# INDUSTRY-INSPIRED EXPERIENTIAL LEARNING AND ASSESSMENT OF TEAMWORK

**Flex TIO**

School of Engineering, Nanyang Polytechnic, Singapore

## ABSTRACT

In preparing students for work, final year students studying the Diploma in Multimedia and Infocomm Technology at Nanyang Polytechnic, Singapore, are not only trained to hone their technical skills in software development, but also on how to use those skills as described in the CDIO Syllabus (2.4, 3.1 and 3.2).

This paper presents a series of industry-inspired experiential classroom activities that provide opportunities for students to work effectively in a team. These activities, accompanied by instructional scaffolding, comprise guidelines on dealing with team-based communication, setting and expecting professional behaviour from teammates, to an assessment method that uses a set of criteria commonly adopted by the industry.

Through these activities, students learn about interpersonal skills and leadership traits that are expected of them. Students also get to experience first-hand how the technical skills they have acquired such as software designing and collaboration tools can serve to augment more effective teamwork in software development. Finally, students are also given two in-class assessments, one in which they are required to work under pressure and observation, and the other requires the entire class to complete a single large-scale project which is guided by lecturers. Through self-assessments, peer assessments and feedback from the observations, students are able to reflect on how well they and their classmates have done, giving them valuable insights to how well they can perform in a team.

This paper shares the feedback received from students and their perceptions on the usefulness of the different activities towards their learning.  Reflections on the strengths, areas for improvement in the approach and the future course of action to enhance students' learning are also shared.

## KEYWORDS

Assessment, Experiential Learning, Rubrics, Software Engineering, Teamwork, CDIO Standards 7 and 8

*Note: In the context of Nanyang Polytechnic, the term 'course' refers to a 'program' while the term 'module' refers to a 'course'. For example, Diploma in Multimedia and Infocomm Technology is a course, Java Enterprise Development is a module.*

## INTRODUCTION

Programming enterprise software solutions usually requires programmers to work together in teams. Students need to be armed with both hard skills and soft skills to work in real-life projects. Hence, soft skills such as teamwork and communication are also important (Figl, 2010).

A set of experiential activities to learn about communication and teamwork soft skills is designed for final year students from the Infocomm Solutions specialisation of the Diploma in Multimedia and Infocomm Technology, Nanyang Polytechnic, Singapore, who are expected to learn about programming enterprise software solution. This set of activities is described to be *"industry-inspired"*. This means that students are informed of the activities' meaning and relation to real-life work as they undertake these activities.

This paper shares our experiences in using the industry-inspired activities in the Java Enterprise Development module to coach and assess students in personal attitude, teamwork and communication, mapping closely to the CDIO Syllabus 2.4, 3.1 and 3.2 (Crawley, et al., 2011).

This paper also shares the findings from 97 responses of an anonymous survey conducted across three semesters focusing on the small group collaborative project and interviews made with 10 students focusing on the large group collaborative project, on whether these activities are useful and meaningful from the perspectives of the students.

## RUBRICS AT THE HEART OF LEARNING

In our previous study (Tio, et al., 2014), we found that instrumental usage of rubrics for both coaching and assessment could guide students towards achieving better results and train them in acquiring the module learning outcomes. Similarly, in this study, students were briefed on the interpretation and usage of the rubrics, in terms of expected personal attitude and communications they could practice for effective teamwork. Finally, students made use of this set of rubrics for both self-assessment and peer assessment for their collaborative assignments.

The rubrics used were adapted from the Capacity, Achievement and Relationship (CAR) selection criteria and appraisal framework developed by Shell[1], in the spirit of an appraisal aligned with the industry. In particular, only the Achievement and Relationship aspects of the framework were used as guideposts for the personal attitude, and teamwork and communication aspects respectively.

