# AC 2012-3117: A STUDY OF INDIVIDUAL LEARNING IN SOFTWARE ENGINEERING TEAM PROJECTS

**Dr. Colin J. Neill, Pennsylvania State University**

Colin J. Neill is Associate Professor of software and systems engineering at Penn State University's School of Graduate Professional Studies, where he is the Director Engineering Programs. Neill has developed and taught more than a dozen courses in support of the graduate programs in software engineering, systems engineering, engineering management, and information science in topics including software systems design, system architecture, project management, and systems thinking. He has published more than 70 articles in refereed journals and conference proceedings, including Systems Engineering, IEEE Computer, Journal or Systems and Software, Software Process: Improvement and Practice, and IEEE Software. He is the author of Antipatterns:Managing Software Organizations and People and Associate Editor-in-Chief of Innovations in Systems and Software Engineering.

**Dr. Joanna F. DeFranco, Pennsylvania State University, Great Valley**

Joanna DeFranco earned her Ph.D. in computer and information science from New Jersey Institute of Technology, M.S. in computer engineering from Villanova University, and B.S. in electrical engineering from Penn State University Park. She teaches graduate courses including: Problem Solving, Project Management, Software Systems Design, Computer Forensics, Ethics and Values in Science and Technology, Advanced Software Engineering Studio and Information Technology seminar. Previous to entering academia, DeFranco held a number of positions in industry and government, including Software Engineer for Motorola in Horsham, Penn., and Electronics Engineer for the Naval Air Development Center in Warminster, Penn. She has published a number of articles in journals and conference proceedings in the area of collaborative problem solving, group cognition, e-learning, and global engineering.

**Dr. Raghvinder S. Sangwan, Pennsylvania State University, Great Valley**

Raghu Sangwan is an Associate Professor of software engineering in the Engineering Division at Pennsylvania State University's Great Valley School of Graduate Professional Studies in Malvern, Penn. He joined PSU in 2003 after a more than seven-year career in industry, where he worked mostly with large software-intensive systems in the domains of healthcare, automation, transportation and mining. His teaching and research involves analysis, design, and development of software systems, their architecture, and automatic and semi-automatic approaches to assessment of their design and code quality. He also holds a visiting scientist appointment at the Software Engineering Institute at the Carnegie Mellon University in Pittsburgh, Penn. Sangwan received a Ph.D. in computer and information sciences from Temple University, Philadelphia, Penn., in 1997. He is a Senior Member of the IEEE and ACM.

# A Study of Individual Learning in Software Engineering Team Projects

## Abstract

A large scale experiment to determine if improved team cognition leads to improved individual learning has been designed. Specifically, the goal of this research is to determine if working on an effective team benefits or impedes a student's learning of the course content. The literature appears to focus on team performance, team outcomes, and benefits of teams by combining individual resources; but does not focus on the benefits of the individuals on the team, where a benefit could be learning for example. Our results, however, do not support the hypothesis that the guidelines that facilitate effective student teams also improve individual team member learning.

## Introduction

Software engineering projects at anything other than the smallest scales involve teams of engineers. It is not surprising then, that courses in software engineering often include group and team projects as part of both the students' learning and their assessment. An informal survey of the courses in our own graduate software engineering program revealed that over half include group projects that contribute significantly (30% or more) to a student's final grade. This is apparently in line with other engineering disciplines. In a survey of instructors at eight engineering schools Felder[6] found that 24% always assigned a group project while another 52% assigned them in some courses, while a second broader survey showed that 80% of capstone courses included team-oriented projects[10].

Given the prevalence of team projects we have conducted a number of experiments over the last three years to investigate the effectiveness of engineering teams, and mechanisms to improve that effectiveness[2,3,5]. Through these experiments we have established a simple model for student collaboration that aids more rapid convergence on the problem at hand and the chosen solution. We have further demonstrated that the efficacy is borne from greater team mindshare – more correctly termed shared (or team) mental model convergence[4]. While we have established that the collaborative model aids a team in achieving its purpose, we still wanted to test whether improved team outcomes also implied improved individual learning for each student. That is to say, do the team outcomes reflect individual learning in the team members?

