THE EFFECTIVENESS OF PEER REVIEWS AND PROJECT LOGS IN TEAM PROJECTS

Gary McDonald
Merry McDonald
Computer Science/Information Systems
Northwest Missouri State University
Maryville, Missouri 64468

ABSTRACT: Much has been written during the past five years about team projects in computer information systems and computer science courses. Several authors have mentioned using peer reviews, but most used such reviews to evaluate team members and their contributions to the project. While some authors mention that they have incorporated peer reviews as a means of evaluating projects at various stages, few have focused on the peer review process or its effectiveness. A topic that has received somewhat less attention is the importance of writing in computer information systems and computer science courses. Project courses offer many opportunities for varied writing assignments. Typical assignments include documentation for users and reports to supervisors. These are important and useful types of writing assignments, but both are formal or transactional in nature. While transactional writing is important, expressive writing, such as that found in diaries or journals, first-draft papers, and personal letters, is also important. Many believe that expressive writing promotes independent thought better than other forms of writing. An excellent forum for expressive writing in project courses is through project logs. This paper describes how we have incorporated the peer review process and project logs into team project assignments and evaluates the success of each.

Keywords: Team projects, Peer Reviews, Project Logs, Database Systems, Expressive Writing

INTRODUCTION

The importance of good communication skills and good team process to the success of project teams in industry has been documented by White and Leifer [19]. Employers indicate that many recent college graduates are poor writers and are not prepared to work in teams [10]. Computer science and information systems programs have recognized the importance of teaching students to work in teams for some time, and numerous papers have appeared on how to successfully incorporate team projects into various courses [1,8,13,17,18]. Incorporating writing into computer courses has received less attention, but some papers are now beginning to appear on the subject [3,7,14].

Our own experience has indicated that students are resistant to working on teams. This is understandable, since they report a number of problems when working on teams, ranging from poor communications among members to difficulties in scheduling meeting times [12]. We have also found that students are not enthusiastic about written assignments. They enjoy writing code, but not documentation or reports. For a number of years team projects have been required in our Database Systems course, an upper-division course required for computer management systems majors and taken as an elective by many computer science majors. These projects include implementation of a database system and accompanying written documentation and reports. This paper describes and evaluates our incorporation of peer reviews and project logs in the team projects. These activities are suitable for any course involving team projects, including courses outside the area of computing.

OVERVIEW

During previous semesters, students in the Database Systems course were assigned three to four projects, with the last one being a major project requiring approximately six weeks to complete. Students worked on the projects in teams of four, with one student designated as the team leader. The team leader was responsible for coordinating the work of the team members and also served as a liaison between the team and the instructor. A typical major project was to implement a specified database application and produce a tutorial to accompany the system. The tutorial was to be written for a naive user. In addition, each team had to submit a report to its "supervisor," indicating each team member's contribution to the project, reporting on any technical problems encountered, and making suggestions for future improvements in its system. The projects were turned in near the end of the semester, requiring that they be graded during final exam week. Team members were responsible for picking up the graded projects, and some were never picked up.

We wanted to make the team experience more meaningful, to provide a better learning experience by providing more feedback, and to strengthen the writing component. To accomplish these goals, we made the following changes:

(1) We introduced a peer review process, in which each team evaluated another team's project. Following the peer review, each team was allowed to revise its original project.

(2) Students were asked to keep individual project logs. Entries were to be made two to three times per week and were to be an informal account of the student's experiences on the project.

(3) Deadlines were revised so that projects could be turned in earlier, enabling the instructor to set up conferences with teams after the final versions of the projects were graded.

(4) Several times during the course of the project, the instructor emphasized the importance of the team experience. Students do face a number of problems when working in teams, and they sometimes feel that the only reason team projects are assigned is to lighten the instructor's load by reducing grading time. Talking about the importance of working in teams and sharing supporting documentation helps to convince students that they are engaged in a valid educational endeavor which will be of use in their careers.

> ***Students are convinced that they learn best when they can get detailed feedback on an early version of their work and then hand in a final, revised version for a grade [9].***

The major changes centered around the peer review process and the project logs. The remaining sections of the paper provide more extensive details on these two activities.