For the rubrics to be used easily by students in their self-assessments and peer assessments, students were asked to give an overall grading for themselves for the collaboration assignments. As Arter (2000) pointed out, a holistic rubric allowed for quick scoring and snapshot and an analytic rubric was more useful for complicated skills. Our rubrics were hence designed with an additional analytic component with behavioural

[1] "FAQ – Students & Graduates", Retrieved from http://www.shell.com/careers/contact-careers-and-faqs/faqs-students-graduates.html

indicators to take advantage of the benefits of using the analytic rubrics. Table 1 and Table 2 show the details of the holistic and analytic rubrics.

Table 1. Holistic rubric on level of competencies and their definitions

| **Level of Competencies** | **Definition** |
|---|---|
| Far Exceeding | Consistently goes above and beyond his own duties – carries more than his/her fair share of the load. |
| Exceeding | Does what he/she is supposed to do sometimes going beyond his/her duties, very well-prepared and cooperative. |
| Competent | Does what he/she is supposed to do, acceptably prepared and cooperative. |
| Developing | Does what he/she is supposed to do to a limited extent, minimally prepared and cooperative. |
| Not Meeting | Consistently fails to complete his/her share of the project, unprepared. |

Table 2. Analytic rubric on expected behavioural indicators

| | **Level of Competencies** | **Competent** | **Developing** | **Not Meeting** |
|---|---|---|---|---|
| **Behavioral Indicators** | **Personal Attitude**<br>Models the teamwork aspect of the “Achievement” indicator in the Shell CAR framework | Stays on tasks all the time without reminder<br>Searches for solution actively<br>Takes initiative to help others or asks others for help | Needs reminders from group members to do the work<br>Is easily distracted<br>Pretends to be busy<br>Comes late for assessment | Selfish actions<br>Works alone even when faced with difficulties that cannot be solved on his/her own<br>Comes very late for assessment |
| | **Teamwork and Communication**<br>Models the “Relationship” indicator in the Shell CAR framework. | Is Open<br>Is Trusting<br>Is Supportive<br>Is Respectful | Disrupts the flow of discussion<br>Communicates negative talks (eg. Complains and arguments) | Ignores or ridicules others<br>Feuds with other team members |

A key feature of this set of rubrics was the focus on individual, how one could respond, communicate and perform to contribute in a team. The scores could be assigned to an individual student based on how well he or she was able to work in a team. Indeed, interviewed students shared that individual grading was a fairer way of assessment compared to a common score assigned to the entire team. In addition, out of 97 students who participated in the survey across three semesters, 97% find the rubrics to be useful and they helped students to be aware of the expectations when working in a team-based environment (see Figure 1).

