Improving Individual Learning in Software Engineering Team Projects

Dr. Joanna F. DeFranco, Pennsylvania State University

Joanna F. DeFranco is Assistant Professor of Software Engineering and a member of the Graduate Faculty at The Pennsylvania State University. Prior to joining Penn State, she held faculty positions at Cabrini College and the New Jersey Institute of Technology. She also held a number of positions in industry and government including an Electronics Engineer for the Naval Air Development Center in Warminster, PA and a Software Engineer at Motorola in Horsham, Pa. Dr. DeFranco received her B.S. in Electrical Engineering from Penn State University, M.S. in Computer Engineering from Villanova University, and Ph.D. in Computer and Information Science from the New Jersey Institute of Technology. She is a member of ASEE and has had numerous publications in journals and conference proceedings. She is also on the curriculum advisory board for a local technical high school.

Dr. Colin J. Neill, Pennsylvania State University

Colin J. Neill, associate professor of software engineering and systems engineering and Director of Engineering at Penn State Great Valley, earned his Ph.D. in software and systems engineering, M.Sc. in communication systems, and B.Eng. in electrical and electronic engineering from the University of Wales, Swansea, United Kingdom. He teaches a wide range of software and systems engineering courses in design, architecture, project management, systems thinking, and IT strategy. He has written more than seventy articles on software design, architecture, process, and management, and serves as associate editor-in-chief of Innovations in Software and Systems Engineering.

Improving Individual Learning in Software Engineering Team Projects

Abstract

The focus of our research is on determining the factors that facilitate both team success and individual learning during team-oriented project-based learning. Of particular interest is the efficacy of collaborative learning approaches in general for individual engineering students. Our results from a large scale experiment provide no evidence that working on a successful and effective team affects individual exam performance. Thus, we will propose a qualitative study to determine the best ways to structure team work to enhance individual leaning.

Introduction

For a number of reasons, team-based projects are frequently included in software engineering programs. Educators integrate team projects into the curriculum to emulate real world development situations, expose students to the challenges and benefits of team-working, and allow students to tackle problems of larger scales and complexities than they could otherwise do alone. Furthermore, there is evidence that collaborative learning methods are more effective than the traditional teacher-centered methodology [3,4]. Finally, the Accreditation Board for Engineering and Technology (ABET) expects students to gain proficiency in team work[5]. While the reasons for their inclusion are numerous, team-based projects, and team-based assessments in particular, are not without their problems.

One focus of our research has been in addressing the difficulties students experience in team-work. We have developed a framework of guidelines and practices that facilitate effective teams. The framework has been successful with respect to team performance and project outcomes, but as our studies have progressed we have discovered that individual learning of team members is not positively impacted in the ways we had anticipated.

One explanation may be that effective teamwork aids in the development of structural and procedural knowledge – the knowledge needed to apply course content[7], but does not necessarily improve declarative and propositional knowledge – the knowledge needed for success in traditional tests. Put another way, students can build effective systems while not fully understanding the underlying theories and concepts they are applying. In essence, the old saw "that an effective team is greater than the sum of its parts" appears to hold true, but as we know our education system concerns individual achievement and learning. Thus, another focus of this research is to determine the factors surrounding the improvement of <u>both</u> procedural and declarative knowledge.

We employed the first three stages of a model for effective collaboration called the Cognitive Collaborative Model (CCM). The CCM is a six-stage cognitive model that takes into consideration the cognitive and social activities that occur during collaborative problem solving by facilitating problem formulation, problem analysis, and design tasks within teams[6]. The CCM prescribes tactics to ensure collaboration, but does not imply any specific analysis or design techniques, and thus does not impact any techniques taught to the students such as object-oriented analysis and design, as in this case.

Using pre- and post-testing, we studied course outcomes of software engineering graduate students learning software systems design that have also utilized the CCM in a systems design project and contrasted these results with a control group.

