

ASSESSING THE MATURITY OF TEAMWORK CAPABILITIES THROUGH CDIO PROJECTS

Vu T TRUONG, Bao N LE, Man D NGUYEN, Thang M NGUYEN

Duy Tan University, VIETNAM

ABSTRACT

CDIO project is an important part of the general CDIO approach. The project is mostly used to connect relevant knowledge, to address essential requirements, to develop certain soft skills, and to help solve problems in an experimental and collective manner. Above all, the CDIO project is usually considered as the best tool to build and put together teamwork skills, which engineering students usually lack.

In this paper, using data collected during the deployment of CDIO's Capstone Projects in the Software Engineering program of DTU at various stages, we measure, assess and analyze the maturity levels of our students' teamwork capabilities. Our measurement and assessment in this literature is based on an "in-house" rubric, which addresses key aspects of teamwork capabilities and outcomes along five dimensions of (1) Shared Leadership, (2) Team Orientation, (3) Effort Redundancy, (4) Learning Results, and (5) Team's Autonomy. In the scope of our rubric, besides a standard survey, we also hold focus group sessions to collect additional information from students about what they hope to improve on their teamwork capabilities and how they perceive their maturity in teamwork capabilities after each CDIO Capstone Project. Results from this study are of great interest to schools, which are looking for ways to fasten their students' teamwork maturity process and to balance out between the training of "hard" skills and "soft" skills in CDIO adoption.

(226 words)

KEYWORDS

Agile Software Development, CDIO/Capstone Projects, CDIO Standard 5, 6, 7 & 11, Effort Redundancy, Plan-driven Development Process, Teamwork Capabilities, Team Orientation

INTRODUCTION

Although it is commonly recognized by institutions and industries alike that to be successful in today's workplace, a good amount of teamwork is necessary and essential, most engineering programs provide little, if any, real-world instruction and guidance in this area. Employers often report that newly-hired graduates typically do not know how to communicate well and that they have insufficient amount of experience and preparation for the professional workplace. Most engineering tracks, in general, and IT (Information Technology) programs, in particular, still focus on individual contribution rather than on managed-group efforts despite of the fact that such coordinated efforts are becoming more of the norm in most industries (Lingard, 2010).

Engineering is a profession, in which the ability to apply technical principles while remaining effective with personal and interpersonal skills is extremely important. In an article on "Educating a New Engineer", Peter Denning stated "a student must learn not only the technical side of engineering, but also skills of listening, completing and communicating". This is increasingly correct given today's large and complicated engineering projects, in which engineers always have to work in team and hardly any end product is the work of any one individual. So, despite that engineers and IT specialists are traditionally regarded as introverted and independent individuals, it is important for engineering students today to "learn the skills for working effectively as members of groups (or teams)" (Denning, 1992). The Accreditation Board for Engineering and Technology (ABET) has recently emphasized teamwork skills in its criteria for accrediting both computing and engineering programs. Specifically, the engineering guideline states that graduates must demonstrate "an ability to function on multi-disciplinary teams" and the computing guideline requires "an ability to function effectively on teams to accomplish a common goal" (ABET, 2009).

Even though there have been many academic tactics for the teaching of teamwork in engineering and IT, there is no better way of teaching it than under the real-world settings of the industries. However, we do not always afford such a luxury, and so, the question of how to bring the real-world settings to the classroom has become a major one for every educator. The CDIO (Conceive, Design, Implement, Operate) Initiative, with its focus on engineering education improvement, has provided a relatively good answer through the design of its CDIO project. CDIO Standard No. 5 and 6 has described how important it is to include corporate partners from the industry in the CDIO/Capstone projects as well as how systematic the team's settings and workspace for the CDIO/Capstone project should be arranged so as to make the most for the students' learning curve (CDIO™ Initiative, 2010). In general, the CDIO/Capstone project is expected to connect relevant knowledge, to address essential requirements, to develop soft skills, and to help solve engineering problems under a collective and team-based manner.

In the first part of this paper, we will present the purpose, structures, and settings of the CDIO/Capstone project designed for IT disciplines at Duy Tan University using the guidelines and advice from the CDIO Framework with the purpose of maximizing the learning of teamwork capabilities. In the second part, we demonstrate how the maturity levels of our IT students' teamwork capabilities are measured, assessed and analyzed using an "in-house" rubric. The rubric utilizes five different dimensions of (1) shared leadership, (2) team orientation, (3) effort redundancy, (4) learning results, and (5) team's autonomy, which were proposed by Moe et al for the test of teamwork effectiveness in the Agile software development methodology (Moe et al, 2009). Based on Moe et al's five dimensions and other research literature, we basically developed a questionnaire rubric to measure our

students' teamwork capability maturity, adapted only for IT students working on their CDIO/Capstone projects. The purpose of the questionnaire rubric is to assess the effectiveness of our CDIO approach in IT Capstone projects while the reason for choosing the five dimensions of Moe was due to the fact that most of the Capstone teams at DTU are small ones (of 4 to 6 students per team), which are best suited with Agile software development methodology (Rico & Sayani, 2009).

IMPLEMENTATION OF CDIO PROJECT IN SOFTWARE ENGINEERING & IT-RELATED DISCIPLINES AT DTU

In this part, we will describe about the implementation of the CDIO/Capstone project in Software Engineering, Network Security and Information Systems at the Faculty of Information Technology, DTU in an effort to foster quality engineering training and teamwork learning:

A. CDIO/Capstone Project and its Purpose:

Capstone projects are set up for students to solve some practical problem through the integration of many skills and knowledge to develop new ideas, design, and implementation approaches (IEEE Computing Society and Association for Computing Machinery, 2004). For DTU, the main purpose of IT CDIO/Capstone projects is to provide students with a sense of "real-world", "on-the-job", teamwork activities. Using the CDIO Framework, the CDIO/Capstone projects at DTU make use of typical teaching/learning methodologies like active learning, team learning, experimental learning, design-implement experiences, etc. (CDIO™ Initiative, 2010). Another motivation behind the adoption of CDIO projects is support our ABET accreditation effort for IT programs of the Faculty of Information Technology and the International School of DTU.