THE PEER REVIEW PROCESS

Students are convinced that they learn best when they can get detailed feedback on an early version of their work and then hand in a final, revised version for a grade [9]. We felt that one of the primary weaknesses in the projects we assigned was the lack of adequate feedback, so we revised our project schedule to allow time for feedback and subsequent revision. However, instead of having the instructor provide the feedback, we had each team evaluate another team's project and then submit a written report to the other team, indicating strengths and weaknesses of the project and suggestions for improvements. This approach helped to accomplish the goals of providing more feedback to teams and strengthening the writing component. An important side benefit of the process was to give each team experience in testing and evaluating software.

The project assigned to the teams was to implement a database for a company to keep track of sales representatives, customers, orders, order lines, and parts. Each team had to design a menu-driven system for company personnel to use, with simple maintenance facilities (add, change, delete) for each table. They also had to include query options for each table and produce a fairly complex customer report to display information about each customer, his or her sales representative, and any orders the customer had on file. A tutorial and a report to the "supervisor" also had to be included. We had six teams of four students each, and they were given six weeks to implement the assigned project. At the end of that period two copies of a completely implemented system were turned in to the instructor.

A copy of each project was then given to a second team for evaluation, for a period of ten days. The students were told that they could communicate freely with one another about the projects during this time period. They knew that they would be revising their own projects after the evaluation period, and they were told that their revisions could be based not only on the suggestions made by the evaluating team, but also on ideas they picked up as they evaluated the other team's project and discussed the projects with one another. If they saw a feature in one team's project that they wanted to add to their own project, they had permission to talk to the members of the other team to find out how the particular feature could be implemented. The only restriction was that they could not directly copy or exchange code.

At the end of the evaluation period, two copies of the peer evaluation were turned in to the instructor. The instructor kept one copy; the other copy was passed on to the team being evaluated. The instructor also evaluated and assigned a grade to the first version of the project for each team. However, no information

about the instructor's evaluation or grade was given to the teams at this point. For each project, the instructor's evaluation was compared to the peer review team's evaluation to help determine a grade for the peer review portion of the project.

After the evaluation process was completed, teams were given ten days to revise their projects. The final version of the project had to include written documentation of how the team responded to the suggestions made by the evaluating team and a statement of what the team learned in doing the peer review. The instructor graded the final versions of the projects. Each team's final grade was based on the instructor's grade of the original version, with up to 15% added to that grade, based on improvements made to the initial version.

EFFECTIVENESS OF THE PEER REVIEW PROCESS

A number of authors have mentioned using peer reviews in some capacity in team projects. Sumner [15] used interim in-class reports for each phase of a project to provide teams with feedback from other students. Pigford [11] describes a project where teams implemented a system to run on microcomputers. Each team's system was tested by another team. Each team then implemented the same system to run on mainframes. Henry [6] describes projects in which students did much of the actual grading. Few authors, however, have emphasized the peer review process. We wanted to investigate the process more closely and determine its effectiveness.

We compared the evaluations made by the peer review team and by the instructor. We started by looking at two broad areas. Under the heading of "Implementation Errors," we listed any errors found by the instructor or by the peer review team, such as failure to follow specifications, system crashes caused by certain input data, and incorrect output. The peer review teams had to develop their own test data, and they were not given specific directions for testing for different types of errors. Other comments by the peer review teams and the instructor were grouped under the heading of "Suggestions for Improvement." Most of these comments dealt with the user interface: report formats, ease-of-use and attractiveness of on-screen forms, and appropriateness of prompting messages. Figure 1 summarizes the results.

The data suggests that the students did not perform very well in testing for implementation errors. They missed many errors that the instructor found, and in only one instance did a team find an error that the instructor had missed. Most projects worked correctly and were virtually error-free on "nice" sets of test data, but the students apparently failed to test the projects they were evaluating (or their own projects) on more complex or lengthier data sets. This finding supports Bullard's assertion that students do "not seem to be accustomed to generating their own test data" [1]. The students excelled, however, in coming up with suggestions for improvements, listing considerably more than the instructor. Because of the subjectiveness of this category, there was much less overlap between the instructor's and the peer review teams' suggestions, but virtually all the suggestions made by the teams were quite good. Werth states that "students are surprisingly adept at critiquing each other's work" [18], and our findings here certainly corroborate that statement, at least in some areas.