Background

An engineering graduate's ability to function within a multidisciplinary team is one of the 11 program outcomes ABET requires of accredited engineering programs. There appears to be little consensus regarding the efficacy of student teams, however. On the one hand considerable literature is devoted to the group and team learning benefits of social constructivist approaches such as classroom discussion and experiential learning[13,18]. Through peer interaction and collaboration students are able to synthesize and evaluate their ideas collectively[10,16,19] and are forced to reflect upon and reason about their ideas at greater depth than when working individually[17].

In contrast, there are numerous studies that show the difficulty students experience working in teams. Student's frequently cite that they have little influence over their team-mates; they believe their grade will not reflect their contribution or competence; and the transaction cost of scheduling meetings, and working collaboratively are not worth the rewards, of which they see few[9]. These bad team experiences can have a profound impact on those students who are subsequently soured on teamwork far beyond their studies and in to the workplace[8]. Indeed, one recent study even demonstrated how the default mode of team-working in student populations (cooperation) can drastically impact learning[12]. They found that when paired together in engineering labs students will cooperate using a divide and conquer approach rather than collaborate as a unit, resulting in a significant reduction in the passing rate for the class and a twenty percentage point difference in median scores.

A fair interpretation of these contrasting results, then, is that collaboration must be fostered in student teams to ensure that the potential benefits of collaborative learning are realized and the negative impact of cooperative working is minimized. More importantly, however the framework of guidance given to student teams should reflect the true objective of team-working – that each student learns the course content.

The Cognitive Collaborative Model

The CCM is based on the Dual Common Model for Problem Solving and Program Development where the focus is on individual cognitive aspects of problem solving and programming[20]. Deek (1997) extensively reviewed existing problem solving methodologies and developed a comprehensive problem solving model which integrates general problem solving methodology with program development tasks. The model takes into consideration the cognitive knowledge and skills needed at each stage of the process. The integrated model, called the Dual Common Model (DCM), identifies for each problem solving/program development task, the specific cognitive techniques required to accomplish that task. A brief overview of the problem solving tasks is as follows:

1. Formulating the problem: This stage leads to an organized representation of all relevant problem information: the goal, givens, unknowns, conditions and problem constraints.

2. Planning the solution: During this stage, the user identifies and evaluates or assesses alternative possible solutions, and also partitions the problem by refining the overall problem goal into sub-goals.

3. Designing the solution: This involves sequencing sub-goals, determining whether the sub-goals require further decomposition, establishing relationships among the various solution components and the associations between data and sub-goals.

4. Translation: At this stage, program development skills are used to translate the solution design into a coded solution.

5. Testing: At this stage the program is tested to verify that it meets the solution specifications.

6. Delivery: At this stage the solution and results are documented, presented or disseminated.

The CCM is also made up of six stages; however, the tasks of the CCM facilitate team cognition. Each stage of the CCM is further broken down into three phases. When measuring the team project outcomes and/or the shared mental model of the team, the first two stages of the CCM are of most significance: Problem Formulation and Problem Analysis. The phases and objectives of these two stages are shown in Figure 1.