Specifically, at DTU, there are three CDIO/Capstone projects for students majoring in Software Engineering, Information Systems and Network Security, usually at the second half of each academic year, starting from the sophomore year. Since the Capstone projects for the sophomore and junior year mostly provide an introduction and simulation of the real-world IT industry, we will focus on the CDIO/Capstone project for the senior year. For this final-year CDIO/Capstone project, students in each team have to work with a real-world client to design, develop, and implement an IT product or solution, for a period of 6 to 8 months, i.e., two semesters. The project is actually split into two stages for the first and second semester, which we usually call as Capstone 1 and Capstone 2. Faculty mentors from DTU are also assigned to each CDIO/Capstone team project to ensure smooth communication with the business clients and to support the students as needed (Todd & Magleby, 2005).

B. CDIO Project Structures and Times Frames:

In the final-year CDIO/Capstone project, students will be assigned to teams of 3 to 5 members. Following the guidelines of ACM/IEEE, team member assignment is done by the Faculty of Information Technology, DTU so as to ensure that the skills and capabilities of any one team member will supplement for those of others (ACM, 2001). Each team will then bid for one of the real-world projects from several IT-business clients of Duy Tan University. The IT-business clients of DTU are usually software and IT companies from Da Nang, Hanoi, and Ho Chi Minh City. One or two IT mentors from the Faculty of Information

Technology, DTU will also be assigned to each CDIO/Capstone team project. These faculty mentors are usually the ones who have connected the IT-business clients with DTU.

Of the 6 to 8-month period of the final Capstone project, there are two stages of Capstone 1 and Capstone 2, as mentioned - each lasts for a period of 3 to 4 months. For Capstone 1, each student will need to spend a minimum of 16 hours per week for team interaction, regulation formulation, conflict resolution, software process selection, and proposal write-up. It is expected that Capstone 1 will take up to 192 hours (i.e., 16 hours x 12 weeks) per student in every team, but as soon as the project proposal is approved by the clients, the team will move on to the next stage, which means excellent teams may cut down their Capstone 1 to less than four months. In Capstone 2, students will have to spend time on requirement engineering, product architecture, system design, coding, testing, product implementation and maintenance. At least 40 hours of work per week or a total of around 480 hours will be expected from every student for this stage. In short, students will go through every step of the complete software development life cycle (Moe et al, 2009), and they need to understand what to do in each step for each role in the team. Of course, there may be some deviations depending on the software development methodology the students select, for example, it will be quite different between an Agile approach and a plan-driven structure.

C. Specific Implementation Tasks in the IT CDIO/Capstone Project:

Team Formulation:

One of the first activities in the CDIO/Capstone is the skills-review session in which each member will identify how he or she may best contribute to the project. Together, the team must define the roles and responsibilities for each team member (IEEE Computing Society and Association for Computing Machinery, 2004). A number of staff members from the Faculty of Information Technology will attend the role identification meeting, and may give direction or guidance.

While nobody in the team is "the Boss", a project-management role is needed to help coordinate, plan, and track the team's activities and efforts. The Project Manager (PM) keeps the team focused and makes sure that various tasks get done on time. The PM will also review each member's works to make sure that they meet the proposed technical and quality requirements. The PM is the most important role in any IT CDIO/Capstone project, as a result, the team needs to select the most skilful, persuasive and energetic member for this role. Sometimes, this role can be rotated amongst team members from one development phase to another so that everyone will have a chance to play it (Moe & Dingsøyr, 2008).

Beside the PM roles, there are other roles that team members must also identify and assign such as Software Configuration Manager, Software Quality-Assurance Manager, Testing Manager, etc. Since there are several development phases in an IT product life cycle, team members can take turn switching roles with each other in each phase. Role switching can be sometimes damaging to the team's performance, so, the team is required to consult with their mentor(s) any time they do this.

Creation of Product Ideas and Project Structures:

In this step, IT-business clients will hold a series of presentations to introduce their projects and related problems and/or solution suggestions with all the Capstone teams at DTU.

Based on their capacity and capability, each team will bid for the most suitable project. After the bidding process, each team will be assigned one or two mentors for their project. The team will actively plan for meetings to design, develop and implement the selected project. Even though the project idea may not be original because it may have been presented by the business clients from the first place, students are encouraged to develop original ideas in their team formation or technology application for the project at hand.

Design & Implementation:

Before any actual design and implementation activities start, successful team formation and continuous maintenance of the team's effectiveness later on are arguably the biggest challenge to most students in the CDIO/Capstone projects. From the beginning, team members do not have the right to select one another. Then, they have to join force together for the most important project before graduation. Hence, members need to put aside their differences, their interests, and their ego so as to develop a strong team for the project. It is important for students to understand that the Capstone team is "larger" than themselves and every team member must jointly contribute to the success of the project (ACM, 2001). Faculty mentors and business clients will continuously provide students with inputs about the expectations and goals of project, and students need to adjust themselves and their team, accordingly. Of course, all of these come with great challenges, and by overcoming those challenges, students will be able to move up to higher level of maturity in terms of their teamwork capabilities.

Once the team and its project are in place, team members need to actively assess their capabilities and the feasibility of the project in order to set up an effective project plan. Even though the project plan is usually put together by the PM, it is important that the plan be discussed and agreed on by all the team members to ensure their commitment for the project (Lingard, 2010). Depending on the software development process selected, the PM will build detailed plans which are consistent with all the criteria and requirements of that software development process. Later on, especially before each development phase of the software development process, the PM needs to share all information about the actual number of hours that each team member will have to spend for their tasks. This will create a sense of fairness and team members can also plan for their next tasks after that.

As the design work goes on, the team needs to discuss with its business clients from time to time about the desired scope of the project. When the project scope is finally approved upon, the team will need to revise its Software Requirements Specification (SRS) to fit with the approved scope. They may also have to revise the use cases and use-case diagrams and narratives designed previously as well as to fill in missing details and to indicate all the preconditions, post-conditions and triggers for each use case. Once all of these are done, the Design phase is basically completed. Design work, however, does not stop here: depending on the problems and risks encountered later, the PM may decide to go back and revise the design accordingly.

