1988

Software Engineering Team Project Group Member Evaluations: Some Empirical Results

Herbert E. Dunsmore
Purdue University, dunsmore@cs.purdue.edu

David P. Moffett

Steven T. Ward

Report Number:
88-740

SOFTWARE ENGINEERING TEAM PROJECT GROUP MEMBER EVALUATIONS: SOME EMPIRICAL RESULTS

Hubert E. Dunsmore
David P. Moffett
Steven T. Ward

CSD-TR-740
January 1988

Software Engineering Team Project
Group Member Evaluations:
Some Empirical Results

Hubert E. Dunsmore
David P. Moffett
Steven T. Ward
Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907

January, 1988

Abstract

Students in a software engineering course at Purdue University in the Fall of 1987 participated in team projects during and after which they completed Group Member Evaluation Forms. These evaluations revealed that: (1) project work load distribution is an area for improvement, (2) our students liked working in a team software development environment, (3) they were happy with their 4-5 person group sizes, (4) contribution critiques were very consistent (including self- critiques) and seemed to correlate well with perceived amount of work done, and (5) there was no consensus about how to handle parasites (i.e., students who do not contribute adequately to the team effort).

1. Introduction

At Purdue University in the Department of Computer Sciences we have been teaching CS 404 each semester since the spring of 1986. This course is a senior-level introduction to software engineering intended for the computer science student with a strong background in programming, algorithms, data structures, operating systems, etc. Lectures address what modern software engineering involves, including the techniques and principles generally employed. The most important facet of the software engineering course is the laboratory component in which students participate in a full-semester team software development project.

Early each semester the students organize themselves into teams of 4-5 people. We allow them to select their own team composition rather than assigning them to teams. Our justification is that this can make it easier for them in terms of group meeting schedules and they can work with students with whom they are already familiar. Thus, the teams begin with some degree of cohesiveness. Our only caveat is that each team must have some minimal C programming experience - the language used in coding the project.

Each team works on the same project. At the beginning the teams are given specifications. The semester is spent in their developing requirements, prototypes, design, test plans, and code, and culminates with an in-class demonstration of their software product. In parallel, the instructor (the first author of this paper) covers in class each week material relevant to the phase of the project on which the teams are currently working.

The project approach has been a resounding success. Our students are able to take the "theory" from the lectures and put a good deal of it into practice on the project. In exit questionnaires they repeatedly state that the CS 404 project was one of the most (if not THE most) useful experiences in the entirety of their computer science education. These claims are particularly gratifying in light of the fact that this course is known as a "killer" course - requiring 20-30 hours per week (or more).

In our continuing effort to fine-tune the course, we have recently begun to consider some aspects of the project that could be handled differently and made more equitable to the students. Among several others, the following questions have arisen:

- Is the project work load distributed equitably among the team members?
- Is the team project experience viewed as a positive one or merely a necessary evil of this course?
- Is the 4-5 person team size a good choice for a one semester project?
- What can be done about the inevitable problem of parasites (i.e., students who do not contribute adequately to the project but who benefit from the work of more industrious colleagues)? Surprisingly, in four semesters teaching this course this problem has occurred only rarely. But, when it does occur it represents a significant time investment for lab instructors and a tremendous distraction for the software development team affected.

2. Group Member Evaluations

In the Fall of 1987, 23 students were enrolled in CS 404. Their team project was to construct a relational database software package. They organized themselves into 5 software development teams - 3 comprised of 5 students each and 2 comprised of 4 students each. Later in this paper we refer to these teams via the letters A-F. These letters were not used during the semester, have been assigned randomly, and have no mnemonic association with the groups at all. During the course of the semester, teams were required to select a group leader, and to submit requirements documents, user's manuals, prototypes, design documents, test plans, and code.

We decided to collect some information concerning the questions listed above along with a good deal of other information that would be helpful to us in fine-tuning the CS 404 software engineering course. The lab instructors (the second and third

authors of this paper) created an evaluation form to be administered at the approximate half-way point in team software development and also to be administered at the end of the semester. It contained questions related to group dynamics, group skills, and group communication. It was also a vehicle whereby each team member could independently critique the contributions of the other team members. The forms were completed in secrecy and submitted to the lab instructors. In each of the following sections we report on some of the data collected via these evaluation forms.