We also looked at how many of the errors were corrected and how many of the suggested improvements were made. In the case of implementation errors, four of the eight errors found by the peer review teams were corrected. The students did not see the instructor's comments prior to the time they had to make revisions, so in most cases none of those errors were corrected unless the peer review team had also reported the error. The exception to this was team 5. They apparently used the 20 days set aside for evaluation and revision to further test their own project. In their final version, all six errors found by the instructor had been corrected, although only one of those six had been found by the peer review team. Response to "Suggestions for Improvements" was better, since 25 of the 39 suggestions made by the peer review teams were implemented in the revised versions. Seven suggestions that the instructor made, but which were not mentioned by the peer review teams, were implemented in the revised versions, even though the teams had not seen the instructor's suggestions. Thus a total of 32 revisions were made in this category, three more than the total number that the instructor would have suggested. While we appreciate the students' enthusiasm for improving the user interface and the overall appearance of the product, we find their apparent tendency to focus on appearance rather than correctness disturbing.

FIGURE 1: Evaluations by Instructors and Peer Review Teams

	Implementation Errors		Suggestions for Improvement	
Team	Instructor	Peer Review Team	Instructor	Peer Review Team
1	5	0	3	2
2	6	3	4	8
3	4	0	4	9
4	4	1	6	8
5	6	1	4	4
6	3	3	8	8
Total	28	8	29	39

THE PROJECT LOG

As mentioned earlier, one of our goals for the semester was to strengthen the writing component of the course. Database Systems lends itself well to being a writing-intensive course. As described thus far, the major project alone required extensive written documentation, addressed to a number of different audiences. The tutorial for the system was written for a naive user. The report to the supervisor was addressed to a person with at least as much technical knowledge as the writer. The peer evaluation report was addressed to peers, and, finally, there was a report to the instructor of the course describing the results of the peer review process and the revisions made. All of this writing was formal in nature and is regarded as transactional writing, which has the purpose of informing or instructing an audience in conventional prose [4]. Transactional writing is certainly important, and in recent years the computer education community has begun to recognize the need for incorporating more writing skills into the curriculum. In 1983, the Michiana Chapter of the Data Processing Management Association developed a profile of the ideal computer science graduate, in which was included not only the ability "to work independently or on teams," but also the ability "to communicate both orally and in writing to technical and nontechnical audiences" [21]. Fairley lists the inability to express oneself clearly in English as one of the skills most lacking in entry-level programmers [2].

Expressive writing is another necessary skill. Entries in diaries or journals, first-draft papers, and personal letters are examples of expressive writing [4]. While many believe that writing is an excellent activity for promoting independent thought, and that expressive writing promotes such independent thought better than many other forms of writing, expressive writing has been conspicuously absent in the schools [4]. In the computer education community, expressive writing has only recently begun to attract attention. Hartman [5] and Sanders [14] describe the use of microthemes (one form of expressive writing) as a learning tool and indicate several benefits of such writing.

While many believe that writing is an excellent activity for promoting independent thought, and that expressive writing promotes such independent thought better than many other forms of writing, expressive writing has been conspicuously absent in the schools [4].

Since all of the writing assignments in our Database Systems course were in the transactional category, we felt that any additional writing assignments should be in the expressive category. Students were asked to keep a project log, beginning on the day the project was assigned and continuing until five days after the final version of the project was turned in. Entries were to be sent to the instructor using electronic mail. Students were asked to make at least two or three entries per week. They were told that the log was to be an informal account of their experiences on the project and their interactions within the team and that they should feel free to use it as a means of venting their frustrations as well as reporting on more positive aspects of the project. The project log was graded on the basis of number and length of entries, the distribution of entries throughout the life of the project, and content.