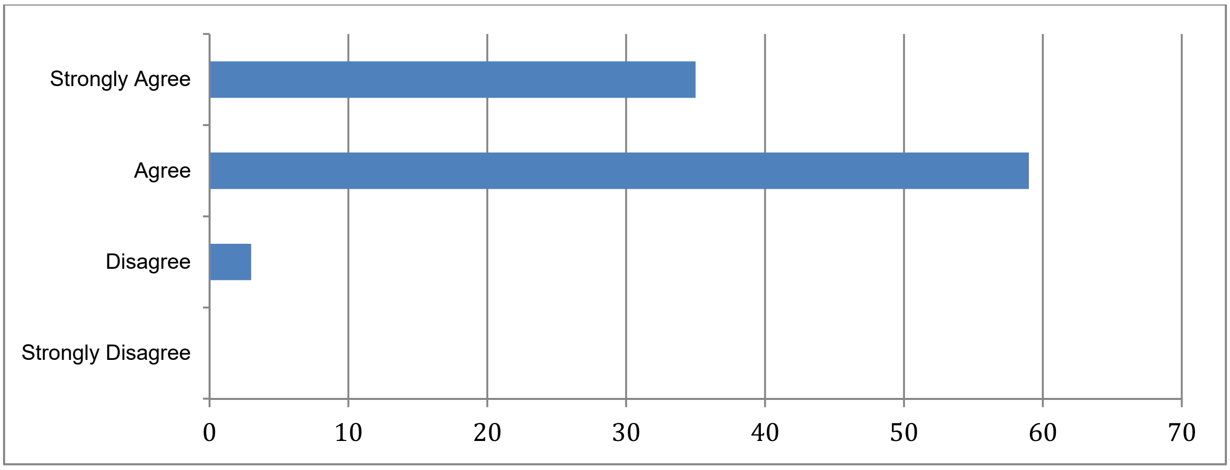

Figure 1. Student responses on the usefulness of the industry-inspired rubric developed (n=97).

## INDUSTRY-INSPIRED EXPERIENTIAL LEARNING AND ASSESSMENT

In this study, students participated in three industry-inspired activities:

1. Pair Programming,
2. Small Group Collaborative Assignment, and
3. Large Group Collaborative Assignment

### *Pair Programming*

First, students practiced how to communicate and work with each other in pairs. Pair programming, an agile software development technique where two programmers worked together on one workstation, was used for this purpose. Research (Faja, 2013) suggested that pair programming improved student engagement, student learning outcomes and perceived learning. However, these were largely related to technical learning outcomes.

For the Java Enterprise Development module, pair programming was used as an introduction to help students understand the attitude and communication required for a professional and productive working relationship. Students were given guidelines on the best practices of pair programming and they were asked explicitly to practise them during their class assignments. These best practices largely included how communication was to be carried out in a pair programming session and how to (and not to) communicate with each other. 94% of the students agreed that pair programming helped to prepare them to communicate and program effectively in a team (see Figure 2).

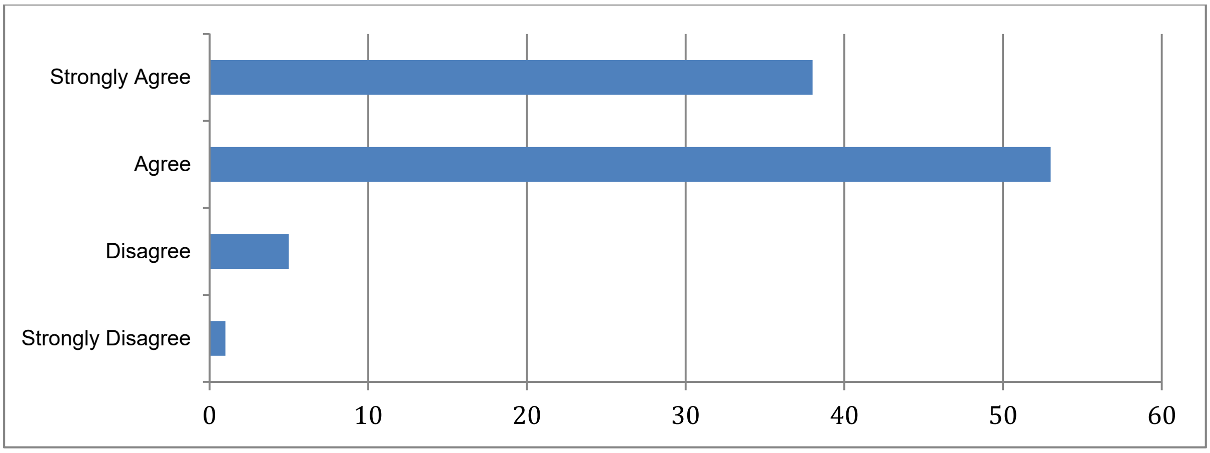

Figure 2. Student responses on the effectiveness of pair programming in learning about communication and working as a team (n=97).

