International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:11, No:11, 2017

CDIO-Based Teaching Reform for Software Project Management Course

Liping Li, Wenan Tan, Na Wang

Abstract**—**With the rapid development of information technology, project management has gained more and more attention recently. Based on CDIO, this paper proposes some teaching reform ideas for software project management curriculum. We first change from Teacher-centered classroom to Student-centered and adopt project-driven, scenario animation show, teaching rhythms, case study and team work practice to improve students' learning enthusiasm. Results showed these attempts have been well received and very effective; as well, students prefer to learn with this curriculum more than before the reform.

Keywords**—**CDIO, teaching reform, engineering education, project-driven, scenario animation simulation.

I. Introduction

WITH the fast development of information technology, the importance of project management has aroused the world's attention. Software project management aims to make the software project complete successfully according to preplan cost, schedule, scope, quality and so on. This course has a very important significance to cultivate student project management abilities. Students must learn in advance, the principles, methods and some tools of project management, and then they can cooperate with team members better and gradually grow to become an excellent software project manager later on.

Software project management is also a course which has certain depth and breadth, which not only require much theory but also lots of practice. But, for most undergraduates, who without any experiences in actual software development and management, they feel the content which they learn in classroom is boring, useless and difficult to understand. In school, most students do not work hard on this course and just want to pass the exam. This makes the teaching difficult and results in failure to attain the teaching purpose. The survey indicated that the current teaching method used in the software project management curriculum does not meet the requirements of the market and needs reform [1]. Results of the survey also found that more than 98% of respondents thought that teaching should not only be based on the basic theory of teachers' teaching, but pay more attention to the combination of engineering practice, the project cases, as well as frontier development [2]. Especially, the goal is to improve students' software design ability, self innovation ability, teamwork ability and interpersonal communication ability.

Our school is an application-oriented university which pays attention to the training of students' practical ability. Some reforms and measures must be adopted to change the bad situation and to improve the teaching quality of the software project management course. This paper proposes some reform ideas for this curriculum based on CDIO.

According to the CDIO syllabus, we reform the teaching mode, teaching methods, teaching means and teaching resources etc. This change is the shift from the Teacher-centered to the Student-centered classroom. We adopt scenario animation show, practice teaching, case study and team work to enhance students' learning enthusiasm, and change their passive learning into active learning. The purpose is to create a good learning atmosphere, let students better understand and use the learned knowledge to solve actual problems. The outcome of this reform showed good effect, as evidenced by feedback from students and their grades.

The remainder of this paper is organized as follows: Section II introduces the core principle of CDIO; Section III presents some reform ideas of course teaching, like cases study and animation show etc. Section IV proposes the reform ideas on practice teaching and assessment; Section V presents the reform on teachers' ability; and finally, Section VI draws a conclusion for this paper.

II. The Basic Concept of CDIO

The CDIO aims at developing a new model for engineering education, focusing on *Conceive, Design, Implement and Operate* skills. A good software engineer need not only grasp the software theory and basic knowledge, but also must have a deep understanding of the software industry and software project. They also require having a solid innovation design ability, communication skills, teamwork spirit, ability of lifelong learning etc. The CDIO Syllabus is a set of goals for engineering education, including technical knowledge, personal, interpersonal and professional skills etc. Table I is the condensed CDIO syllabus [3]. It is the basis for curricular improvement in constructive alignment. The detailed educational syllabus of CDIO can be referred to [4]. Because software project management is a comprehensive curriculum which has strong engineering background. This paper introduces the teaching reform of how CDIO principles were applied in designing of this course in our school.

Liping Li, Wenan Tan, and Na Wang are with the Computer and Information Engineering Institute, Shanghai Polytechnic University, Shanghai, 201209, China (e-mail: liliping@sspu.edu.cn, watan@sspu.edu.cn, wangna@sspu.edu.cn).