EFFECTIVENESS OF PROJECT LOGS

We had never required students to submit logs or journals in any of our other classes, so we viewed the project log assignment as an opportunity to learn how to incorporate expressive writing in computing classes. We expected that the students would doubt the value of such an assignment, and we were also concerned about the additional grading time that would be required. As it turned out, the logs were interesting to read and easy to grade. However, students need guidance in writing their initial log entries. Several samples of good log entries, passed out when the project is assigned, would probably result in much better entries from the students. The potential value of the project log as a means of self-discovery and a tool for thinking [4] also needs to be thoroughly explained.

The main problem during our trial semester was that many students made very few log entries. Other students made regular entries, but they were simply technical reports of "What our team did today." However, even though virtually all of the students were initially hesitant and unsure about the worth of the log, a few students did their best to make it work. They made regular log entries, many quite lengthy, and tried to report not only on the activities of the team but also their reactions to these activities. These students convinced us that project logs are workable and instructive (to both students and teachers). Woodfield suggests the study of small group process theory as an area of research [20], and logs could be particularly useful to the instructor in studying group dynamics of project teams. The benefits to the student come from the fact that, if properly done, the logs help the student to focus on the team interactions, the role of each individual on the team, and the impact of individual actions on the success of the team project. Thus, in addition to the technical knowledge gained from the project, students can also increase their "people skills." Many students feel frustrated with team projects, perhaps because of communication or scheduling problems. Project logs can be an effective means of enabling students to verbalize, confront, and solve the problems which are causing their frustration. The project logs might be of more benefit if some class time was spent discussing group process strategies and techniques [16].

FIGURE 2: Questionnaire Results

	A	B	C	D	E
I enjoyed working in teams.	22	39	17	22	0
Working on a team was a valuable experience.	48	39	13	0	0
The peer review process was a valuable experience.	35	43	9	13	0
The peer review of our team's project was very helpful.	26	48	22	4	0
Team projects should be continued.	54	32	14	0	0
I enjoyed making log entries.	4	14	32	27	23
Logs are a valuable part of the project.	14	13	32	32	9
The project log stimulated me to think about issues I might have ignored otherwise.	9	18	36	23	14
The project log provided me with an additional avenue of communication.	14	36	32	9	9
The project log should be continued.	5	27	36	27	5

A=Strongly Agree; B=Agree; C=Neutral; D=Disagree; E=Strongly Disagree
(Entries indicate percentage for each response.)

STUDENT RESPONSE

At the end of the semester we distributed an evaluation instrument to each member of the class to help us determine the success of the team projects in general, and the peer review process and the project logs in particular. Figure 2 summarizes the results of that questionnaire.

In regard to team projects and the peer review process, the results are encouraging. Even though there are some students who are less than enthusiastic about working in teams, the overwhelming majority recognize the importance of learning to work in teams and think team projects should be continued in the future. The peer review process received a very favorable response. The responses by the students to the project logs were not as positive, although a few students did respond quite favorably. We hope that those students who responded positively were those who turned in the best log entries and had a high degree of initial commitment to the log process, but we have no way of determining that from the data we have. We expect that we will get a much more positive response from students if we spend more time explaining the potential benefits of the logs to them and give them more guidance in the types of entries they should make.

CONCLUSION

This paper describes our experiences with incorporating peer reviews and project logs into team projects. Those experiences and the data we collected have convinced us of the usefulness of each. The peer review process, as we have described it, provides the students with experience in reviewing software, while also providing them with feedback on their own projects, enabling them to improve the final versions. The major weakness appears to be that students are not proficient at devising adequate test data. In future semesters, we plan to provide more instruction in this area. In addition, the importance of a correct implementation, which meets the specifications and is robust, needs to be emphasized. Project logs can also be used very effectively and may provide an important tool for studying group dynamics in team projects. Students can derive many benefits from such logs, but if they have no previous experience in writing logs or journals they will need extensive initial guidance.

