A NEW CDIO-BASED CROSS-CULTURE TRAINING PROGRAM FOR INTERNATIONAL SOFTWARE ENGINEERS

Yingqiu Li, Tao Wen, Lu Yang

Dalian Neusoft University of Information

ABSTRACT

One of main tasks of modern software engineering education is to enable students to appreciate the Fundamentals of Software Engineering: the methodologies, techniques and tools for software development. Besides, the soft skills like management, communication and teamwork are very important for a cross-culture development team. This can be achieved by introducing a new CDIO-based software project within the framework of practice-oriented cross-culture program. The program is carried out by students, consisting of the Chinese students in the fourth year and IT engineers from Japanese companies, to work out a software project based on teamwork within two months.

The project is done in a full lifecycle development of software engineering. The students are supposed to finish the whole lifecycle software development with the methodologies, techniques and tools learned in class. In order to build a self-directive development team, some team facilitation skills are introduced. At the beginning of each Plan-Do-Check-Act (PDCA) cycle, development teams make the plan, while at the end of each PDCA cycle, teams summarize their experience, find out some questions and try to propose solutions, which aim to improve the efficiency and effect of the next PDCA cycle. This kind of Project-based Education and Learning (PBEL) is helpful to deepen students' understanding of theoretical knowledge, cultivate students' ability of autonomic learning, self-management, critical thinking and problem-solving.

Results show that both Japanese engineers and Chinese students gain rapid progress in the training. They have positive influence to each other due to the different cultural background and active cooperation, which guarantee the project to be finished punctually.

KEYWORDS

CDIO-Based Cross-culture Training, team facilitation, PBEL

CDIO is an acronym for Conceive, Design, Implementation and Operation. It is a synoptic and abstract expression of “Learning through Practice” and “Project-based Education and Learning, PBL”. Different from the traditional teaching methods based on the discipline, PBL stresses on active learning and merges the learning into the complex and real-world scenario where the students can learn the knowledge hidden in the problems by autonomic exploration and cooperation. As a result, the problem solving skills and active learning ability can be developed. Such pragmatic method provides systems approach to strengthen ability and creates 12 standards for implementation process and outcome assessment (1).

So far, more project-based engineering education has been carried out in universities (2, 3, 4). Responding to the demands from students, instructors, industry, society and other stakeholders and combining the status quo of Chinese tertiary education and practice experience on IT talent nurturing, Dalian Neusoft University of Information inherited and developed the achievement of international engineering education-CDIO INITIATIVE to create TOPCARES-CDIO nurturing model for application-oriented talents from 2009 (5). The model implemented throughout campus is an all-round and systematic reform.

Under such background, the Cross-culture Training Program for International Software Engineers is developed since 2010. It is a joint program cooperated between software engineering major and Japanese educational training organization. Participants comprised of IT engineers and the fourth-year students of software engineering in an international classroom accomplish the software development project following software development process in group work instructed by professional teachers. The Japanese IT engineers involved in the program are from famous IT enterprises and engage in system development, maintenance and consulting for two or three years. But they are not qualified with development experience in an entire program, working experience in a cross-culture environment and management experience. Chinese students are capable of software development, but less real experience of project development.

The following abilities are required to be enhanced by the training program.

1) Master the popular objective-oriented system analysis and design methods.
2) Understand how to conceive-design-implement-operate a complex software system in the context of group work.
3) Master management method of software project and use it in the real-world project.
4) Communicate effectively in the cross-culture working environment.
5) Write and present technology report in English.

Based on the CDIO Initiative, we design the training content and implementation process and integrate the above five abilities in the training program. The training program is optimized in the past four years. The program source, functions, structure, process management and assessment methods are introduced in details in this paper.

1 PROGRAM SOURCE AND CONTENT

The training is designed to focus on one software development project which is a real-world project for recruitment required by one of famous Chinese outsourcing company. Since 1995, this company has delivered world-class Business / IT consulting, solutions, and outsourcing services to global clients. With the dramatic increase in business these years, the company has been an international software company with a wide range of business in Japan, Singapore, the United States, Australia and Europe. A huge number of software developers are recruited from campus and human resource market each year for the business needs. 80 recruiters engage in the related work and review great amounts of recruitment information. As a result, a recruitment management system is in demand for controlling recruitment process and classifying resumes. The functions of the recruitment system include:

1) Job Application
   - Online application
   - Resume uploading
2) Resume processing
   - Resume standardization
   - Candidate status labels
3) Recruitment management
   - Custom Recruitment Flow
   - Candidates state share
4) Data Statistics
   - Common statements
   - Custom statements
5) Personal workbench management
   - Recruiting Process
   - Work transfer
6) Authentication
7) Role Management

The development lifecycle in the project is two months. The organizers adjust the development demand, resize the project and reduce the complexity so that the project is suitable for in-class teaching. Students can combine theory and practice by learning how to conceive, design, implement and operate in a real-world project. For example, we design a field trip for students to observe what recruiters do in company at the stage of project demand. Students can know what is real in need by communicating with the recruiters. Additionally, students can experience the real project by contacting directly with clients and learn how to acquire demand.

2 PROJECT STRUCTURE

The project is run in the group comprised of two Japanese participants and two Chinese students. There are six roles in the group, including Team Leader (TL), Chief Architect (CA), Requirement Engineer (RE), Quality Assurance Manager (QAM), Configuration Manager (CM), Test Engineer (TE). Each member has to play one or two roles. In the given stage of the development, a member plays team leader and leads other members. The responsibilities of different roles are listed in the followed.

What TL to do:

1) Be responsible for the overall project planning and progress toward the implementation.
2) Be responsible for monitoring, maintaining, and adjusting the project plan.
3) Review and assist in issue resolution, coordinate task dependencies, and establish milestone project goals that keep the project on time and within budget.
4) Continually monitor scope, time frame, budget and risk.
5) Manage project resources to assure maximum efficiency, effectiveness and resource utilization.
6) Conduct Status meeting and retrospective meeting.

What CA to do:

1) Lead and coordinate technical activities and artifacts throughout the project.
2) Establish the overall structure for each architectural view: the decomposition of the view, the grouping of elements, and the interfaces between these major groupings.

What RE to do:

1) Be responsible for writing the Architecture Drivers Specification, document and requirements Traceability Matrix.
2) Conduct Client Meeting.
3) Maintain SRS (Requirement model, Architecture Drivers Specification, Requirements Traceability Matrix).

What QAM to do:

1) Facilitate / coordinate review activities.
2) Review documentation on outgoing products.
3) Supervises the quality control team at all times. For example, checking team members' reports regarding testing.
4) Ensure that the project processes are being followed
   - Planning / tracking /risk management process.
   - Client communication / requirements elicitation process.
   - Meeting / review process.
   - Configuration management (versioning / change management).

• Design / design evaluation process, implementation / testing process.

What CM to do:

1) Track and control changes in the software.
2) Identify configurations, configuration items and baselines.
3) Implement a controlled change process. This is usually achieved by setting up a change control board whose primary function is to approve or reject all change requests that are sent against any baseline.
4) Implement a controlled change process.
5) Manage the process and tools used for builds.
6) Manage the software and hardware that host the system.

What ET to do:

1) Develop test plans and follow guidelines set by test plan.
2) Build test environments and update bug records.\
3) Review software specification and create documents to identify test scenarios.
4) Conduct system integration test.
5) Track software engineering, test processes and procedures.
6) Deliver project quality with development teams.

3 HOW TO ORGANIZE TO DEVELOP THE PROJECT

The iterative development is adopted in the project. One iterative development is composed of six activities, demand acquisition, system analysis, system design, detailed design, implementation and test. Figure 1 shows these activities and corresponding models.