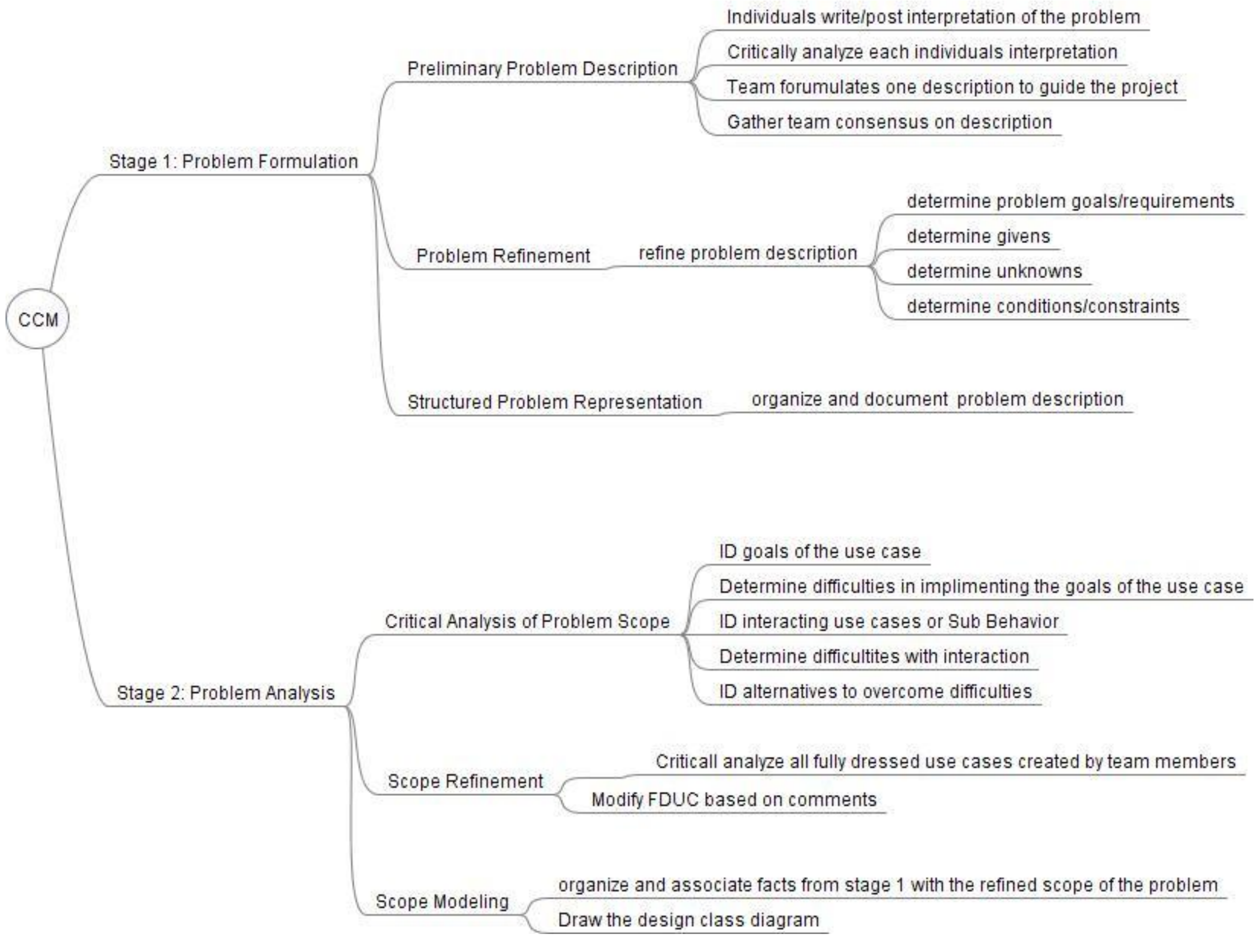

Figure 1. Collaborative Aspects of the CCM

The goal of the first three phases of the Problem Formulation stage (Preliminary Problem Description, Preliminary Mental Model, and Structured Problem Representation) is for the team to collectively understand the problem, and this is done through structured questions that each team member must answer and then discuss to resolve inconsistencies, misconceptions, and assumptions. In addition, the individuals need to communicate effectively and the group also needs to listen and ensure each member has the correct understanding of the problem.

For example, in the first stage, the collaborators are to agree upon a preliminary problem description to make sure each team member has the same understanding of the problem. The model guides each team member to create a description in their own words and share it with each team member. The descriptions are discussed openly and any differences of opinion that cannot be resolved are voted upon until a consensus is reached. Next, the team is charged with answering questions, shown in Table 1, that focus on expanding and deepening their understanding of the problem as they have described it. Table 1 also shows the specific collaborative aspects present during this phase of the model.