### *Small Group Collaborative Assignment*

Following the pair programming practice, students participated in a small group collaborative assignment. This comprised five phases:

1. Group Formation
2. Choice of Leadership
3. Small Group Collaborative Assignment and Observations
4. Peer Appraisal and Self Appraisal
5. Debrief and Discussion

Three weeks before the assignment, students were assigned in groups of five to seven and each group was asked to choose a technical lead. The actual small group collaborative assignment was done in-class so that the instructor could observe the students at work. At the end of the session, students were asked to reflect and give an appraisal to themselves and fellow group members. Finally, a debrief and discussion session between the instructor and the students regarding the exercise was held.

#### *Group Formation*

The assignment of students to groups was done by the instructor of the module according to two criteria. First, students who seldom worked together were placed in the same team. The main aim was to model the relationship between colleagues in real work-life, where people often did not have a choice over who they would be working with. Second, students with varying technical abilities with at least one student who was more competent in programming were placed in the same group. This was to ensure that all groups could be self-sufficient at completing the assignment.

The aim of grouping students who were not familiar with each other together in a group was to reduce friendship bias that may affect the reliability of the peer appraisal, noting that prior research showed that peer ratings could be both valid and reliable even with the presence of friendship bias (Love, 1981).

One significant difference was the decision to put five to seven students in a group, as opposed to what research suggested to keep team sizes up to four (Figl, 2010). The main aim was to allow a more complex assignment to be set for the students at a scale which students had not seen before in class, in an attempt to reflect the scale of enterprise projects. Students could also get a chance to work with more people who may have different working styles they were not familiar with. 96% of the students agreed that such grouping process was useful as they experienced what it was like to work with different types of people and some of whom they had not had a chance to work with before (see Figure 3).

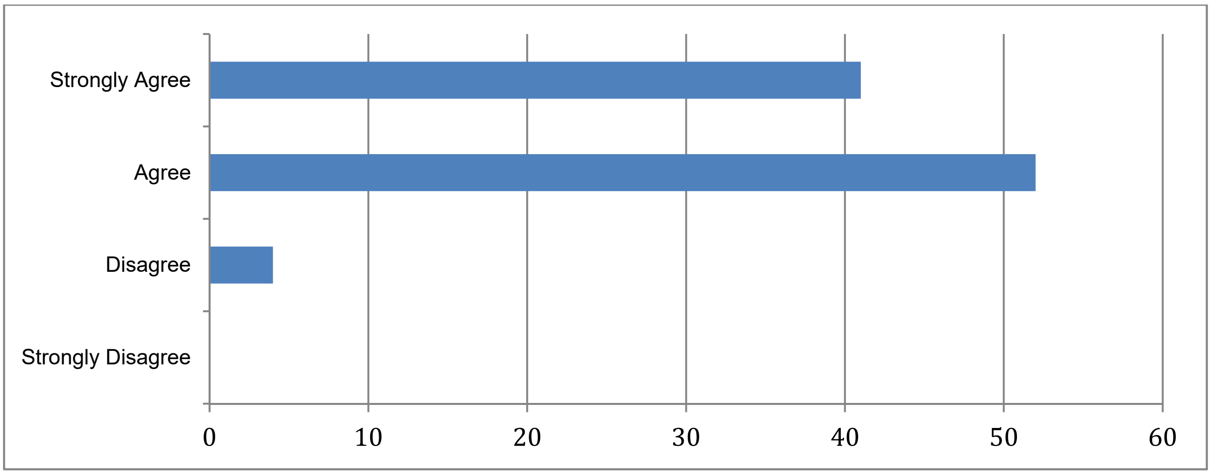

Figure 3. Student responses on the usefulness of the experience working with other students whom they seldom worked with (n=97).

*Choice of Leadership*

Students were also asked to select a technical lead for the group. They were first given the scope of work of the technical lead. They then individually selected a member from their group to be the technical lead with reasons for their choice. The reasons were analysed and compiled as a report presented to the class by the instructor.

This activity served two purposes. The first was to crowd source knowledge from students and let them become aware of the different reasons that their friends used to select a leader (as opposed to lecturer quoting theories on what qualities leaders should possess). The second purpose was for the students who received at least one vote as the leader to be aware of what their friends thought about them.