In the process of project Implementation, many tasks of coding, testing and redesigning will be carried out in sequence or at the same time in a complex manner. As a result, the team will need to stick together, to identify dependencies among their tasks, to foster clear communication, and to jointly sort out any problem that they may run into so as to ensure the quality and success of the end product or prototype. Besides the end product or product prototype, Capstone teams at DTU are also required to write up an overall report of their accomplishments, conclusions, recommendations and lessons from the CDIO/Capstone

project. This report must be clear, complete, well-written, and delivered on time. It will serve as the basis for final assessment as mentioned in the next section.

D. CDIO/Capstone Project Assessment:

After the final review with faculty mentors, Capstone teams will need to make a presentation to the Board of Capstone Assessment which usually consists of 5 members including the faculty mentor(s), the business client(s) and other DTU instructors (ACM, 2001). Each team will have a maximum of 30 minutes for their presentation, and the Board of Capstone Assessment will spend at most one and a half hour for Q&A (Questions & Answers) and evaluation of the project outcomes. Questions will be given to specific team members regarding their roles and contribution to the project. At this point, it is usually easy for the Board of Capstone Assessment to identify the strengths and weaknesses of each team and its members as well as to evaluate the level of collaboration and cohesion amongst the team members.

Besides the presentation, Capstone teams are also graded based on the team's final report and the end product or prototype, which reflect all the effort and progress of the Capstone team throughout the CDIO final-year project. As mentioned earlier, the final report has to be of good quality. Good reports are the result of great teamwork and process, supported by the strong performance of the end product or prototype. While students may cheat the audience with a well-written report, it will be difficult for them to disguise ineffective team behavior, lack of progress, and poor-quality product in their presentation. The Board of Capstone Assessment will give out the final grades for each team member based on a combined evaluation of the final report, the board presentation and the end product or prototype. Specific criteria used in the grading are, as followed: Product Quality (40%), Software Development Process Deployment (30%), and Teamwork & Team Management (30%). It should be noted here that even though our focus of Teamwork & Team Management accounts for only 30% of the total grade points, its measurement is also included in the Product Quality and Software Development Process Deployment.

E. Mentoring and Training Teamwork Skills in CDIO/Capstone Project:

While certain Software Development Process is always used as the major tool to monitor the working progress of each CDIO team in IT, in reality, we usually have to add in a great number of mentoring and training efforts and practices to ensure that the students are going down the right path with their projects. Specifically, mentoring and training work is very much needed to constantly help (1) Clarify team members' roles and processes, (2) Build better oral and verbal skills, (3) Develop outcome-oriented approach in students' solution development, (4) Resolve and prevent ongoing or potential conflicts in teams. (Welch, 2000)

Clarifying team members' roles and processes: Even though the roles of team members were well-defined from the beginning, we have run into many situations, in which one team member might take on the work of another, and conflicts emerged as a result. These conflicts were usually trivial, but they could hurt the team's synergy and collaboration. One of the favorable conditions we try to capitalize on is that our projects are all for IT majors, so, by providing a clear list of self-asked questions for each position of the PM, system designer, coder and tester, our mentors help remind the students of their roles and processes every time they get confused. Our self-asked questions are mostly focused on the followings:

- How a certain role in the team initiate his or her processes.

- How a certain role in the team seek for the right information.
- How a certain role in the team shares the right information to another role.
- Certain kinds of questions that a certain role in the team should make to other roles on a weekly basis.
- Certain questions about the appreciation that one role should have for another in order to get his or her job done.
- How a certain role summarizes and integrates his or her collected information for presentation to the whole team in team meetings.

Building better oral and verbal skills: Since our students in the CDIO projects are all technical ones, most of them do not possess strong skills in both public speaking and report writing. For every team, most of the time, the PM is very outspoken, handling almost all of the team communication while other members usually only focus on their technical work. As a result, our mentors have to create a compulsory Q&A session at the beginning of each team meeting. Before that session, team members of different roles were already asked to prepare a half- to one-page report about their current work and problems. The mentor(s) will randomly pick some team members and ask them to present their reports in front of the whole team. In the process, the mentor(s) will question other members about how they might have connected up to help their partners. While various excuses for not providing enough team support are a usual theme under such Q&A sessions, over time, we found that team members become more comfortable when they present their reports. They also commented that by writing down small reports, they have become more organized in their writing skills and argumentation approach. Some of our mentors have gone further and required more detailed reports over the course of every two to three weeks; however, lengthy reports did not appear to serve our original purpose that well.

Developing outcome-oriented approach in students' solution development: Interestingly enough, as our students go down the Design and Implement phases, they found themselves making the most out of the Conceive component in their search for optimal software development solutions. Contrary to our beliefs, ordinary methods of brainstorming or problem solving, most of the time, only led to radical opinions by the team members, in which every one team member held his or her ideas as more superior and relevant than those of others. As a result, our mentors had to encourage every team member to "conceive" about what they in their mind for the end results of a product or process. By describing their conceived ideas to other members in details, discussions amongst the members become easier, and even, in many cases, other team members can help contribute more to those already conceived ideas. While this outcome-oriented approach through idea-conceiving is very much helpful in handling solution development for small software modules, we have to admit that for big and comprehensive design problems, it is not as effective as expected. For comprehensive design problems, the outcome, in many ways, relies on the brilliance of the system designer(s), and our mentors' role is usually to foster the demonstration of such brilliant ideas by the right team member(s).

Resolve and prevent ongoing or potential conflicts in teams: Handling team's conflicts is probably the most difficult task to our mentors. Most of the time, the conflicts are very subtle, and would not show themselves until it is already too late: For example, for months before the due dates, our mentors usually received very positive reports about the team's progress. However, when the due dates came, our mentors would receive all kinds of emails from different team members complaining about the work of other members in team, and why the poor performance of other members hindered them from delivering the best results. At this

point, it would be very difficult for our mentors as to whether to save the project or to save the working orders of the team formation. By saving the outcome of the project, usually, the mentors will have to give more power to strong team members to take control of the team's work. This, in turn, may undermine the learning experience of other members. On the other hand, by restoring the team's working orders, it usually takes certain amount of time, and many times, the team will fall behind its schedule. More than often, it is not easy to find a balanced solution here. What our mentors have learned was that we always need to have something due at least every one to two weeks into the project to keep the team cohesive and on high alert most of the time. The work to be due also should not be modular to each role (as usual), but rather should be the integrated work of different roles. So, if any one team member fails to carry out his or her role, our mentors can quickly identify and help the team tackle such issue right in time. Another tactic that our mentors use is to pay extra attention to team members who are reserved or silent. Usually, these members tried to avoid conflicts with other members, but in the end, they usually became the biggest source of conflicts because many tasks had gone unchecked or uncompleted. By scheduling private appointments with these members, most of the time, our mentors got very good feedback about what was actually going on with the team. And of course, when conflicts all broke out, transparency in handling different issues is always the key for our mentors to settle things.