Table 1. Stage 1, Phase 2 of the CCM

Questions for Problem Formulation (*Stage 1*) Preliminary Mental Model (*Phase 2*):	Example of the internal structure of the CCM for Stage 1, phase 2.
Q: What are the goals of the system? *Q: Do any of these goals require clarification?* *Q: Are there any other explicit or implicit problem requirements?* *Q: What are the givens?* *Q: Are there any flow control related inputs or givens?(flow control: think about the order of a few scenarios.)* *Q: What are the unknowns?* *Q: Are there any conditions and constraints of the system.*	***Collaborative Modality:*** message board for Problem Understanding ***Collaborative Processes:*** pull and push information from the preliminary problem description, idea generation activity, negotiation, coordination ***Collaborative Side Effects:*** adoption barrier to the groupware, eagerness, free riding, consensus building, conflict, cognitive synchronization ***Collaborative Administration:*** initiate the vote for correctness of problem facts

The final part of this stage is where the team will identify and organize any relevant information of the problem thus creating a knowledge base from which the team will begin their Problem Analysis (stage 2).

The goal of the second stage phases of the CCM, Critical Analysis of Problem Scope, Scope Refinement, and Scope Modeling, are for the team to answer questions and gather information to further analyze the problem. Specifically, they are going through the process of goal decomposition where they are refining goals into smaller sub-goals that are more easily solved. For example, in the initial phase of stage two, the team is beginning to critically analyze the problem scope. The team members then share their ideas for use cases. A vote commences to determine the direction that will be followed. Now that the team has agreed upon direction, the scope of the problem is refined and the relevant analysis models (use case models and domain models in the case of OOA&D) are updated.

In the remaining four stages of the CCM the team would be translating the plan into a detailed design, implementing the design, testing, and finally delivering the solution. Working through the first two stages of the CCM, the team is able to conceptualize the problem resulting in a more effective plan and in theory able to implementing a better solution.

Hypothesis and Experimental Methodology

We conducted this research using sections of a three credit graduate course in software systems design. The participants were working professionals enrolled in a professional master's degree in software engineering that requires industrial experience for admission so it is likely had experience working in teams. They would also have had experience participating on team projects in other courses in the program as software systems design is not the first course in the program. However, none of those courses would have provided guidance or training in teamwork.

The data collected was from three different sections of the same course. One section was the control group. The other two sections were the condition groups. Prior to lesson-one, each student completed a timed online benchmark exam (pre-test) to assess their understanding of the course topics before any material had been covered. The students in all sections were randomly divided into teams of three to work on the project. Each team was then assigned one of two projects that were of equivalent complexity.

The condition teams were given the CCM instructions per the CCM described previously and their progress through the phases of the CCM was recorded by the team members in the team discussion forums. The first time we ran the experiment, the condition teams were given the CCM instructions via a document each team member needed to download. As we didn't feel this was an effective way to ensure that the teams followed the CCM instructions, for the second condition group we changed the directions and questions in the discussion forums so that they asked the teams to post the outcomes and their reasoning for each phase of the CCM, thus ensuring that those teams followed the guidelines of the framework. In contrast, the teams the control group were only given the project deliverable descriptions, as usual for this course. The discussion forums were monitored by each course section instructor to confirm participation of each team member as well as each team following their given instructions. At week five of the seven week graduate course, each student completed a timed online post-test.

In a previous experiment we showed that the CCM facilitates more effective sharing of individual mental models within a team, and thus generates a better team mental model. We have determined this by analyzing the similarity among team concept maps created by each individual of their understanding of the problem their team is solving[11]. In this experiment, we determined the degree of individual learning attained by the differential of the pre- and post-assessments.

Data Analysis and Results

To assess the pre- and post-tests we employed three judges who were the instructors of each section. To avoid any instructor bias, the instructors did not evaluate their own course assessments, thus we had two sets of results for each test. Each instructor employed a common rubric point system agreed upon prior to the start of the courses.

Since this analysis involves the averaging of the two judges' scores for each group, we must first test the inter-rater reliability of the two pairs of judges for each group. Tables 2 shows the comparison statistics of the evaluations of each pair of judges who assessed the condition and the control groups.