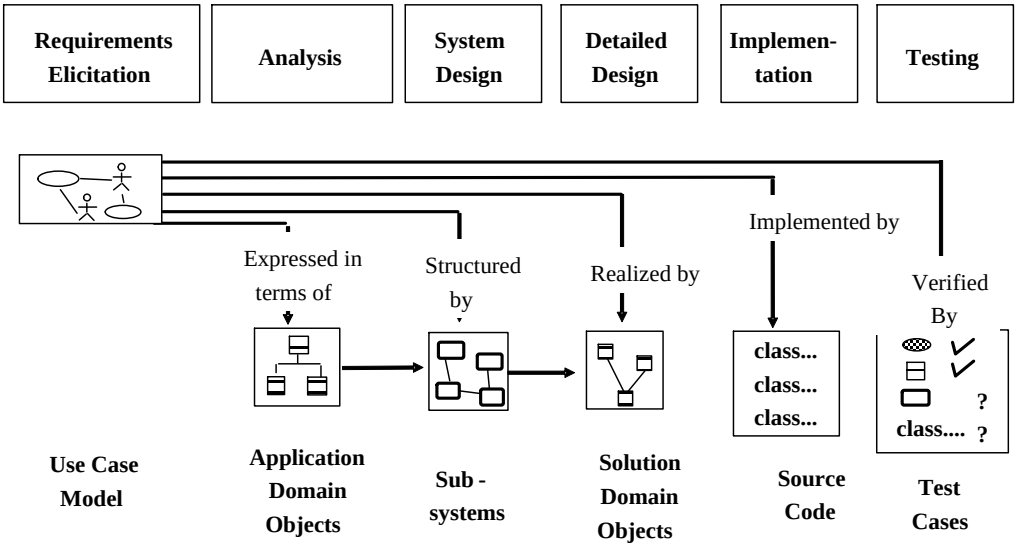

Figure 1.Software Lifecycle Activities and Models

Using the experience of Japanese project management for reference, the Team Facilitation is introduced to the project so as to build a self-directed and self-management development team. The Team Facilitation is composed of Activity, Tools of “MIERUKA” and Rhythm. Shown in Figure 2.

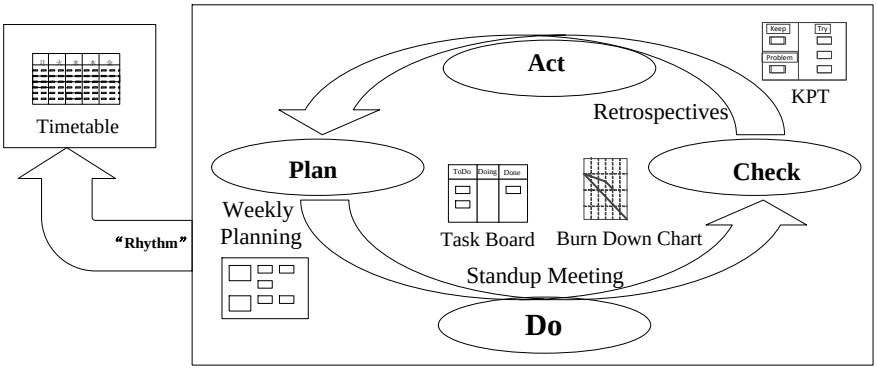

Figure 2. Structure of Team Facilitation

In this project, one week is a PDCA management cycle. The activities in this cycle include Weekly Planning, Standup Meeting, Retrospectives, etc. The Weekly Plan is drawn in the “Plan” stage firstly. This Plan is derived from the upper project plan, which aims at sharing Task Goal in the development team and definite what to do. The team has a “Standup Meeting” in each morning where all members review and discuss the project progress. Each member reports the work status, daily plan and problems. The meeting is short and brief, less than 15 minutes, and all members stand in front of “Task Board” to finish it. The members host the meeting in turn. If a problem has to be discussed deeply, the TL will coordinate relating members to participant after the morning meeting. The “Retrospectives”, carried out at the end of one management cycle, is aimed to facilitate project progress by self-study. Members can review the project and sum up good experience; meanwhile they can take action to solve the problems. The “Retrospective” activity can be undertaken by drawing KPT on the white board. KPT refers to Keep, Problem, and Try. Firstly, the white board is divided into three parts. Then members, by group discussion, sum up “Keep: things to keep doing”, pose “Problem: things which are problem” and find “Try: things to try; do better; and be improved”. Figure 3 show the KPT details.

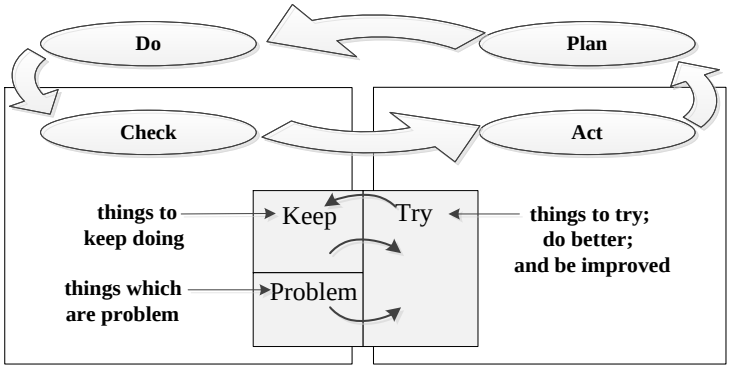

Figure 3. KPT

The word “MIERUKA” in Tools of “MIERUKA” means visual. It refers to manage development process of project by visual tools. Firstly, there is a very useful tool-Task Board, shown in Figure 4. The Task Board is made by the team in “Standup Meeting” each morning. It has three areas, *To Do, Doing, Done*. The daily work, derived from Weekly Plan, is listed in the *To Do*. *Doing* includes that work is being in processed. *Done* means the completed work. Team members can adjust the work status according project progress.