ASSESSING THE MATURITY LEVELS OF TEAMWORK CAPABILITIES

Even though the plan-driven methodology and other highly-structured or big-group development processes are also adopted by our students in their CDIO/Capstone projects, given the fact that all of our Capstone teams consist of only 3 to 5 students, the communication and work relationships between the members should be those of an Agile development process. An Agile software development process is for small teams of 3 to 8 members, in which every member can connect with the rest, and the team adopts more of a collaborative style of management and operation (Rico & Sayani, 2009). As a result, the five dimensions proposed by Moe et al for assessing the status of Agile teamwork should be the best instrument for us to measure the maturity level of our students' teamwork capabilities through their CDIO/Capstone projects:

Shared Leadership: At the two extremes of management style are (1) centralization, in which all the decisions of the team are made by an individual or a small group of individuals, and (2) decentralization, in which each member independently make his or her own individual decisions for the team. While Shared Leadership is not a centralized style of management, neither is it a decentralized one. Shared Leadership fosters collective and mutual decision-making on the basis of key knowledge, skills and abilities possessed by different members in the teams (Moe et al, 2009). As a result, the work of the PM under such settings should be focused on maintaining the creating and team's vision and workflow while delegating project tasks to members of the right skills and knowledge.

Team orientation: This is reflected through how much team members are willing to give priority to team goals over their individual goals (Moe et al, 2009). The more priority given by the members, the more effective and efficient the team becomes. A high level of teamwork commitment, however, is usually resulted from the fact that the whole team constantly tries to promote its cohesion and that each team member recognizes his or her responsibilities and is committed to undertaking those responsibilities (Stettina & Heijstek, 2011).

Redundancy: Redundancy will be in place if team members possess a wide array of abilities, skills and knowledge so that they may help with the functions of other team members should there be a need (Moe et al, 2009). Cross-functional team formation is usually very efficient, and is even more important to the success of the project, when the team is small. As a result, students in CDIO/Capstone projects are always encouraged to get familiar with the tasks of other team members.

Learning: Learning is reflected by the fact that the development team is able to develop a common picture of the problems faced, and thus, is able to find solutions that improve team performance (Moe et al, 2009). Moreover, they have to continually adapt to transformations arising from the circumstances. Successful teams tend to constantly improve their ways of doing things.

Autonomy: This dimension describes the extent to which the team is autonomous in its work and decision-making (Moe et al, 2009). Team's autonomy is defined as the ability to avoid or avert the influence of outsiders such as mentors, business clients, other teams, etc. in the case of our CDIO/Capstone projects. It, however, does not mean that the team will not take in feedbacks and advices from outsiders to avoid dilemma like group-thinking (Moe & Dingsøy, 2008).

Based on the five dimensions identified, Moe et al has developed a set of open-ended questions for a structured interview of 20 minutes with the development team. With answers from different members, the team will be rated on a scale of 0 to 10 for each dimension. The ratings then will be presented in the form of a radar plot for the analysis and identification of what improvements the team should make (Moe et al, 2009). As much as the idea of a radar plot is intriguing to most people, one cannot refrain from asking the question of whether the ratings for such radar plot is relevant or reliable enough, especially when the ratings are made out of qualitative answers in the interviews. As a result of this, Stettina and Heijstek has transformed the open-ended questions of Moe et al into a Likert-scale questionnaire in their article of "Five Agile Factors: Helping Self-Management to Self-Reflect" (Stettina & Heijstek, 2011).

Recognizing the quantification benefit of the approach of Stettina & Heijstek, we adopted their questionnaire for our CDIO/Capstone project a year ago at DTU. However, we found two shortcomings of this approach:

- (1) The questions made by Stettina & Heijstek are general for most team formation rather than for IT or software development teams.
- (2) Students tend to give high scores to the questions of Stettina & Heijstek to save face for themselves and their team even though their team interaction was not good at all.

Table 1. Questionnaire Rubric for Maturity Assessment of Teamwork Capabilities in IT Projects

Duy Tan University CDIO/Capstone Project		Academic Year and Semester:				
Team's Name:		Student's Name:			Student's Role (PM/Designer /Coder/Tester):	
Project Name:						
Criteria				Tick One		
<i>5: Strongly Agree - 4: Agree - 3: Neutral - 2: Disagree - 1: Strongly Disagree</i>						
1. Shared Leadership		5	4	3	2	1
Team members are involved in making decisions about every detail of the project, from planning to designing to coding to testing.						
When it comes to important decisions about work scheduling, system design, user's tests, etc., the Project Manager (PM) did consult with other team members.						
The project plan, up until now, is still clear and well-organized without the need for any revision.						
The original role assignment and later role switching (if any) fit the requirements of the project plan and the development of the end product/prototype.						
2. Team Orientation		5	4	3	2	1
Technical and non-technical alternative suggestions of every team member are always welcome by the whole team.						
The PM, system designer(s), coder(s) and tester(s) in my team never appeared to be annoyed if another member of a different role proposed some alternative ways of completing his or her task(s).						
As a PM (or system designer or coder or tester), I regularly get both technical and non-technical inquiries from other team members about my tasks.						
As a PM (or system designer or coder or tester), I usually provide comments and feedbacks to team members of other roles like system designer (or coder or tester or PM).						
3. Effort Redundancy		5	4	3	2	1
As a PM (or system designer or coder or tester), I find it is also easy to complete the task of the system designer or coder or tester (or PM) in my Capstone project.						
I am comfortable as a PM (or system designer or coder or tester) because whenever I have problem, technical or non-technical, my team can always help.						
Even though my role is that of a PM (or system designer or coder or tester), I am always willing to help other team members get their tasks done if they fail.						
If the PM or system designer(s) or coder(s) or tester(s) gets sick for two weeks or more, my team can still function well.						
4. Learning Results		5	4	3	2	1
The team always keep records of good practices in designing, coding, testing, and team management for later reuse.						
The team make necessary changes in its ways of doing things whenever some problem(s) in designing, coding, testing, and team management are identified.						
My team members always provide comments whenever they feel something is not going in the right direction, even if that may annoy some other members.						
5. Team's Autonomy		5	4	3	2	1
Even though all the team members may have other projects and schoolwork, they give the most time and effort for this CDIO/Capstone project.						
The mentors, business clients, and other teams did not cause much change to the original project plan and other team operations during the course of the project.						
Decisions mutually made by the team members have never been turned down or overturned by the mentors, business clients, or any others.						