The implications of this, if not the case, are broad. Grades assigned to individuals based upon a team project would be inaccurate representations of those students' true attainment and the role of team projects would be questioned. Of course, one could still argue that provided a team delivered a successful product or project, one that is analogous to 'real world' software projects, the goal of the educational program is achieved, but we would still like to know that successful outcomes are borne of individual learning rather than simply effort or repetition.

The collaborative model employed is Cognitive Collaborative Model (CMM) and 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 during collaboration[1]. The CCM model 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.

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

Anecdotally we frequently hear from employers that it is "soft skills" that they most desire in their engineering and technical staff. While this is likely only true when those employees are proficient engineers, it is clear that engineering is a team activity[11] and the ability of an engineer to work effectively in a team is a keenly sought after skill[12].

Beyond the external need for team-oriented engineers, there is evidence that collaborative learning methods are more effective than teacher-centered methods[17,18] and even when compared to active learning and constructivist approaches[19]. We would expect, then, that engineering students working in teams would see improved individual learning. Of course, a critical aspect is the nature of that teamwork. Kittleson and Southerland [2004] observed engineering capstone teams and found that they were cooperative, in that they divided the work between the team members, rather than collaborative – an outcome we have seen in our courses, and a motivating factor behind the CCM. This divide-and-conquer approach was also seen in a study of pair-working in engineering labs, and it led to significant declines in individual achievement[7]. This implies that team discourse (the degree and nature of interaction between the team members) is an important factor in individual learning [15] and is certainly in keeping with social constructivist learning theory.

There may be other factors, however, that inhibit learning in engineering teams. Cognitive style, the problem-solving preferences of individuals, appears to influence learning. We have previously investigated the impact of one aspect of cognitive style, Adaption-Innovation theory[20], and found that while teams comprised of diverse problem solvers (some who prefer to adapt existing solution versus those who prefer to create new solutions) may encounter greater difficulty in collaborative working, those issues appear to be resolved with the facilitated teamwork that comes with the CCM[3]. An alternative cognitive style theory (Field independence / Field Dependence[9] may be more revealing, however as it classifies people as independent or dependent learners, where independent learners prefer, as one might expect, to work alone, and often have poor interpersonal skills[8]. Furthermore, the skills of independent learners align more closely with engineering than those of dependent learners, which leads to a tendency for engineers to be independent learners[13]. The impact of this on their ability to work collaboratively in a team, and further, for them to learn through teamworking and collaborative approaches may be substantial and may require additional training in interaction[14].

**Theory**

The main goal of the CCM is to assist in facilitating critical thinking and effective problem solving among collaborators. The CCM described briefly in this paper is made up of six stages: Problem Formulation, Problem Analysis, Solution Design, Solution Translation, Solution Testing, and Solution Delivery. Each stage is further broken down into three phases. For the purposes of this study we will only focus on the details of the first two stages of the CCM: Problem Formulation and Problem Analysis. The three phases of the Problem Formulation stage (stage 1) are: Preliminary Problem Description, Preliminary Mental Model, and Structured Problem Representation. The goal of this stage is for the team to answer questions and gather information for each team member to understand the problem. In addition, the individuals need to communicate effectively and the group also needs to listen and make sure 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. Each team member discusses and votes to determine one problem description. Next, the team is charged with answering questions to develop a preliminary mental model. For example, the questions help the team to discuss and determine givens, unknowns, conditions and constraints on the problem. 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 three phases of the second stage of the CCM, Problem Analysis, are: Critical Analysis of Problem Scope, Scope Refinement, and Scope Modeling. The goal for this stage of the CCM is 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 this direction is refined where detail is added to use cases for phase 2 and a design class diagram is the output of phase 3.

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 implementing a better solution.

## Hypothesis and Experimental Methodology

The goal of this research was to investigate the individual learning on successful engineering teams. In previous publications we have shown support of the following three hypotheses:

H1. Use of the CCM by team members will improve the project outcomes for that team.

H2. Use of the CCM will facilitate the forming of a team mental model.

H3. Use of the CCM will facilitate team learning.