TABLE I
CONDENSED CDIO SYLLABUS

1		**Technical Knowledge and Reasoning**
	1.1	Knowledge Of Underlying Sciences
	1.2	Core Engineering Fundamental Knowledge
	1.3	Advanced Engineering Fundamental Knowledge
2		**Personal and Professional Skills and Attributes**
	2.1	**Engineering reasoning and problem solving**
	2.1.1	Problem Identification and Formulation
	2.1.2	Modeling
	2.1.3	Estimation and Qualitative Analysis
	2.1.4	Analysis With Uncertainty
	2.1.5	Solution and Recommendation
	2.2	**Experimentation and Knowledge Discovery**
	2.2.1	Hypothesis Formulation
	2.2.2	Survey of Print and Electronic Literature
	2.2.3	Experimental Inquiry
	2.2.4	Hypothesis Test and Defense
	2.3	**System Thinking**
	2.3.1	Thinking Holistically
	2.3.2	Emergence and Interactions in Systems
	2.3.3	Prioritization and Focus
	2.3.4	Trade-offs, Judgment and Balance in Resolution
	2.4	**Personal Skills and Attitudes**
	2.4.1	Initiative and Willingness to Take Risks
	2.4.2	Perseverance and Flexibility
	2.4.3	Creative Thinking
	2.4.4	Critical Thinking
	2.4.5	Awareness of One's Personal
	2.4.6	Knowledge, Skills and Attitudes
	2.5	**Professional Skills and Attitudes**
	2.5.1	Professional Ethics, Integrity,
	2.5.2	Responsibility and Accountability
	2.5.3	Professional Behavior
	2.5.4	Proactively Planning for One's Career
3		**Interpersonal Skills: Teamwork and Communication**
	3.1	**Teamwork**
	3.1.1	Forming Effective Teams
	3.1.2	Team Operation
	3.1.3	Team Growth and Evolution
	3.1.4	Leadership
	3.1.5	Technical Teaming
	3.2	**Communications**
	3.2.1	Communications Strategy
	3.2.2	Communications Structure
	3.2.3	Written Communication
	3.2.4	Electronic/Multimedia Communication
	3.2.5	Graphical Communication
	3.2.6	Oral Presentation and Inter-Personal Communication
	3.3	**Communication In Foreign Languages**
	3.3.1	Communication in English
	3.3.2	Communication in Intra-EU Languages
	3.3.3	Communication in Extra-EU Languages
4		**Conceiving, Designing, Implementing And Operating Systems In The Enterprise And Societal Context**
	4.1	**External And Societal Context**
	4.1.1	Roles and Responsibility of Engineers
	4.1.2	The Impact of Engineering on Society
	4.1.3	Society's Regulation of Engineering
	4.1.4	The Historical and Cultural Context
	4.1.5	Contemporary Issues and Values
	4.2	**Enterprise And Business Context**
	4.2.1	Appreciating Different Enterprise Cultures
	4.2.2	Enterprise Strategy, Goals and Planning
	4.2.3	Technical Entrepreneurship
	4.2.4	Working Successfully in Organizations
	4.3	**Conceiving And Engineering Systems**
	4.3.1	Setting System Goals and Requirements
	4.3.2	Defining Function, Concept and Architecture
	4.3.3	Modeling of System and Insuring Goals Can Be Met
	4.3.4	Development Project Management
	4.4	**Designing**
	4.4.1	The Design Process
	4.4.2	The Design Process Phasing and Approaches
	4.4.3	Utilization of Knowledge in Design
	4.4.4	Disciplinary Design
	4.4.5	Multidisciplinary Design
	4.4.6	Multi-Objective Design (DFX)
	4.5	**Implementing**
	4.5.1	Designing the Implementation Process
	4.5.2	Hardware Manufacturing Process
	4.5.3	Software Implementing Process
	4.5.4	Hardware Software Integration
	4.5.5	Test, Verification, Validation and Certification
	4.5.6	Implementation Management
	4.6	**Operating**
	4.6.1	Designing and Optimizing Operations
	4.6.2	Training and Operations
	4.6.3	Supporting the System Lifecycle
	4.6.4	System Improvement and Evolution
	4.6.5	Disposal and Life-End Issues
	4.6.6	Operations Management

III. REFORM ON COURSE TEACHING

Software project management includes nine knowledge areas and five standard processes of PMBOK® Guide (Guide to the Project Management Body of Knowledge) [5]. Because software project management requires much experience and involves many important theories, students find it difficult to learn well. Most universities use only the traditional teaching method of a chalk and blackboard, this traditional teaching method lacks visual effect. It is difficult to fully demonstrate the process of project management, and therefore, students feel abstract when they only learn principles by traditional teaching method. Even with the multimedia teaching, simple slide transitions also cannot make full use of the demonstration effect.