In order to overcome the first shortcoming, based on the questionnaire of Stettina & Heijstek, we have revised, integrated and developed our own questionnaire rubric, specifically for IT CDIO/Capstone projects only. Table 1 demonstrates the questionnaire that we currently use to measure the level of maturity of our students' teamwork capabilities. The questionnaire adopts a five-point Likert scale, and the scale is put in a reversed order. Based on responses from our students, the mean value of each dimension will be calculated. Besides the mean values, the consistency of the responses to questions within each dimension is very much essential to answering our question about students' teamwork capability maturity. The consistency of students' responses will be estimated using the variance (σ^2), presented through Equation 1:

Eqn (1)

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$$

Whereas, N is the number of responses to an individual question; X is the response score (1 to 5), and μ is the mean value of all the responses to the question. A low variance of the responses would indicate a high level of agreement within a team with respect to any one question and vice versa.

As for the second shortcoming, we realized that rather than leaving the questionnaire to the students all alone, if the students complete the survey under some controlled settings, usually their responses are more objective. As a result, we would hold focus group discussions, in which for each dimension, we first ask students related questions about the actual work in the team, and then, we give them some time to discuss before allowing them to complete the questions for that dimension. Using this approach, our students have responded to the questions in the questionnaire rubric based on the facts retrieved from the focus group discussions rather than from their sheer subjective assessment or estimates. Table 2 below demonstrates the set of questions that we have used in our focus group discussions in order to encourage our students to think about what their team had gone through in terms of communication, collaboration, effort distribution, problem-solving situations, and decision-making process.

Table 2. Kick-Start Questions for Focus Group Discussions about Teamwork Capability Maturity

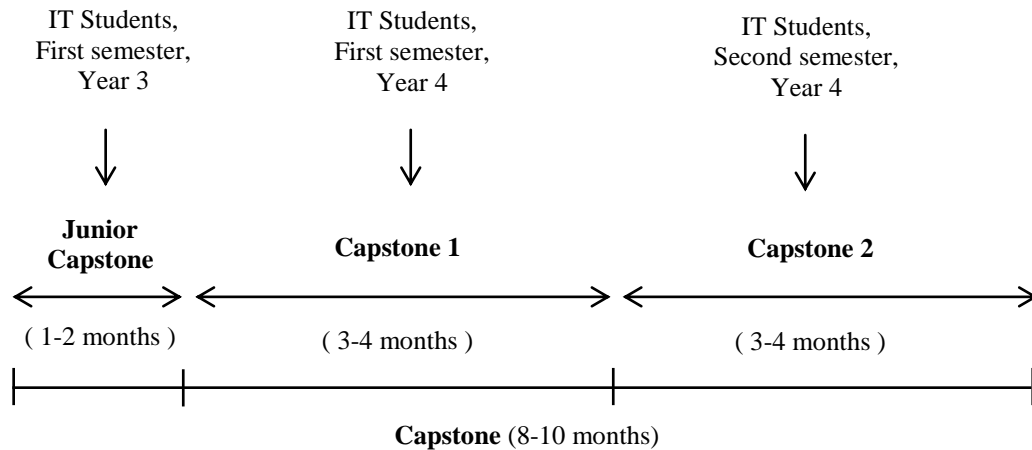
Duy Tan University CDIO/Capstone Project		Academic Year and Semester:	
Team's Name:	Team Member' Name:	Team Member's Role (PM/Designer/Coder/Tester):	
Focus Group Coordinator's Name:			
Project Name:			

Kick-Start Questions for Focus Group Discussion	Note
1. Shared Leadership	
Discuss a time when you were left out of the decision-making process of the team. (if any). Explain the reasons why.	
Is there anyone in your team who may not be willing to consult with other members when it comes to important decisions?	
How many times did your team have to go through the process of revising and/or correcting the vision and scope of the project plan?	
Given the chance, what would you have done with the structures of your team to make it a better one?	
2. Team Orientation	
Did your team get into many arguments because of the different alternatives available for any one task? Did your team choose the best alternative?	
Is it right or wrong spending time arguing about different alternatives for a simple task?	
Did you give comments to your team members about their tasks? How did they react to that?	
Did you get any comments from your team members about your tasks? Elaborate on a time when such comments were useful to your work.	
3. Effort Redundancy	
Do you think you could carry out many tasks, specifically for your project, at the same time like designing, coding, testing? Explain the reasons why.	
Was there a time when you could not move forward with your effort? If yes, did you get any help from your team members at that time?	
Was there a time when you tried to help another team member even though you had not completed your tasks yet?	
Can you identify the team member that without whom, your team's project would have been a crash? Explain the reasons why.	
4. Learning Results	
Did your team keep a journal of what worked and what did not work during the course of the project?	
Was there a time when you or your team ran into the same problem that it had encountered before in the project? If yes, elaborate on the details of such a time.	
Was there a time when your team members ran into the same problem that you had encountered before? If yes, what did you do under such an occasion?	
5. Team's Autonomy	
Was there any of your team members who was too busy with other projects and did not spend enough time with the team?	
Did the mentor(s), business client(s) or any others interfere in the decision-making process of your team? Elaborate on how your team dealt with it (if any).	
Was there a time when the mentor(s) or business client(s) find your team's decisions more creative or original than what they had proposed?	