Table 2: Inter-rater reliability

Group	Test	Judge	Mean	Std Dev.	T	p=
Control	Pre	1	26.9	16.5	-3.02	**0.005**
		3	46.7	22.4		
	Post	1	76.5	14.9	-.09	0.929
		3	76.9	11.8		
Condition 1	Pre	1	26.3	13.8	-3.84	**0.0**
		2	44.5	16.7		
	Post	1	73.7	16	-0.93	0.357
		2	78.7	18.8		
Condition 2	Pre	2	46.4	18.2	0.45	0.653
		3	43.8	16.1		
	Post	2	90.42	6.52	-0.24	0.811
		3	90.94	6.54		

While the post-test results for both the condition and control groups were reliable, with no significant difference between the judging (p=.929, p=.357, p=.811), the pre-tests for the control group and condition group 1 show a significant difference between the judges scores (p=0.005 and p=0.0). This potentially confounds the results, so that averages across judges may not be reliable. We will therefore show the overall statistics for the condition versus control groups for all judges as well as the average.

A T-test was performed to assess the resulting differential for each student between the two groups (CCM vs. No CCM) to determine if the groups were significantly different. The results of the analysis are summarized in Table 3.

Table 3: T-test of the Post- and Pre-test differential of each group

		Control (n=18)	CCM 1 (n=21)			CCM 2 (n=18)
Judge 1	T = -.39 p = .698	μ = 49.6 σ = 18.8	μ = 47.3 σ = 15.9			No Judging
Judge 2	T =.55 p = .588	No Judging	μ = 34.2 σ = 20.0		T=-1.61 p=.117	μ = 44.1 σ = 18.3
Judge 3		μ = 30.2 σ = 25.3	No Judging		T=2.42 **p=.022**	μ = 47.2 σ = 15.8
Avg for CCM 1	T = -.15 p=.884	μ = 39.9 σ = 21.1	μ = 40.8 σ = 17.1	Avg for CCM 2	T = .92 p = .366	μ = 45.6 σ = 16.4

These results reveal that Judge 1 found that the control group marginally outperformed the condition group, while a comparison of Judge 2 and Judge 3 reveal the CCM 1 marginally outperformed the control group. Judge 3 found that CCM 2 significantly outperformed the control group. The overall average of all judges was also not significant and thus the hypothesis, that use of the CCM, and therefore effective teamwork, will facilitate improved individual learning is not confirmed. We suspected that although we encouraged collaborative work among the team members that the team members actually worked cooperatively (individually) on most aspects of the project.

Discussion and Future work

The research presented in this paper is part of a large scale study that focused significantly on the creation of team guidelines and instructions to facilitate effective collaboration in distributed teams and the analysis of those guidelines with respect to project outcomes and team coherence[10,11,14,15]. In both regards, that framework has been shown to be effective, but this latest study revealed that despite improvements in team performance, students' individual learning is not improved, which calls into question the legitimacy of those teams as a learning mechanism.

Our goal for our future research is to determine why the declarative knowledge was not enhanced along with the structural or procedural knowledge. A plausible explanation is that the negotiated knowledge structure that emerges in effective teams may negatively affect the declarative knowledge typically found in a course assessment.

We therefore propose a mixed-method research study examining the reasons for the absence of individual learning in team-centered courses and the refinement of the CCM guidelines for educators and students for the effective facilitation of learning teams. We will accomplish this through the continuation of our existing quantitative research methodology, focused on measuring the impact of the CCM, supplemented by new qualitative research that is focused on students' experiences in facilitated teaming in order to uncover students' specific roadblocks to learning in teams.

The additional qualitative approach, will determine specific roadblocks to learning. A basic qualitative interpretive orientation allows the researchers to explore the lived experience of the student. A purposeful sample of students will be chosen to participate. Since the subject base will be online, observations will not be performed; however, in-depth semi-structured open-ended interviews and surveys will. Questions for each interview will be developed from the survey responses allowing time for follow up, to dig deeper into the experience of each participant so that each student feels understood. Interviews will be carried out via phone conversations and/or email communications, whichever is most convenient for participants. The researcher will not be assessing the students at any point in their graduate programs, so there will be no conflict of interest, ideally increasing the strength of the findings. Transcriptions of the interviews will be analyzed using the constant comparative method. Themes will be identified and a coding and retrieving process will commence to reveal broad categories in the data. To enhance trustworthiness, the interpretation and labeling of categories will require consensus among all of the researchers. These categories will be used to inform faculty about the best way to structure group work in an online environment.