Four separate experiments were conducted to test the three hypotheses. The first two experiments focused on H1 by examining the project outcomes from graduate student engineering teams. The third experiment addressed H2 by eliciting and assessing each team member's mental model of the project using concept maps. In the fourth experiment both the individual course assessments as well as the team's project artifacts were evaluated to address H3.

In this fifth experiment, we tested a fourth hypothesis:

H4. Use of the CCM in teams will improve individual learning.

Specifically, the goal of this experiment was to determine if individual learning would be enhanced by teams utilizing the CCM. The subjects participating in the study were 39 graduate software engineering students in two sections of the same course, taught from the same materials. One section was provided the CCM guidelines for their team projects at their first team meeting as a paper handout. The students were assigned to random teams of three or four people where they were given one of four equivalent assignments.

Rather than assess the artifacts created by the teams, as we have done in previous reported experiments, we chose to conduct pre- and post-testing to determine the degree of individual learning. As the pre-test, all students were given a benchmark exam that tested their understanding of the course topics before any material had been covered. At week five (of the 7 week course) each student was then assessed again, through a second exam similar in nature to the first as the post-test.

The course, software systems design, focuses on the principles of object-oriented analysis and design. This course covers the basics of object-orientation (coupling, cohesion, encapsulation, inheritance, polymorphism, etc.) as well as the principles of analysis, design, and architecture of object-oriented software systems and their representation using the UML.

The pre-test consists of eight questions (seven short answer and one multiple choice) assessing what the students already knows about basic object-oriented concepts. The post-test covers the UML diagrams, notation, and semantics as well as other analysis techniques related to a use-case driven development process.

To grade the pre- and post-tests we employed three judges: a professor with experience teaching the course, but not involved in either section (Judge 1), the instructor for the control group (Judge 2), the instructor for the CCM condition (Judge 3). 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. The rubric included specific point values associated with various plausible answers. Below is an example pre-test question with its corresponding point system:

---

What is an object (instance) and what is a class? **(4 points)**

- A class is an abstraction that encapsulates data and its related operations. It is used as a blueprint to create an instance of itself called an object. **(4 points)**
- A class implements a specification used for creating objects that share the same attributes and behavior. **(4 points)**
- A class is a blueprint for an object and an object is an instance of a class. **(3 points)**
- A class is used for creating objects. **(2 points)**
- Class and object are object-oriented concepts. **(1 point)**

---

## Data Analysis and Results

To determine if the condition group saw greater improvement, on average, than the control group we used a T-test statistic on the differences between the Post- and Pre-test scores for each group. The individual scores by each judge, for each student in each group, are shown in Table 1and Table 2.

**Table 1: Pre- and Post-test scores (%) for each student in the Control group.**

| Student | Judge 1 Pre | Judge 3 Pre | Average Pre | Judge 1 Post | Judge 3 Post | Average Post | Post-Pre |
|---|---|---|---|---|---|---|---|
| 1 | 21.9 | 40.6 | 31.3 | 80.2 | 70.9 | 75.6 | 44.3 |
| 2 | 21.9 | 25.0 | 23.4 | 57.0 | 59.3 | 58.1 | 34.7 |
| 3 | 0.0 | 3.1 | 1.6 | 80.2 | 76.7 | 78.5 | 76.9 |
| 4 | 43.8 | 53.1 | 48.4 | 77.9 | 75.6 | 76.7 | 28.3 |
| 5 | 28.1 | 71.9 | 50.0 | 80.2 | 83.7 | 82.0 | 32.0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 25.0 | 68.8 | 46.9 | 54.7 | 57.0 | 55.8 | 8.9 |
| 7 | 9.4 | 31.3 | 20.3 | 34.9 | 41.9 | 38.4 | 18.1 |
| 8 | 12.5 | 37.5 | 25.0 | 53.5 | 62.2 | 57.8 | 32.8 |
| 9 | 25.0 | 43.8 | 34.4 | 72.1 | 73.3 | 72.7 | 38.3 |
| 10 | 65.6 | 59.4 | 62.5 | 83.7 | 83.7 | 83.7 | 21.2 |
| 11 | 53.1 | 75.0 | 64.1 | 89.5 | 85.5 | 87.5 | 23.4 |
| 12 | 46.9 | 90.6 | 68.8 | 61.6 | 59.9 | 60.8 | -8.0 |
| 13 | 31.3 | 43.8 | 37.5 | 73.3 | 75.6 | 74.4 | 36.9 |
| 14 | 18.8 | 28.1 | 23.4 | 83.7 | 76.7 | 80.2 | 56.8 |
| 15 | 12.5 | 18.8 | 15.6 | 81.4 | 77.9 | 79.7 | 64.0 |
| 16 | 31.3 | 68.8 | 50.0 | 73.3 | 72.1 | 72.7 | 22.7 |
| 17 | 21.9 | 43.8 | 32.8 | 79.1 | 79.7 | 79.4 | 46.5 |
| 18 | 15.6 | 37.5 | 26.6 | 77.9 | 82.0 | 79.9 | 53.4 |