SURVEY RESULTS

Using the questionnaire rubric in Table 1, we carried out a survey (in the framework of focus group discussions) with our junior and senior IT students of the last and current academic year. We selected three groups of IT students in different stages of their CDIO/Capstone projects, i.e., Junior Capstone, Capstone 1 (first semester of the final year) and Capstone 2 (second semester of the final year).

Figure 1. Students at Different Stages of IT CDIO/Capstone projects



Out of 146 students who were surveyed, 143 students attended the sessions and answered the questionnaire, making up a response rate of 98%. The high response rate was due to the fact that we held mandatory focus group discussions, and except for the two students who were sick, the rest of the students had to attend. The reason we made the sessions mandatory was because we would like our measurement to be complete and objective, at the same time.

Table 3. Results of the Survey

Team ID	N° Students	N° Responses	Shared Leadership	Team Orientation	Effort Redundancy	Learning Results	Team's Autonomy	Average Variance
			X	X	X	X	X	
			σ^2	σ^2	σ^2	σ^2	σ^2	
K17_1	5	5	2.9	3.2	3.3	2.6	2.8	0.9
			0.6	3.0	0.1	0.8	0.2	
K17_2	5	5	2.8	2.8	2.3	2.2	2.4	1.2
			0.5	0.8	0.4	1.0	3.2	
K17_3	5	5	2.9	2.6	3.0	3.1	2.3	2.0
			0.5	1.7	0.9	2.8	3.9	
K17_4	5	5	3.3	2.7	2.7	3.1	2.6	1.4
			1.3	1.5	1.7	2.1	0.5	
K17_5	5	5	3.7	3.8	3.0	3.0	2.3	1.8
			0.9	2.8	0.1	2.0	3.4	
K17_6	4	4	3.5	3.6	2.9	3.3	2.7	1.8
			2.7	1.2	3.2	0.8	0.9	
K17_7	5	5	3.4	3.3	3.1	3.2	2.7	0.9
			0.2	0.5	1.2	0.3	2.4	
K17_8	5	4	3.5	3.5	3.2	2.9	2.6	1.9
			3.1	2.2	1.7	2.3	0.2	
K17_9	4	4	3.1	2.6	3.7	3.5	2.8	

			1.2	2.6	3.0	1.9	1.0	1.9
K17_10	5	5	3.0	2.9	2.6	3.2	<u>2.5</u>	
			0.5	1.2	0.4	0.3	1.0	<u>0.7</u>
Capstone Orientation Average			3.2	3.1	3.0	3.0	2.6	
			1.2	1.8	1.3	1.4	1.7	1.5
K16_1	5	5	4.1	3.9	3.1	3.9	<u>2.9</u>	
			0.1	0.1	0.8	6.0	0.1	1.4
K16_2	5	5	4.0	4.0	3.7	3.9	<u>3.5</u>	
			0.4	0.4	0.1	0.3	2.8	0.8
K16_3	5	5	3.8	4.0	3.7	3.9	<u>3.5</u>	
			0.2	0.4	0.5	9.0	0.8	2.2
K16_4	5	5	4.1	3.9	3.2	3.9	<u>3.0</u>	
			0.1	0.1	0.6	0.7	1.2	0.5
K16_5	5	5	4.4	3.7	4.2	4.2	<u>3.1</u>	
			0.5	0.1	0.6	0.1	0.4	<u>0.3</u>
K16_6	5	5	4.1	4.1	<u>3.7</u>	3.8	<u>3.7</u>	
			0.6	0.8	0.4	0.4	0.2	0.5
K16_7	5	5	4.1	4.0	<u>3.5</u>	3.8	<u>3.5</u>	
			0.2	0.2	0.1	0.1	1.9	0.5
K16_8	5	4	3.9	3.9	3.5	3.8	<u>3.1</u>	
			1.3	0.9	1.1	0.3	0.3	0.8
K16_9	4	4	4.4	4.3	3.8	3.7	<u>3.1</u>	
			0.3	0.5	0.3	0.3	0.4	0.4
K16_10	5	5	3.7	4.1	3.4	3.7	<u>2.9</u>	
			0.4	0.3	0.6	0.8	1.2	0.7
Capstone 1 Average			4.1	4.0	3.6	3.9	3.2	
			0.4	0.4	0.5	1.8	0.9	0.8
K15_1	5	5	4.1	4.1	<u>3.3</u>	3.8	<u>3.3</u>	
			0.7	0.1	0.4	0.1	0.1	<u>0.3</u>
K15_2	5	5	4.6	4.0	3.8	4.1	<u>3.0</u>	
			0.3	0.4	0.4	1.0	2.7	1.0
K15_3	5	5	4.3	4.7	3.8	4.5	<u>3.0</u>	
			0.4	0.1	0.4	0.6	0.5	0.4
K15_4	5	4	4.3	4.3	4.1	4.2	<u>3.5</u>	
			0.5	0.1	0.1	0.1	0.5	<u>0.3</u>
K15_5	5	5	4.2	4.3	3.8	4.0	<u>3.5</u>	
			0.2	0.6	1.6	1.1	0.6	0.8
K15_6	5	5	3.6	4.3	<u>3.1</u>	4.3	3.5	
			0.2	0.1	1.9	1.2	1.4	1.0
K15_7	5	5	4.6	4.2	3.6	4.1	<u>3.5</u>	
			0.7	0.8	0.4	1.0	0.1	0.6

K15_8	4	4	4.7	4.1	4.3	4.6	3.4	
			0.3	0.2	0.8	0.6	0.3	0.4
K15_9	5	5	4.3	4.3	4.0	4.3	3.2	
			0.2	0.6	1.1	0.2	0.6	0.5
K15_10	5	5	4.5	4.3	4.0	4.3	3.1	
			0.1	0.2	0.5	0.4	0.3	0.3
Capstone 2 Average			4.3	4.3	3.8	4.2	3.3	
			0.4	0.3	0.8	0.6	0.7	0.6

In Table 3, the teams who took the Junior Capstone are marked with initials of “K17”, the teams who took Capstone 1 are with “K16” and the teams who took Capstone 2 are with “K15”. For each team, the survey data presented include the team ID, the number of team members, the number of responses, the mean value (\bar{X}) and the variance (σ^2) for all five dimensions (min, max). In the last column, the average variance of all responses is shown. Figure 2 and 3 below depict the overall mean values (\bar{X}) and variances (σ^2) across three different CDIO/Capstone projects.