3. Equitable Project Work Load Distribution

Comments in previous semesters had alerted us that one of the points of contention in group dynamics and relations seemed to be the feeling that some students were doing more than their share of the team's work. We made no effort to help the Fall 1987 teams in this regard and suspected that this might be viewed as a problem area by our students. On the end-of-semester evaluation form we asked the question "How equitably did your group distribute the work load?". There were five allowable responses: excellent, good, reasonable, poor, and terrible. The results below show the distribution of the 23 responses:

excellent	1
good	6
reasonable	14
poor	1
terrible	1

Clearly there is a consensus here. Our students did not find the project work load distribution to be a critical problem, but they were not completely satisfied with it either.

4. The Team Project Experience

On the mid-semester evaluation form we asked "How well do you like working in a group/team environment?" and on the end-of-semester evaluation form we asked "How well did you like working in a group/team environment?". Again, the five allowable responses were excellent, good, reasonable, poor, and terrible:

	midterm	end
excellent	12	9
good	8	7
reasonable	3	5
poor	0	2
terrible	0	0
	3.39	3.00

The average responses 3.39 and 3.00 are computed by assigning each response a numerical value as follows: excellent=4, good=3, reasonable=2, poor=1, and terrible=0. Clearly, at mid-semester there was a good-to-excellent feeling about working on a software development team. As we expected after the final few weeks' crush to get the project done, the end-of-semester responses were not as positive as before. But, interestingly, the slippage was not severe, and the students departed the course still rating their software development team experience as good.

On the end-of-semester evaluation form we also found that individual feelings of satisfaction with the group experience correlated well with other team member's responses:

team	excellent	good	reasonable	poor	terrible	average
A	3	1	1	0	0	3.40
B	2	3	0	0	0	3.40
C	1	1	2	0	0	2.75
D	0	1	2	2	0	1.80
E	3	1	0	0	0	3.75

Notice, for example, the apparent agreement at the good-to-excellent plateau by E, A, and B team members. Members of C were a little less positive about their experience and the D team declared themselves in the reasonable-to-poor range. As an aside, this order is roughly congruent with our evaluation of the quality of the work done and products produced by these groups. We suspect that group problems and poor results are reflected in lower "How well did you like working in a group/team environment?" responses.

Keeping in mind that all groups were 4 or 5 persons large, we also asked them "What do you think would be an ideal group size?". Their mid-semester and end-of-semester responses:

team size	midterm	end
8	1	0
7	1	0
6	2	3
5	9	11
4	6	6
3	4	3
	4.7	4.6

The weighted average responses are 4.7 and 4.6 as shown. There was almost no change of opinion from the mid-semester to end of the semester on this and quite a lot of agreement (17/23 at the end) that 4-5 people is ideal for the type of project we assigned.

5. The Parasite Problem

On the end-of-semester form, we asked each student to critique ALL members of his team (including himself). The summary question for each member was "How satisfied were you with this person's contributions to the group?". Again, the five allowable responses were excellent, good, reasonable, poor, and terrible.

One thing which interested us was whether these critiques would be consistent. Examining the 84 critiques of OTHER team members (i.e., excluding self-critiques), we found almost total consistency. Every one of the 23 students' contributions were rated by their peers consistently using the same (e.g., all excellent) or adjoining categories (e.g., some good - some excellent). There was not one single instance of disjoint categories (e.g., some excellent - some poor). Thus, team member critiques appear to be consistent enough if we need to use them to identify problem cases.

Consider two interesting items: This semester's parasite, a low-contributing member of a 4-person team, received 1 poor and 2 terribles. Another semi-parasite member of a 5-person team received 1 reasonable, 2 poors, and 1 terrible.

Surprisingly, we also found that students were mostly honest in their own self-critique. Categorizing the 23 responses to "How satisfied were you with this person's contributions to the group?" when directed at themselves we found:

15	Rankings same as rest of the group
4	Rankings more critical than rest of the group
2	Rankings more complimentary than rest of the group
2	No Self-Rankings

Note that 19 of the 23 ranked their contributions no better (4 were obviously being modest) than their peers ranked them. Only 2 of the 23 (the parasite and the semi-parasite!) reported more contribution than their peers reported for them. Interestingly, 2 people declined to critique themselves.

In addition to the "How satisfied were you with this person's contributions to the group?" question, we also asked for open-ended comments on ALL team members via the two questions "What did this person contribute best to the group? (Please Explain)" and "How could this person have been more helpful to the group? (Please Explain)". Examining these responses we discovered two interesting results: (1) a strong correlation between the contribution rating (discussed above) and comments about the "amount of work done" by this person, and (2) surprisingly good agreement between comments about oneself and comments offered about them by the rest of the group (even for the parasitic people).