**Table 2: Pre- and Post-test scores (%) for each student in the Condition group.**

| Student | Judge 1 Pre | Judge 2 Pre | Average Pre | Judge 1 Post | Judge 2 Post | Average Post | Post-Pre |
|---|---|---|---|---|---|---|---|
| A | 21.9 | 21.9 | 21.9 | 48.8 | 55.8 | 52.3 | 30.5 |
| B | 9.4 | 34.4 | 21.9 | 47.7 | 38.4 | 43.0 | 21.1 |
| C | 18.8 | 40.6 | 29.7 | 74.4 | 66.3 | 70.3 | 40.7 |
| D | 12.5 | 25.0 | 18.8 | 69.8 | 80.2 | 75.0 | 56.3 |
| E | 28.1 | 53.1 | 40.6 | 73.3 | 84.9 | 79.1 | 38.4 |
| F | 9.4 | 25.0 | 17.2 | 72.1 | 74.4 | 73.3 | 56.1 |
| G | 25.0 | 31.3 | 28.1 | 77.9 | 84.9 | 81.4 | 53.3 |
| H | 21.9 | 37.5 | 29.7 | 77.9 | 83.7 | 80.8 | 51.1 |
| I | 15.6 | 34.4 | 25.0 | 47.7 | 51.2 | 49.4 | 24.4 |
| J | 53.1 | 65.6 | 59.4 | 79.1 | 93.0 | 86.0 | 26.7 |
| K | 43.8 | 62.5 | 53.1 | 70.9 | 73.3 | 72.1 | 19.0 |
| L | 34.4 | 59.4 | 46.9 | 53.5 | 48.8 | 51.2 | 4.3 |
| M | 3.1 | 21.9 | 12.5 | 41.9 | 38.4 | 40.1 | 27.6 |
| N | 28.1 | 53.1 | 40.6 | 72.1 | 86.0 | 79.1 | 38.4 |
| O | 28.1 | 37.5 | 32.8 | 88.4 | 90.7 | 89.5 | 56.7 |
| P | 50.0 | 71.9 | 60.9 | 86.0 | 87.2 | 86.6 | 25.7 |
| Q | 34.4 | 59.4 | 46.9 | 69.8 | 70.9 | 70.3 | 23.5 |
| R | 34.4 | 71.9 | 53.1 | 84.9 | 89.5 | 87.2 | 34.1 |
| S | 43.8 | 59.4 | 51.6 | 68.6 | 77.9 | 73.3 | 21.7 |
| T | 25.0 | 34.4 | 29.7 | 87.2 | 86.0 | 86.6 | 56.9 |
| U | 12.5 | 34.4 | 23.4 | 77.9 | 80.2 | 79.1 | 55.6 |

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 3 and 4 show the comparison statistics of the evaluations from the two judges who assessed the CCM condition pre- and post-tests. Tables 5 and 6 show the comparison statistics of the evaluations from the two judges who assessed the control groups assessments.