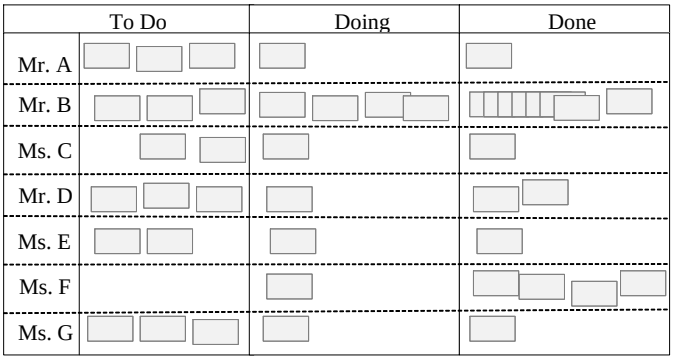

Figure 4. An example of Task Board

Figure 5 shows the specific layout of a task in the Task Board.

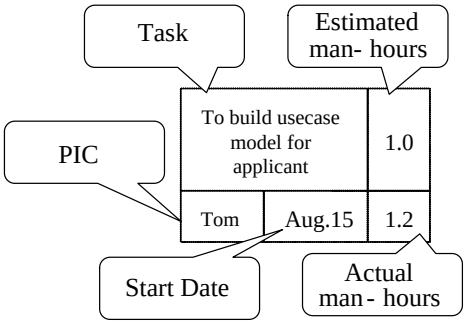

Figure 5. Layout of a Task

Firstly, it shows what task is in briefly. Then it shows who is the responsible person, when to start the project, and estimated the workload. It also lists the real workload when the project is completed. The project progress and what each member to do can be presented by the simple and easy-to-use tool, the Task Board.

Another tool is *Burn Down Chart*. By drawing the rest work of each day, the team can supervise the task progress and estimate when the tasks can be completed.
Rhythm means that the development work which each member involved in can be defined in one “Timetable”, so that members in the team can cooperate effectively.

4 ASSESSMENT AND EVALUATION

Both the assessment and evaluation focus on the project participants. The presentation evaluation and peer evaluation are the judgment methods. In details, the evaluation for each participant is composed of individual evaluation and team evaluation, and the percentage is 50 per cent and 50 per cent.

1) Individual evaluation. Including self-evaluation and TL evaluation. The percentage of two parts is 20 per cent and 80 per cent respectively. The contents as followed:
   - Taking Responsibility: Completes assignments in a thorough, accurate, and timely manner that achieves expected outcomes; exhibits concern for the goals and needs of the department and others that depend on services or work products; handles multiple responsibilities in an effective manner; uses work time productively. This part accounts for 30 per cent.
   - Problem Solving/Creativity: Identifies and analyzes problems; formulates alternative solutions; takes or recommends appropriate actions; follows up to ensure problems are resolved. This part accounts for 30 per cent.
   - Collaboration/Teamwork: Uses diplomacy and tact to maintain harmonious and effective work relationships with co-workers and constituents; adapts to changing priorities and demands; shares information and resources with others to promote positive and collaborative work relationships; supports diversity initiatives by demonstrating respect for all individuals. This part accounts for 20 per cent.
   - Communication/Interpersonal Skills: be able to effectively communicate and to influence others in order to meet organizational goals; share information openly; relate well to all kinds of people; be able to speak well and write effectively. This part accounts for 20 per cent.
2) Team evaluation. The teacher mark the team by observing the participants how to organize the development process, evaluating the project outcome and presentation. The contents include:
   - Quality of the oral presentation (20%): Were the points and issues clearly and concisely covered in a professional manner?
   - Quality of time management (10%): Was the time allocated effectively in presenting the project?
   - Documents (20%): How well did the team finish the documents? Can others get enough information from the documents? Did they follow the writing rules of each document?
   - Quality of phase work (30%): Did the team do a good job in this phase?
   - Management (20%): What is the overall quality of management?