An observation regarding the students' choices was that they were collectively capable of selecting a suitable leader among themselves. They were able to identify the need for the technical lead to possess both hard and soft skills in order to lead them effectively. This was despite the fact that they may not know the team members very well due to the way they were grouped. 99% of the students agreed that the activity allowed them to understand what people usually looked for in a leader and what it took to become a leader (see Figure 4).

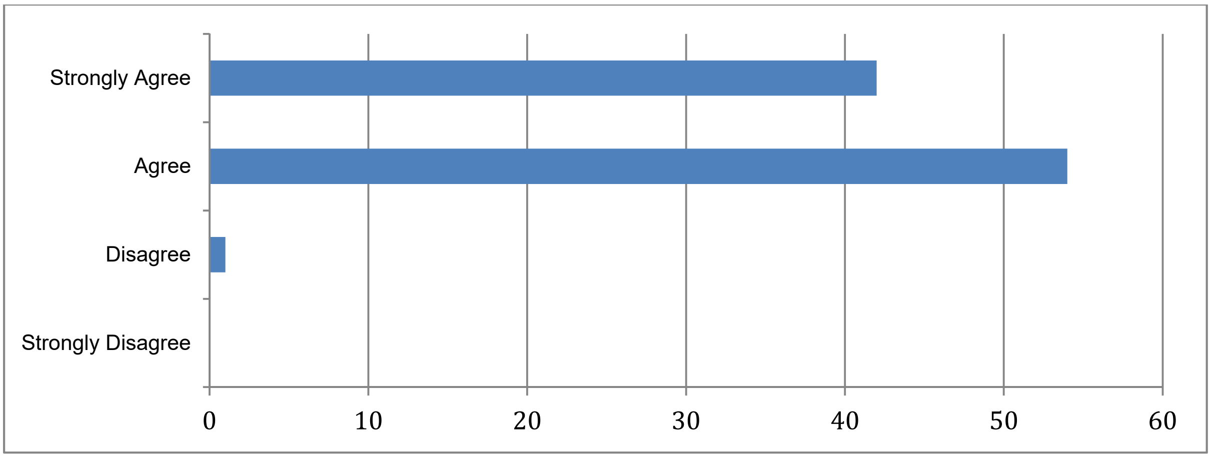

Figure 4. Student responses on the usefulness of the Choice of Leadership activity (n=97).

*Small Group Collaborative Assignment and Observations*

Students received their assignment on the day of the in-class observation. They were given the guidelines regarding roles assignment and time management to help them manage the assignment that was to be done within two hours. Students were expected to solve any issues they encountered along the way on their own without receiving any technical assistance from the instructor. While students worked on their in-class assignment, the instructor observed the students, gathering both positive behaviours demonstrated and areas for improvement according to the guidelines set out by the rubrics. These observations were shared with the students during the debrief session held after the completion of the small group collaborative assignment.

A member of staff who was invited to one of the sessions to observe the small group collaborative assignment in action noted that “the lab was conducted as a simulation of real-life project development” and “there were active discussion, consultation and communication among team members during coding”.

*Peer Appraisal and Self Appraisal*

After the completion of the in-class assignment, students were asked to do an appraisal for themselves and their peers on how they performed in terms of personal attitude and communication based on the rubrics. The word ‘appraisal’ was used in place of the word ‘assessment’ intentionally to emphasise the similarities between a work performance appraisal exercise in the industry and assessments in school.

In addition to grading each other using the rubrics as described in the earlier section of the paper, students were also asked to give reasons to the grades they assigned for themselves and their peers. These reasons could aid the instructor to review the students’ understanding of the attitude and communication required for teamwork.