**Table 3: Inter-rater reliability for Pre-test in the Condition group.**

| | Judge 1 | Judge 2 |
|---|---|---|
| T = -3.84 | μ = 26.3 | μ = 44.5 |
| p = 0.0 | σ = 13.8 | σ =16.7 |

**Table 4: Inter-rater reliability for the Post-test in the Condition group.**

| | Judge 1 | Judge 2 |
|---|---|---|
| T = -.71 | μ = 70.0 | μ = 73.4 |
| p = .481 | σ = 14.0 | σ = 17.1 |

**Table 5: Inter-rater reliability for the Pre-test in the Control group.**

| | Judge 1 | Judge 2 |
|---|---|---|
| T= -3.02 | 26.9 | 46.7 |
| p = 0.005 | σ = 16.5 | σ = 22.4 |

**Table 6: Inter-rater reliability for Post-test in the Control group.**

| | Judge 1 | Judge 2 |
|---|---|---|
| T = 0.01 | μ = 71.9 | μ = 71.9 |
| p = 0.994 | σ = 14.0 | σ = 11.5 |

While the post-test results for both the condition and control groups were reliable, (Tables Table 4 and Table 6) with no significant difference between the judging (p=.481 and p=.994), the pre-tests (Table 3 and Table 5) show a significant difference between the judges scores (p=0.0 and p=.05). 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 7.

**Table 7: T-test of the difference between Post- and Pre-test results for the condition and control groups.**

| | | CCM (n=21) | Control (n=18) |
|---|---|---|---|
| Judge 1 | T = -0.25<br>p = 0.599 | μ = 43.6<br>σ = 14.5 | μ = 45<br>σ = 17.9 |
| Judge 2 (CCM) vs Judge 3 (control) | T = 0.53<br>p = 0.3 | μ = 28.9<br>σ = 18.7 | μ = 25.2<br>σ = 24.6 |
| Average | T = 0.21<br>p = 0.419 | μ = 36.3<br>σ = 15.7 | μ = 35.1<br>σ = 20.3 |

These results reveal that Judge 1 found that the control group marginally outperformed the condition group, while a comparison of the two judges involved in teaching the respective sections reveal marginally better performance from the condition group. Neither result was significant however (p=0.599 and p=0.3, respectively), and the overall average of all judges was also not significant (p=0.419) and thus the hypothesis, H4, that use of the CCM, and therefore effective teamwork, will facilitate improved individual learning is not confirmed. A valid question to ask here is whether the experiment design was flawed; that perhaps the post-test assessment was not sufficiently demanding to discriminate between high and low performing students. Looking at the average percentages for the post-test scores (71.7 for CCM, 71.9 for control), however, reveals that there was still significant margin for improvement in the CCM group had their learning been improved. Thus we will explore alternative explanations in the proceeding discussion.

**Discussion**

Clearly we anticipated that the guidelines and practices that we have found to improve team effectiveness would also lead to improved learning, and that hypothesis was not supported. Reflecting upon the result there may be several explanatory factors. First, the work of the teams on their projects may reflect operational competency rather than deep understanding of the course content. For example, despite the relatively large scale of the projects tackled, and their relative complexity, the solutions exercise only a small cross-section of the concepts and theoretical foundations covered in the class, which is partly why multiple assessment techniques are employed in this course.

Furthermore, while the CCM guidelines encourage the team members to work collaboratively rather than cooperatively, we do still find that students eventually experience role specialization where each team member takes on primary responsibility for some aspect of the project[21], and therefore becomes relatively expert in only a limited portion of the course content.

Finally, the issue raised earlier in the paper regarding cognitive, and therefore learning, preferences may be critical. If engineers are more prone to be field independent learners, any learning modes oriented towards collaborative working may run counter to the preferences of the individual, and this would raise questions regarding the use of social constructivist and collaborative experiential learning approaches in engineering education.

## Conclusions and Future Work

The goal of this study was to determine the impact of an individual on an effective team in a learning environment. Previous studies investigated the factors for team success as well as the degree by which those factors facilitated team cognition. The results of this study showed that despite team learning and improved project outcomes, the individuals with access to the CCM did not learn any more effectively than those without it.

