperation, lack of confidence, conflicts in students' schedule, and members' personal problems. In Section 2, we will address some of these problems and show how they could add to the instructor's problems of individual evaluation of students in courses with team projects. In Section 3, we examine the instructor's issues regarding evaluation and grading individual members in each team and propose an approach which has been proved to be "fair" based on the outcome in our team project-oriented courses. Conclusions are given in Section 4.

2. Problems

In this section we address some of the instructor's very common problems in courses with team projects. More specifically, we examine problems regarding team selection and project selection. Although these problems have been discussed in literature by others, an overview of them in this section is necessary for our main discussion in this paper (given in Section 3.) Team selection is probably the first problem the instructor of a course with team projects has to face during the first or second week of the semester. Since software development is a human

process, choosing right personnel for the team, and assigning roles and responsibilities to each individual member of the team requires good management skills [3.] In an academic environment, this burden is placed on the instructor's shoulder. He/she has to carefully choose appropriate students for each team and possibly decide about their responsibilities (some instructors allow the teams themselves to decide about members' responsibilities.) Team selection is done effectively if it is based on the instructor's knowledge of individuals rather than a random selection, or allowing the students to select their teams. To do this, the instructor can distribute a short questionnaire to the students asking about their technical background and experience as well as their personal feelings and preferences regarding team work [2.] This way, students with better technical background (and possibly work experience) are divided among all teams, and also those students with similar class and work schedules are placed in the same team. Project selection can also be a tough and tricky task for the instructor. The project must be "appropriate" for the class and for the students. It should not only satisfy the course and curriculum requirements, but also completing it in one semester (i.e., analysis, design, and implementation) should be feasible and its goals attainable. In addition, to better serve the purpose, the project should closely resemble a "real-life" project. Besides team selection and project selection there are other important issues that must be considered regarding the instructor's problems in courses with team projects. A careful and "fair" evaluation and grading the team as well as evaluating individual members of each team is the

main subject of this paper. In Section 3, we address this issue and present an approach that has proved to be effective in this regard.

3. Team's assessment and individual's evaluation

A very common and straight approach practiced by many instructors is basically to evaluate each team's final product and assign an appropriate grade for the team. In this approach, all team members receive the same grade (the grade for the team.) Obviously, employing this approach, in many cases, does not lead to a fair evaluation of individual team members. The grade for the member contributed the most to the project would be the same as the member with the least amount of work and effort. In real world, one of the most pleasant task, and somewhat tricky, of project managers is to reward team members when the project is progressing as planned [4.] It is tricky because while the project success is attributable to the team effort, it may owe its success mostly to one (or more) individual's extra work and effort [4.] Likewise, in an academic environment, the instructor not only should acknowledge the team effort, but he/she should also recognize those individual students contributing the most. Among several different approaches of this nature that we have tried in our classes, one has convinced us the most to be "fair" to our students. Our students in different courses, have been pleased with this approach, and have acknowledged the fairness of evaluation and grading using this method. In this approach, we first evaluate the team work as usual and assign a grade, say X ,

accordingly. Next, we ask each individual team member to evaluate all members of the team (excluding him/herself) and assign a grade, say G_i , to member M_i provided the average of all grades assigned equals to X (i.e., the team's grade.) This provision is vital and prevents possible grade inflation by any team. In the table given in Figure 1, grade $G_{p,q}$ is the grade assigned to member M_q by member M_p of the team. We furnish each member with a table like this, and ask them to confidentially complete it. Each individual completes his/her corresponding row of the table. Column "Total" is filled by the instructor before distributing to students. The value of all entries in that column is the same and is equal to $(X \times (n - 1))$, that is, the team's grade times one less the number of team members. Evaluators are also required to support their assessment as follows:

1. For each member whose assigned grade is greater than X (i.e., the team's grade), a brief explanation of why he/she deserves a higher grade (e.g., extra work and contribution.)
2. For each member whose assigned grade is less than X , a brief explanation of why he/she should be penalized and receive a lower grade (e.g., missed group meetings, did not meet deadlines, etc.)
3. For each member whose assigned grade is equal to X , no explanation is required.