*Debrief and Discussion*

A debrief and discussion session was conducted immediately after their appraisals, aiming to provide immediate feedback to the students. The instructor reported his observations on the students’ personal attitude, teamwork and communication displayed during the course of the assignment. This included positive points, areas for improvements as well as possible blind spots that students never noticed before. Students were also given the chance to share their thoughts and feedback through this open forum. 95% of the students found the debrief and discussion session to be useful as they not only learnt about their experiences working in a team, they also learnt about their blind spots (see Figure 5).

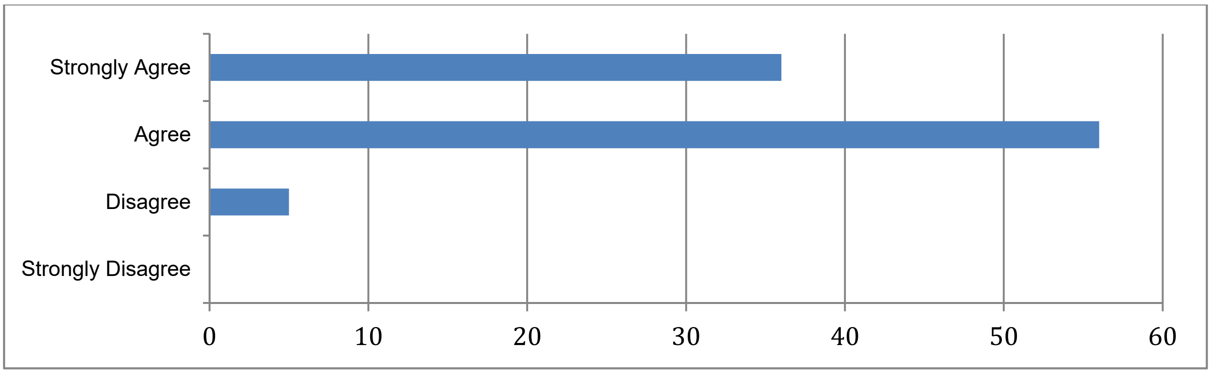

Figure 5. Student responses on whether the debrief session was useful for them (n=97).

### Large Group Collaborative Assignment

The aim of the large group collaborative assignment was to give students additional experience at collaboration work, and to simulate a large scale project handled by a software development house. The assignment, involving the entire class (between 15 to 24 students), comprised several modules to be implemented in parallel by multiple sub teams. Students were given the choice of different roles such as chief technical lead, technical lead, assistant technical lead and team members and were briefed on how to communicate with each other in a large group. The Subversion system, a collaboration tool commonly used in the software engineering domain was also introduced at this point. This was to allow students to feel the difference between developing a project with and without a collaboration tool.

To ascertain if the assignment was useful and meaningful to the students, ten students from across two of the most recent batches were interviewed. There were some differences in the way the assignment was carried out between the two batches; the 2016 batch received some tweaks in the way the assignment was carried out, based on student feedback and suggestions from the 2015 batch. These differences are shown in Table 3.

Table 3. Differences in the manner in which the assignment was carried out

| | **2015 Batch** | **2016 Batch** |
|---|---|---|
| Duration | 3 hours | 6 hours (across two weeks) |
| Guidance on Method of Communication | Simple Guidelines with briefing. | Comprehensive Guidelines with briefing. Printed guidelines are given for reference. |
| Training on Subversion System | "On the job" training with pre-briefing given during the assignment. | "On the job" training with pre-briefing given before the assignment. Tutorial sheet for reference. |

*Findings from the Interviews*

In both batches, students understood the intention of the assignment. However, one of the major gripes experienced by the 2015 batch was insufficient time to complete the assignments. This was identified as a major cause for their lack of appreciation over other aspects that the assignment entailed as they were committed to complete the project within the short timeline that was allocated to them. Even though communication guidelines and training on the Subversion system were given to them, students from the 2015 batch felt that communication was "messy" and "haphazard", faced "difficulties in understanding" how the Subversion system worked. Eventually this also led to the "lack of sense of achievement" in the assignment which majority of the students interviewed felt was an important factor that affected their perceived learning.