Since 2010, we organized 14 trainings for international software engineers. 176 Chinese and Japanese trainees join in the training accumulatively. Under a cross-culture training environment, members complete a lifecycle of software development project which lasted two months and took English as the working language. It is actually a huge challenge for Chinese students and Japanese engineers, as a result of limited time, difficulties in communication and

different culture background. At the beginning of the project, it is hard for participants to discuss and cooperate. However, they can find the effective ways to understand each other by self study, group discussion, cooperative research, so as to achieve consensus in very short time. During such process, all members enhance their communication skills, interpersonal skills and language skills.

By evaluation and comparison, there is dramatic improvement in system analysis and design ability and project management ability by averaging the values of 14 trainings. The change of 18 abilities of analysis and design are shown in the Figure 6. While Figure 7 presents the differences of project management by comparing the date at the beginning stage and at the end of project. Judging from the result, trainees' abilities are improved to some extent. Therefore, we acquire the expected results of training project.

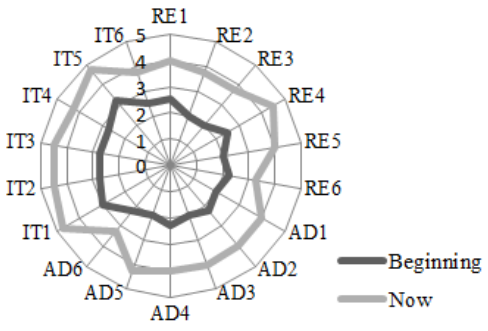

Figure 6.Evaluation of Analysis & Design

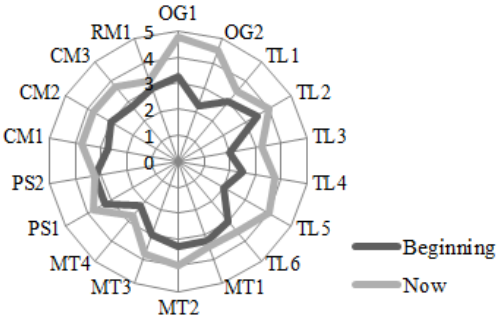

Figure 7. Evaluation of Management

5 CONCLUSIONS

Our training program aims to improve students' ability in software development and the soft skills such as management, communication and team work in a cross-culture environment. To achieve the goals, we design the training content and the implementation process based on the CDIO Initiative. Training Objectives are integrated in a project and students are supposed to finish it in team. In addition, to build a self-directed and self-management team, several PDCA cycles are built in the process and some efficient team facilitation skills, such as activities, tools of "MIERUKA" and rhythms, are introduced into the program. Practices show that students' ability in project management and soft skills have been improved dramatically. As a result, they can achieve the expected results.

REFERENCES

(1) Edward F. Crawley, Johan Malmqvist, Soren Ostlund (2010). *Rethinking Engineering Education: The CDIO Approach* [M].Springer-Verlag New York Inc.

(2) Takemata K., Kanazawa Tech. Coll., Ishikawa, Minamide A., Kodaka A., Nakamura S.. Engineering project-based learning under the CDIO concept [C]. IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), August 2013.

(3) Yadav S.S.,Jianbing Xiahou. Integrated project based learning in software engineering education [C]. International Conference on Educational and Network Technology (ICENT), June 2010 .
(4) S. Anwar, P. Favier, D. Jouglet. A project-based international collaboration in engineering education[C]. ASEE 2008 Annual Conference Proceedings CD ROM publication, session 2160.
(5) Tao Wen. The Exploration of Integrated Talents Nurturing Model Based on TOPCARES-CDIO [J]. *China Higher Education*, July 2011.

BIOGRAPHICAL INFORMATION

Yingqiu Li, Ph.D. is the associate professor and deputy dean in the Department of Computer Science and Technology at Dalian Neusoft University of Information. Her current research focuses on the software engineering, innovation of engineering education and practice.

Tao Wen, Ph.D. is the President of Dalian Neusoft University of Information. As a professor and educator, his current research and scholarly activities focus on education management, network security and start-ups and innovation of engineering education.

Lu Yang, is the dean of School of International Education, Dalian Neusoft University of Information. She is the organizer of international educational cooperation and cross-culture program. She focuses on the project management.

Corresponding author

Dr. Yingqiu Li
Dalian Neusoft University of Information
8 Software Park Road
Dalian 116023
+86 411 8483 2268
liyingqiu@neusoft.edu.cn