Reference [6] gives the top 10 recommendations according to a survey of learning experiences among the three participating Swedish engineering programs students. We refer to three of them:

1) Application is the road to understanding theory. We should concentrate on connecting knowledge of basic concepts with reality. This will encourage a deep learning and understanding of the learned knowledge.

2) Design and give students explicit tasks and activities to complete by both students' interaction and student-teacher interaction. This will encourage teamwork and deeper understanding.
3) Make explicit criteria for course assessment [6]. We should guide students to understand exactly what they must complete and submit, and what they need to do to achieve high scores.

Some reforms have already been made for the software project management curriculum [7]. We have communicated with some peers in IT companies and teachers in other universities. Based on the investigation, we synchronous updated course content, teaching methods and teaching methods of the software project management course. In the process of teaching, we mainly use project-driven and actual cases study [8]. The fundamental procedure is "basic concepts introduction →cases imported → cases analyzing → question → summary". We first explain the principles using multimedia slides, and choose some actual case studies that allow students to understand the process of software project management. Case study teaching mode is one of the best ways for students to understand a subject theory. The actual methods which could be used are shown as below:

Firstly, we use heuristic teaching methods. The teaching focuses on training students' abilities in finding, analyzing and solving problems. In course teaching, we use a whole case study to help students understand the actual procedure of project management. Like, how to plan, budget, schedule and control the project during the process of software project management, as well as providing additional cases to practice after classes. Now, we have built a case study library which contains some good actual cases, including some success and failure project cases, some object-oriented cases and structured cases. We also collect a lot of templates for management process areas, such as template for project initiation phase, planning phase, execution control phase, project closure phase and other standard templates.

Secondly, play some related scenario animations during the process of basic concepts introduction. These animations simulate the procedures of software project management. They can help students to know what will happen and how to handle them at the various stages of the project management. We have already collected some vivid scenario simulation animations. These animations can help students to understand the concepts intuitively and easily. Teachers will ask some questions like "What do you think about it?" "Do you think it is right or wrong?" after students watching these animations. These animations can vivid classroom atmosphere and enhance the students' interest in learning.

Thirdly, in order to improve student's active participation and interaction with teachers in class, we encourage students ask questions in classroom if they have questions. Teachers should answer the questions and often ask a few more words like "Is there a problem?" after one knowledge point is completed.

Fourthly, assign appropriate homework. In addition to the exercises in textbooks, students are required to reading and analyzing relevant news and papers of project management. They must write a paper review after reading. And they also should do some curriculum design.

Lastly, construct a course website, include most teaching materials, cases study, templates, animations and other course resources for browsing and download. By the website, students can learn after class, do test online, ask questions and find answers online etc.

IV. Reform on Practice Teaching and Assessment

Application is the road to understanding theory. Software project management course stresses on training students' skills of practice and engineering. A good project manager should master not only the theory of project management, but also need a lot of practical experience. Most students do not have the chance to develop an actual project. Traditional teaching methods tend to make students lose interest in this practical course, and therefore, combining theory with practice and increasing practice teaching is the emphasis of the teaching reform.

In the actual practice process, it is important to train the students' abilities of cooperating, planning, self-study, communicating, expressing and leading. In fact, these abilities could be improved only by practice. In the curriculum, we use project-driven teaching mode. Students are required to manage a project using the learned principles in teams of 3-5 persons. The teams are self-selected by the students and there must have a team leader. Each team should manage a software project with given tasks. This project is a design-build type project. The project teams can choose their individual project topics from a catalogue of ideas or choose other topics under supervising of the teacher.

In the practice, students should use some tools of project management to help them finish their work. Such as, tools for project management--Microsoft Project, tools for WBS--WBS Chart Professional, tools for process management--Pert Chart Expert etc. These tools can help us manage the software process more efficient.