This requirement helps students to think carefully before evaluating and grading their team members, and it could also discourage bias grading (specially if there has been many disagreements (and possibly arguments) during group meetings.

	Member M ₁	Member M ₂	Member M ₃	Member M _n	Total
Member M ₁	//////////	G _{1,2}	G _{1,3}		G _{1,n}	
Member M ₂	G _{2,1}	//////////	G _{2,3}		G _{2,n}	
Member M ₃	G _{3,1}	G _{3,2}	//////////		G _{3,n}	
.						
.						
.						
Member M _n	G _{n,1}	G _{n,2}	G _{n,3}		//////////	

Figure 1. Evaluation Table

Once we collect the evaluation table from all members, then we compute and assign individual's final grade of the project as indicated in Figure 2. Grade for member M_i is G_i and is computed as $G_i = T_i / (n - 1)$, where $T_i = (G_{1,i} + G_{2,i} + \dots + G_{i,i})$. It is important to note that the instructor has to share this process of evaluating team members with his/her students very early in the semester. This could well encourage all members to not only be more willing to contribute throughout the semester but also gives them a good incentive to better cooperate with the team.

As an example of the above process suppose we have a team of six members, and suppose the grade for the team's project is 27 (out of 30). Each member has to evaluate other members (not him/herself) and assign a grade (not greater than 30) to each member provided the average of all grades assigned by any evaluator equals to 27.

In other words, the sum of all those five grades should be equal to 135 (27x 5 = 135, that is, the team's grade times number of members evaluated). Note that in table of Figure 3 the value in column "Total" is 135 for all six members. In this example Bill has the highest grade (28.5) and Jack has the lowest (25.5). Please note that the average of all above grades is still 27 (i.e., the team's grade for the project). $Avg = (26.5 + 27.5 + 26.5 + 28.5 + 27.5 + 25.5) / 6 = 27$. Let's look at the Jack's evaluation of his teammates. He is suggesting grade of 25 (which is less than 27) for John, the grade of 28 (greater than 27) for Bob and Mary, and grade of 27 (equal to 27) for Bill and Nancy. In this case, Jack is required to give his supportive statement for Bob and Mary, and also his comments of why John does not deserve the team's grade of 27.

	Member M ₁	Member M ₂	Member M ₃	Member M _n	Total
Member M ₁	//////////	G _{1,2}	G _{1,3}		G _{1,n}	
Member M ₂	G _{2,1}	//////////	G _{2,3}		G _{2,n}	
Member M ₃	G _{3,1}	G _{3,2}	//////////		G _{3,n}	
.						
.						
.						
Member M _n	G _{n,1}	G _{n,2}	G _{n,3}		//////////	
Total	T ₁	T ₂	T ₃		T _n	
Grade	G ₁	G ₂	G ₃		G _n	

Figure 2. Table used by the instructor

	John	Bob	Mary	Bill	Nancy	Jack	Total
John	//////////	28	25	29	27	26	135
Bob	26	//////////	26	30	28	25	135
Mary	27	28	//////////	27	29	24	135
Bill	27	27	27	//////////	27	27	135
Nancy	28	26	26	29	//////////	26	135
Jack	25	28	28	27	27	//////////	//////////
*****	*****	*****	*****	*****	*****	*****	*****
Total	133	137	132	142	138	128	*****
Grade	26.5	27.5	26.5	28.5	27.5	25.5	*****

Figure 3. An Example

4. Conclusions

In courses with team projects, it is very important to not only acknowledge the team effort, but also recognize and properly reward individual team members for their extra work and effort. In this paper we examined problems regarding team and individual evaluations, and presented an approach

which has been proved to be fair to our students.

References

- [1] Carol Chrisman and Barbara Beccue, "Evaluating students in systems group projects," ACM-SIGCSE, 1987.

[2] Hassan Pournaghshband, "The Students' Problems in Courses With Team Projects," ACM-SIGCSE, 1990.

[3] Timothy Lethbridge and Robert Laganieri, "Object-Oriented

Software Engineering" : Practical Software Development Using UML and JAVA, McGraw-Hill, 2005.

[4] Frank Tsui, "Managing Software Projects," Jones and Bartlett, 200