Based on the suggestions made by the students from the 2015 batch, the duration of the assignment was lengthened to 6 hours and printed notes on communication guidelines and usage of the Subversion system were given to the students in the 2016 batch. With such changes implemented, students agreed that they felt a sense of achievement working through this assignment. They also felt that the earlier activities such as pair programming and the small group collaboration assignment helped them build rapport with their peers, making it easier for them to communicate with the others.

One interesting thing to highlight regarding communication was that while the students reported that they found that the communication guidelines very useful in giving them a starting point on how to communicate with one another, they were also able to develop their own style of communication in the midst of the assignment that worked better for them.

With regard to the training on using the Subversion system, the 2016 batch felt that more could be done. For instance, giving them a dry run on the usage of the Subversion system before the assignment would help them get familiar with the tool. One student also highlighted that the large group collaborative assignment was similar to the experience he had during his internship attachment period as he was asked to use a Subversion system during his stint. Hence, he felt that the Subversion system was a very important tool for all students in his specialisation to learn.

## DISCUSSION

The industry-inspired experiential activities that were described in this paper showed a way to integrate teamwork into a module. Students who took the module were generally receptive and they valued the experience that was given to them.

There are a few learning points from this study. First, students find the learning activities to be useful when they see activities are strongly contextualised to the domain the students are studying, and the rationales behind undertaking of the activities and its relation to real-life work are explained to them. Second, it is possible to let students work in large groups. Given sufficient guidelines to work with and easing them through a sequence of activities that grow in complexity, students can learn about managing projects and communication meaningfully. Third, teamwork assessment is to be done at an individual level, giving more control over their performance without the fear of their grades being pulled down by their peers. Fourth, it is important to allow time for students to get a sense of achievement through their collaborative projects. While it is useful to experience what it is like working in a group and learn soft skills, a lack of sense of achievement in the technical task, on the other hand, can lower the usefulness and meaningfulness of the activities that they undertake.

Moving forward, we intend to look into how a selection of these activities can be adapted and contextualised for usage in other modules.

## REFERENCES

Arter, J. (2000), “Rubrics, Scoring Guide, and Performance Criteria: Classroom Tools for Assessing and Improving Student Learning”, *Annual Meeting of the American Educational Research Association*, New Orleans.

Crawley, E. F., Malmqvist, J., Lucas, W. A., & Brodeur, D. R. (2011), “The CDIO Syllabus v2.0. An Updated Statement of Goals for Engineering Education”, *7th International CDIO Conference*, Copenhagen.

Faja, S. (2013), “Evaluating Effectiveness of Pair Programming as a Teaching Tool in Programming Courses”, *Proceedings of the Information Systems Educators Conference*, San Antonio, Texas, US.

Figl, K. (2010), “A Systematic Review of Developing Team Competencies in Information Systems Education”, Journal of Information Systems Education*, 21*(3), 323-337.

Love, K.G. (1981), “Comparison of Peer Assessment Methods: Reliability, Validity, Friendship Bias, and User Reaction”, *Journal of Applied Psychology, 66*(4), 451-457.

Tio, F., Kong, J., Lim, R. & Teo, E. (2014), “Developing and Applying Rubrics for Comprehensive Capstone Project Assessment”, *10th International CDIO Conference*, Barcelona.

## BIOGRAPHICAL INFORMATION

**Flex Tio** is a Lecturer in the School of Engineering, teaching software programming and engineering practices. He was a research and development engineer with the Government of Singapore specialising in Visual Analysis and Text Analysis. He is a member of the ICT-Enabled Teaching and Learning and Active Learning Working Committee within the school, exploring education pedagogy involving collaboration and the usage of technology.

***Corresponding author***

Mr Flex Tio
Nanyang Polytechnic
180 Ang Mo Kio Ave 8
Singapore 569830
65-65500542
flex_TIO@nyp.edu.sg