Bibliography

1. M. Bass, Monitoring GSD projects via shared mental models: A suggested approach, Proc 2006 Int Workshop Global Software Dev Practitioner, Shanghai, China, 2006, pp. 34–37.
2. J. Espinosa, R.E. Kraut, F.J. Lerch, S.A. Slaughter, J. Herbsleb, and A. Mockus, Shared mental models and coordination in largescale, distributed software development, Proc Int Conf Inform Syst, New Orleans, LA, December 16–19, 2001.
3. Schroeder, C. M., Scott, T. P., Tolson, H., Huang, T., & Lee, Y. (2007). A meta-analysis of national research: Effects of teaching strategies on student achievement in science in the United States. Journal of Research in Science Teaching, 57(10), 149–174.
4. Springer, L., Donovan, S. S., & Stanne, M. E. (1999). Effects of small-group learning on undergraduates in science, mathematics, engineering and technology: A meta-analysis. Review of Educational Research, 69(1), 21–51.
5. Accreditation Board for Engineering and Technology, Inc., "Criteria for Accrediting Engineering Programs", www.abet.org, ABET, 2012.
6. DeFranco, J.F. Collaborative Problem Solving and Program Development Model, Ph.D. Dissertation, New Jersey Institute of Technology, 2002
7. Jonassen, D., Marra, R., "Concept Mapping and Other Formalisms as Mindtools for representing knowledge", Association for Learning Technology Journal, Volume 2, Number 1, 1994.
8. Buckenmeyer, J.A. (2000) "Using teams for class activities: Making course/classroom teams work," *Journal of Education for Business*, Vol. 76, No. 2, Nov. 2000, pp. 98-108.

9. Caspersz, D.M., Wu, M., Skene, J. (2003) “Factors Influencing Effective Performance of University Student Teams,” in *Proc. 26th Annual International HERDSA Conference*, Christchurch, NZ, July, 2003
10. Corden, R.E. (2001). Group discussion and the importance of a shared perspective: Learning from collaborative research. *Qualitative Research*, 1(3), 347-367.
11. DeFranco, J., Neill, C., Clariana, R., (2011) “A Cognitive Collaborative Model to Improve Team Performance – A Study of Team Outcomes and Mental Model Sharing”, *Systems Engineering Journal*, Volume 14, No 3, 2011.
12. Greco, E. and Reasoner, J. (2010) Student Laboratory Skills and Knowledge Improved through Individual Lab Participation, *Proc. ASEE Annual Conference*, Louisville, KY, June 2010.
13. Hilborn, R.B. (1994) “Team learning for engineering students,” *IEEE Trans. Educ.*, vol. 37, no. 2, 1994, pp. 207–211.
14. Neill, C.J., DeFranco, J.F., (2011) “Improving Team Learning in Systems Design”, *Proc. ASEE Annual Conf*, Vancouver, Canada, June 2011.
15. Neill, C.J., DeFranco, J.F. (2011) “Problem-Solving Style and its Impact on Engineering Team Effectiveness.” *Proc. 9th Conference on Systems Engineering Research*, Redondo Beach, CA. April 2011
16. Nystrand, M. (1996). Opening dialogue: Understanding the dynamics of language and learning in the English classroom. New York: Teachers College Press
17. Reznitskaya, A., Anderson, R.C., & Kuo, L. (2007). Teaching and learning argumentation, *The Elementary School Journal*, 107(5), 449-472.
18. Schroeder, C. M., Scott, T. P., Tolson, H., Huang, T., & Lee, Y. (2007). A meta-analysis of national research: Effects of teaching strategies on student achievement in science in the United States. *Journal of Research in Science Teaching*, 57(10), 149–174.
19. Weber, K., Maher, C., Powell, A., & Lee, H.S. (2008). Learning opportunities from group discussions: Warrants become the objects of debate. *Educational Studies in Mathematics*, 68 (3), 247-261.
20. Deek, F.P., An Integrated Environment For Problem Solving and Program Development, Unpublished Ph.D. Dissertation, New Jersey Institute of Technology, 1997.