Thus, our future work will focus on determining the factors to facilitate both team success and individual learning. Of particular interest is the efficacy of collaborative learning approaches in general for engineering students. We remain committed to these approaches and question whether team projects, which exercise operational effectiveness rather than collaborative learning, per se, should really be considered a learning experience at all, and instead simply summative assessment of the ability of students to apply an admittedly limited portion of the course content.

### Bibliography


1. DeFranco, J.F. Collaborative Problem Solving and Program Development Model, Ph.D. Dissertation, New Jersey Institute of Technology, 2002
2. DeFranco, J.F., and C.J. Neill. “Improving Learning Outcomes using Cognitive Models in Systems Design,” Proc. ASEE Annual Conference, Austin, TX, June 2009
3. DeFranco, J.F., and C.J. Neill. “Improving Team Performance: The Cognitive Style Factor,” Proc. ASEE Annual Conference, Louisville, KY, 2010.
4. DeFranco, J.F., C.J. Neill, and R.B. Clariana. “A Cognitive Collaborative Model To Improve Performance in Engineering Teams – A Study of Team Outcomes and Mental Model Sharing”, *Systems Engineering Journal*, Volume 14, No 3, 2011.
5. Neill, C.J., and J.F. DeFranco. “Improving Team Learning in Systems Design”, Proc. ASEE Annual Conference, Vancouver, CA, 2011.
6. Felder, R.M., R. Brent, T.K. Miller, C.E. Brawner, and R.H. Allen. "Faculty teaching practices and perceptions of institutional attitudes toward teaching at eight engineering schools," in Proc. 1998 FIE Conf., Tempe, AZ, 1998, pp. 101-105.
7. Greco, E., and J. Reasoner. “Student Laboratory Skills and Knowledge Improved through Individual Lab Participation,” Proc. ASEE Annual Conference, Louisville, KY, 2010.
8. Witkin, H.A., and D.R. Goodenough, “Field Dependence and Interpersonal Behavior,” Psychological Bulletin, vol. 84, no. 4, 1977, pp. 661–689
9. Witkin, H.A., and D.R. Goodenough, Cognitive Styles: Essence and Origins, International Universities Press, Inc., NY, 1981.
10. Todd, R. H., S. P. Magleby, C. D. Sorensen, B. R. Swan and D. K Anthony, “A Survey of Capstone Engineering Courses in North America,” Journal of Engineering Education, vol. 84, no. 2, 1995, pp. 165-174

11. Newell, S, “Collaborative Learning in Engineering Design,” Journal of College Science Teaching, vol. 19, no. 6, 1990, pp. 359-362.
12. Gabriele, G. A., L. T. McCloskey, and J. A. Watson, “Guidelines for Forming and Building Student Design Teams,” Proceedings, Advances in Capstone Education Conference, Brigham Young University, 1994, pp. 121-125.
13. Barrett, G.V., and C.L. Thornton, “Cognitive Style Differences Between Engineers and College Students,” Perceptual and Motor Skills, vol. 25, 1967, pp. 789–793.
14. Seat, E., and S.M. Lord, “Enabling Effective Engineering Teams: A Program for Teaching Interaction Skills,” Journal of Engineering Education, vol. 88, no. 4 , October 1999, pp. 385-390.
15. Purzer, S. “The Relationship Between Team Discourse, Self-Efficacy, and Individual Achievement: A Sequential Mixed-Methods Study,” Journal of Engineering Education, vol. 100, no. 4, October, 2011, pp. 655–679.
16. Kittleson, J. M., & Southerland, S. A. (2004). The role of discourse in group knowledge construction: A case study of engineering students. Journal of Research in Science Teaching, 41(3), 267–293.
17. 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.
18. 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.
19. Chi, M. T. H. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. Topics in Cognitive Science, 1, 73–105.
20. Kirton, M., Adaption-Innovation: In the context of Diversity and Change, Routledge 2006.
21. L.L. Levesque, J.M. Wilson, and D.R. Wholey. (2001) Cognitive Divergence and Shared Mental Models in Software Development Project Teams. Journal of Organizational Behavior, Vol. 22, No. 2, March, 2001, pp. 135-144