Figure 2. The final mean values (\bar{X}) for all 3 CDIO/Capstone projects of 143 Respondents

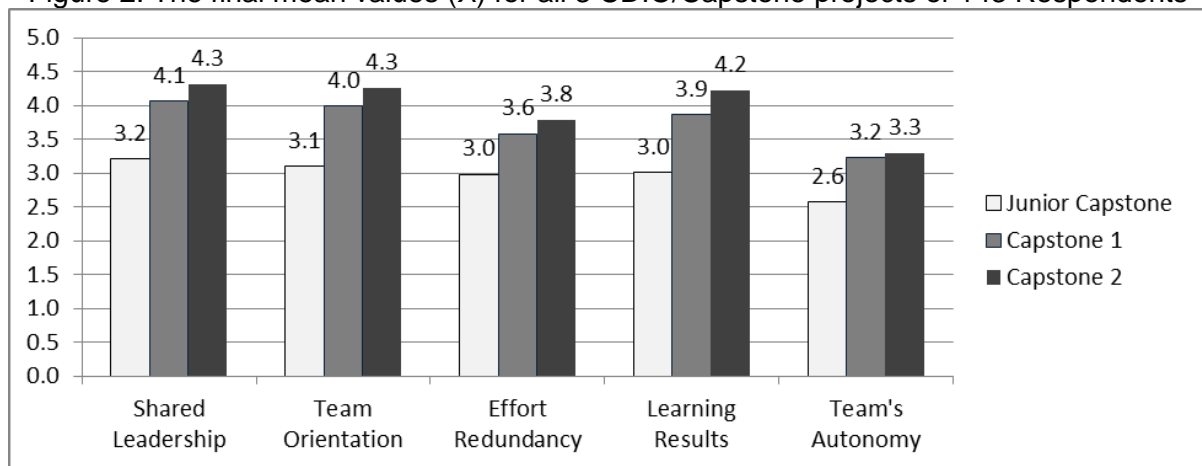
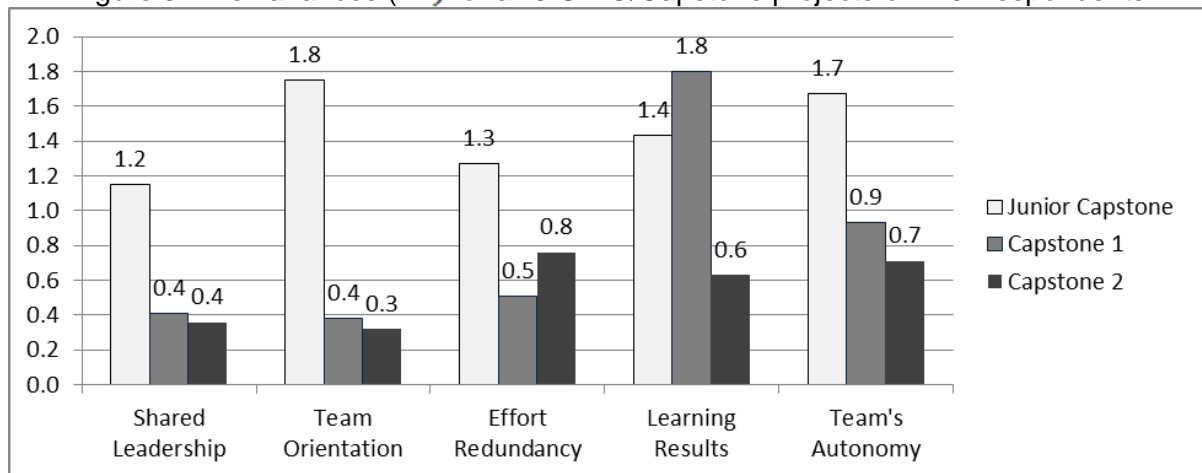


Figure 3. The variances (σ^2) for all 3 CDIO/Capstone projects of 143 Respondents



DISCUSSIONS

Based on the survey results collected, students in all three Capstone projects, i.e., Junior Capstone, Capstone 1 and Capstone 2, mostly agreed that their dimensions of Shared Leadership and Team Orientation were improved the most by the CDIO/Capstone projects. For the students taking the Junior Capstone (K17), four out of 10 teams assigned the highest scores to Shared Leadership, and another four to the Team Orientation dimension. Eight out of 10 teams undertaking Capstone 1 (K16) assigned the highest scores to Shared Leadership (80%) while five teams to Team Orientation. As for the teams taking Capstone 2 (K15), seven teams (70%) assigned the highest scores to Shared Leadership and six to Team Orientation. The high scores for Shared Leadership and Team Orientation indicate the success of the CDIO/Capstone projects in helping our students build a more collective and democratic style of leadership and management. It is also important to point out that there was a significant improvement in the scores for these two dimensions when moving up from the Junior Capstone to Capstone 1 and 2. It appears that the students have built up their teamwork skill-set over time with the CDIO/Capstone projects, and that working with real-world business clients in Capstone 1 and 2 did, in fact, make a difference.

As for the two dimensions of Effort Redundancy and Learning Results, even though the scores were not as high as those for Shared Leadership and Team Orientation, they shared the same patterns as the latter two, in which most students agreed that Capstone 1 and 2 helped leap their interdisciplinary skills and learning capabilities to a new level. One interesting thing, not from the survey, but from the focus group discussions concerning these two dimensions was that most PMs did engage in many cross-functional tasks to get their project done, and many of them were confident that most of the team members will do the same thing for the sake of the project. Also, from the feedback of our focus group discussions, members who took unimportant roles, e.g., that of a tester, tend to complain that they did not learn much teamwork or cross-functional skills because of their dependencies on the work of others. This led us to rethink about how our role assignment should have been done, and perhaps, role-switching should have been mandated in some teams.