REFERENCES

1. Bullard, C.L., Caldwell, I., Harrell, J., Hinkle, C., Offutt, A.J., "Anatomy of a Software Engineering Project," *SIGCSE Bulletin*, 20:1 (February, 1988), 129-133.
2. Fairley, R., *Software Engineering Concepts*, McGraw-Hill, NY, 1985.
3. Flaningam, D.L., Warriner, S., "Another Way to Teach Computer Science Through Writing," *SIGCSE Bulletin*, 19:3 (September, 1987), 15-16.
4. Fulwiler, T., "Writing: An Act of Cognition," In *New Directions for Teaching and Learning: Teaching Writing in All Disciplines*, no. 12, C. W. Griffin, ed., Josey-Bass, San Francisco, CA, December, 1982.
5. Hartman, J.D., "Writing to Learn and Communicate in a Data Structures Course," *SIGCSE Bulletin*, 21:1 (February, 1989), 32-36.
6. Henry, S., "A Project Oriented Course on Software Engineering," *SIGCSE Bulletin*, 15:1 (February, 1983), 57-61.
7. Jackowitz, P.M., Plishka, R.M., Sidbury, J.R., "Teaching Writing and Research Skills in the Computer Science Curriculum," *SIGCSE Bulletin*, 22:1 (February, 1990), 212-215.
8. Leeper, R., "A Project Course in Database," *SIGCSE Bulletin*, 22:1 (February, 1990), 78-80.
9. Light, R.J., *The Harvard Assessment Seminars: Explorations with Students and Faculty about Teaching, Learning, and Student Life*. Available from the Harvard Assessment Seminars, Graduate School of Education, Cambridge, MA, 1990.
10. Magner, D.K., "Many Colleges Design Courses and Programs to Prepare Seniors to Live in the 'Real World'," *The Chronicle of Higher Education*, 36:27 (March 21, 1990), A33-A35.
11. Pigford, D.V., "A Management System for Monitoring and Assessing the Group-Oriented Database Project," *SIGCSE Bulletin*, 19:1 (February, 1987), 9-

18.

12. Pournaghshband, H., "The Students' Problems in Courses With Team Projects," *SIGCSE Bulletin*, 22:1 (February, 1990), 44-47.
13. Saiedian, H., "A Practical Approach to the Database Management Systems Course," *Proceedings of the Eighth Information Systems Education Conference*, (Chicago, Illinois, October 13-14, 1990), 49-54.
14. Sanders, D., "Writing in the Fourth Generation," *Proceedings of the Eighth Information Systems Education Conference*, (Chicago, Illinois, October 13-14, 1990), 76-80.
15. Sumner, M., "The Senior Information Systems Design Project Seminar," *SIGCSE Bulletin*, 19:1 (February, 1987), 2-8.
16. Tellep, A., "Information Concerning Group Processes Is Beneficial to Students in a Systems Development Course," *Proceedings of the Eighth Information Systems Education Conference*, (Chicago, Illinois, October 13-14, 1990), 7-11.
17. Tenny, T., "Leadership Style vs. Success in Student Chief Programmer Teams," *SIGCSE Bulletin*, 19:1 (February, 1987), 103-114.
18. Werth, L.H., "Integrating Software Engineering into an Intermediate Programming Class," *SIGCSE Bulletin*, 20:1 (February, 1988), 54-58.
19. White, K.B., Leifer, R., "Information Systems Development Success: Perspectives from Project Team Participants," *MIS Quarterly*, 10:3 (September, 1986), 215-223.
20. Woodfield, S.N., Collofello, J.S., Collofello, P.M., "Some Insights and Experiences in Teaching Team Project Courses," *SIGCSE Bulletin*, 15:1 (February, 1983), 62-65.
21. --, "Profile of Ideal Graduate in Information Processing," Business/ Education Forum, Michiana Data Processing Association, South Bend, IN, 1983.

AUTHORS' BIOGRAPHY

Gary and Merry McDonald are Professors of Computer Science and Information Systems at Northwest Missouri State University in Maryville, Missouri. They both received Ph.D.'s from the University of Texas at Austin. Their current research focuses on expert systems and databases.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096