We have selected below a number of examples to illustrate this point:

Student: "... could have done more work ..."

Other group members: 2 of 3 said "... could have done more work ..."

Student: "... was a hard worker ..."
Other group members: 2 of 3 said "... worked very hard ..."

Student: "... should have communicated better ..."
Other group members: all 3 said "... communications problems ..."

Student: "... had good technical ability ..."
Other group members: 3 of 4 said "... good technical expertise ..."

Student: "... did not spend as much time as I should have"
Other group members: all 4 said "... needed to spend more time ..."

Student: "... did not work enough on the project ..."
Other group members: all 3 said "... should have worked more ..."

Student: "... needed to do more work ..."
Other group members: 3 of 4 said "... could have done more work ..."

Student: "... was too grouchy at times ..."
Other group members: all 3 said "... had bad attitude problem ..."

6. Punishing the Parasites?

We have always employed the policy in CS 404 that all members of the team receive identical grades for the project, regardless of individual contributions. Our justification has been that this reflects the way industrial software development teams are viewed and rewarded. (We realize, of course, that there are some cases in industry when this is not true.) But, since the project accounts for 60% of the student's final grade (the other 40% comes from exams), we are concerned that this is academically unfair. Specifically, this policy does not adequately discourage parasitism. One solution would be to adjust grades (down!) for those team members who do not carry their fair share of the load. Obviously, the results discussed above appear to indicate that we have a somewhat-objective vehicle for identifying parasites. But, we were concerned whether our students thought that such grade adjustment was a good idea.

On both the mid-semester and end-of-semester evaluation forms we asked "Do you think that it would be fair if we assigned points based on the content of these forms?":

	midterm	end
Yes	12	7
No	7	8
Maybe	3	8
No Response	1	0

Clearly there was some shift of opinion as the semester progressed. Over half (12/23) students voted Yes on the mid-semester form, but that number dwindled to about 30% (7/23) by the end of semester.

Furthermore, when we asked how a student's grade should be affected, the responses to the question "If we assigned points based on the contents of these evaluations, what percentage of a student's total grade should be affected by such a process?" showed a consistent indication to adjust grades only by 1-10%:

percentage adjustment	midterm	end
>15	0	0
11-15	2	1
6-10	9	8
1-5	8	7
0	1	5
No Response	3	2

Finally, responses to the open-ended question "What do you think we should do if a student is given poor rankings by most or all of his or her group?" were categorized as follows:

8	Adjust his/her grade
8	Talk to him/her
2	Let the group handle it
5	No Response

Thus, only 34% (8/23) seemed in favor of punishing parasites via grade adjustment. Interestingly only 29% (2/7) of those people who had to deal with our parasite and semi-parasite offerred "Adjust his/her grade" type responses. This defies comprehension.

7. Lessons Learned

In summary, the responses to our Group Member Evaluation Forms taught us several things:

(1) Project work load distribution appears to be an area for improvement. We will

address this issue with future software development teams and try to help them with this.

(2) Our students liked working in a team environment (a little less at the end) with good agreement within groups about how satisfying this had been.

(3) The consensus ideal group size was 4-5. (Keep in mind that WAS the size of their groups).

(4) The contribution critiques were very consistent (including self-critiques) and seemed to correlate well with perceived amount of work done.

(5) There was no consensus about whether team member evaluations should affect student point totals, but there was agreement that if this was done the effect should be minimal.

(6) There were two modes of response (''adjust the grade'' and ''talk to the person'') to the question concerning what to do about identified parasites.

Thus, we think we have a vehicle that will give us rankings of team members consistent enough for identifying problem individuals. But, we are still unsure what to do with that information. For the immediate future we will take the conservative ''talk to'' approach and gather more data as we continue to fine-tune our course.

8. The Last Word

The authors want to thank the hundreds of students who have taken CS 404 during the course of its existence for their perseverance and good nature in working on their team projects. We particularly thank the 23 who completed the Group Member Evaluation Forms discussed above. Our sincerest thanks also to former lab instructors Roberto Kohler and Deborah Neely for their many contributions to this course and the project approach.

Generic copies of the Group Member Evaluation Form are available from Hubert Dunsmore, Department of Computer Sciences, Purdue University, West Lafayette, Indiana 47907. Online copies are available via electronic mail from dunsmore@purdue.edu.