The lowest scores, of course, went to the last dimension of Team's Autonomy. The actual scores, however, were significantly lower than those of the other four dimensions, with seven teams out 10 for the Junior Capstone assigning the lowest scores for Team's Autonomy (70%) while 10 teams (100%) and 9 teams (90%) of the two remaining groups for Capstone 1 and 2, respectively, did the same thing. The major justification that our students made was the heavy workload they had to take at DTU, with at least two or three projects at any given time. Another reason indicated by students taking Capstone 1 and 2 was the lack of guidance from business clients, who mostly asked for deadlines and did not help much, especially when it came to product integration. Students taking Capstone 1 and 2 also seemed to have fewer problems with their mentors from the Faculty of Information Technology and International School.

In terms of the variance for the scores on the five dimensions, the highest overall variances were amongst the scores given by students taking the Junior Capstone, ranging from 1.2 to 1.8 (Figure 3). This is understandable given that the Junior Capstone in the junior year is actually the first full-scale CDIO/Capstone project that they took while the first CDIO/Capstone course in the sophomore year mostly consists of interactive group games to give students a taste of CDIO and engineering. The overall variances in the scores given by students taking Capstone 1 and 2 were generally low, from around 0.3 to 0.9. This shows

that the survey results of our industry-involved CDIO/Capstone project 1 and 2 are reliable. Interestingly, the highest overall variance (1.8) was with the scores on Learning Results of students taking Capstone 1. Feedbacks from the focus group discussions indicate that most of these students were shocked when they first learned about the various skills that they need in order to work well with real-world business clients. This, again, signifies the importance of teamwork and interpersonal skills besides technical ones for today's engineering students.

CONCLUSIONS

In conclusion, while it has been recognized that the maturity level of teamwork capabilities of an engineering student is essential to his or her future career success, finding ways to improve and assess that maturity level is a rather daunting task. Fortunately, the CDIO Framework has come along and its CDIO project settings have provided a relatively good answer. The remaining task of assessing that maturity level was a difficult one to our staff at Duy Tan University for quite some time until we learned about the five dimensions of Moe et al: (1) Shared Leadership, (2) Team Orientation, (3) Effort Redundancy, (4) Learning Results, and (5) Team's Autonomy. Based on the research of Moe et al and Stettina & Heijstek, we have managed to develop our own questionnaire rubric to assess the teamwork capability maturity of IT students through CDIO/Capstone projects. The remaining issue has to do with the fact that our tool is for the assessment of small teams with Agile development process only. At times, some of our Capstone teams do work on some portion of a much bigger project of some big business client, who adopts the Plan-driven development process. Such circumstances always make us question about the validity of our rubric, and there is, indeed, a need to search for another tool for the assessment of the teamwork capability maturity in bigger teams or organizational structures.

REFERENCES

Accreditation Board for Engineering and Technology, Inc. (2009). *Criteria for Accrediting Engineering Programs*. ABET: <http://www.abet.org>.

ACM - Association for Computing Machinery (2001). *Computing Curricula 2001 - Computer Science*. National Science Foundation under Grant No. 0003263: December 2001.

CDIO™ Initiative (2010). CDIO Standards v. 2.0. <http://www.cdio.org>.

Crawley, E.F., Malmqvist, J., Ostlund, S., and Brodeur, D. (2007). *Rethinking Engineering Education: The CDIO approach* (1st Ed.). Springer.

Denning, P. J. (1992). *Educating the New Engineer*. Communications of the ACM, 35 (12).

Holdsworth, A., Watty, K. and Davies, M. (2009). *Developing Capstone Experiences*. Melbourne: Centre for the Study of Higher Education, University of Melbourne.

IEEE Computing Society and Association for Computing Machinery (2004). *Software Engineering 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. National Science Foundation under Grant No. 0003263: August 2004.

Lingard, R. W. (2010). *Teaching and Assessing Teamwork Skills in Engineering & Computer Science*. U.S.: Journal of Systemics, Cybernetics and Informatics.

Proceedings of the 10th Annual International CDIO Conference, Universitat Politècnica de Catalunya, Barcelona, Spain; June 16 - 19, 2014.

Moe, N. B. and Dingsøy, T. (2008). *Scrum and Team Effectiveness: Theory and Practice*. Limerick, Ireland: Proc. 9th Inter. Conf. on Agile Processes in Software Engineering and Extreme Programming, pp. 11-20.

Moe, N. B., Dingsøy, T. and Røyrvik, E.A. (2009). *Putting Agile Teamwork to the Test: A Preliminary Instrument for Empirically Assessing and Improving Agile Software Development*. Business Information Processing, 31, pp. 114-123.

Rico, D. F. and Sayani, H. H. (2009). *Use of Agile Methods in Software Engineering Education*. Chicago, U.S.: Proc. Agile 2009 Conf., pp. 174-179.

Stettina, C.J. and Heijstek, W. (2011). *Five Agile Factors: Helping Self-Management to Self-Reflect*. Proc. 18th European System & Software Process Improvement and Innovation Conf. Roskilde, Denmark: Communications in Computer and Information Science, 172, Springer, pp. 84-96.

Todd, R. H. and Magleby, S. P. (2005). *Elements of a Successful Capstone Course considering the Needs of Stakeholders*. Europe: European Journal of Engineering Education 30 (2), pp. 203-214.

Welch, R.L. (2000). Training a new generation of leader. U.S.: Journal of Leadership Studies.

BIOGRAPHICAL INFORMATION

Vu T Truong (in Vietnamese name order as Truong Tien Vu) is the Dean of the Faculty of Information Technology at Duy Tan University, Vietnam. His research interests include Information Technology education, cooperative communications, and physical layer security.

Nguyen-Bao Le is the Vice Provost of Duy Tan University. He is in charge of the Technology & Science division as well as the R&D Center of DTU. His interests are in data warehousing, OLTP, graphics and animation design.

Man D Nguyen is the Acting Dean of the International School of Duy Tan University. His majors are in Software Engineering and Information Systems Management. He has more than 10 years of experience in software development. His interests are in software testing, mobile application development, and large-scale data processing.

Thang M Nguyen is an instructor in Information Systems of the International School, Duy Tan University. His interests are in software project management, software process, business decision-making, and IT in business.

Corresponding Author

Mr. Vu T Truong
Dean, Faculty of Information Technology,
Duy Tan University
K7/27 Quang Trung, Da Nang
Vietnam 59000
+84 914083188
truongtienvu@dtu.edu.vn



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/).