Assessment is an important part of this project-driven teaching model. We have already set up an explicit criterion for this practice assessment. In the experiment guidebook, we make a list to let students know what they must complete and submit at each phase, and how they can get high scores. At each phase, we choose one team to present their process of project management in classroom; other teams can comment and ask questions for their presentation. At last, the teacher will review and summarize for the presentation. After the presentation, students should revise their project plan according to the comments and suggestions. At the final acceptance phase, each team should submit their project, documents and prepare for an oral defense.

Students are assessed individually as well as in teams. Each team and each student must have an individual oral presentation for their work. A students' final course grade depends on their written reports, daily performance and oral presentations. In this course assessment, not all team members get the same grade, 40% of the grade is on the basis of team

performance and 60% relies on individual achievement. For example, one student who fulfills a lot of work gets A, but another student who does not do anything and are often absent for the class obtain a D grade. We have differentiated pass grades as A, B, C, and D, with the purpose to increase student interest in the course by rewarding those who perform better than just acceptable [3].

In order to make the project accomplish successfully and get high scores, students must finish the assigned tasks, because the report is ongoing according to roles. Through project practice, students could gain a better understanding of the concepts and principles presented in textbooks. These course reforms have proved to be a success with majority of the students giving positive feedback.

V. Reform on Teachers' Ability

The knowledge and practical ability of teachers has a deep effect on engineering education, and therefore, universities and colleges must have excellent teachers with professional capacity and practical skills. In the core courses of software engineering major, we have an excellent teaching team which includes two professors, two associate professors and one lecturer. Our school is an application-oriented university; therefore, most teachers are required to have practical experience in enterprise. To encourage that, teachers are usually given half a year to full-time study or communication in enterprise. This experience is a necessary condition for teachers to promote their academic titles. Teachers can choose their own cooperative practice companies, but before going, they should apply for approval. After the practice, they need to make a presentation and report the results.

Several methods are used to improve teacher's professional and application ability. The first is encouraging teachers to become "double-professionally-titled teachers". The second is regularly carrying out curriculum teaching and research activities, to discuss the method and means of teaching, and allowing teachers to learn the teaching experience from each other. Third is to provide various opportunities and financial support to encourage teachers to attend profession training and all kinds of important education conferences about teaching reform and practice reform etc, where they can communicate and discuss with colleagues in other universities or in IT Company. Teachers should be encouraged to update their knowledge and teaching materials to keep pace with times.

VI. Summary

CDIO represents *Conceive, Design, Implement and Operate,* in order to develop the ability of innovation design, communication, teamwork spirit, ability of lifelong learning etc. This paper introduces some reform ideas for software project management curriculum based on CDIO in our school. Different from the traditional teaching mode, we use heuristic teaching methods, like scenario animations show, project-driven practice, actual cases study, teamwork and teachers' encourage policy etc. The results show that all these reformations can enhance students' study interest and improve the teaching quality, and tha it is suitable for application-oriented universities and colleges.

Acknowledgment

This paper is supported by National Natural Science Foundation of China (NFSC) under Grant No. 61502294, The Natural Science Foundation of Shanghai under Grant No. 15ZR1415200, The Key Disciplines of Computer Science and Technology of Shanghai Polytechnic University under Grant No.XXKZD1604, and Internet Technology of CERNTER under Grant No.NGII20150609.

References

[1] Abdul Rauf, Ibrahim Albidewi. Improvement in Course Curriculum for Software Project Management. The 3rd International Congress on Engineering Education (ICEED). 2011, pp125-127.

[2] Zhi Han, Zhen-Hong Zhang, Xing-Juan Li. Instructional Reform in Software Engineering based on CDIO. Computer Education. 2010(11): 56-59.

[3] Göran Gustafsson, Johan Malmqvist and Dava J. Newman etc. Towards a New Model for First-Year Introductory Courses in Engineering Education Programmes.2002. (paper is available at www.cdio.org).

[4] Crawley, E., *The CDIO Syllabus: A Statement of Goals for Undergraduate Engineering Education.* MIT CDIO Report #1, Dept of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.

[5] PMI 2004. A Guide to the Project Management Body of Knowledge (PMBOK Guide) Third Edition.

[6] Kristina E., Josefin T., Madelaine E.and Åsa W. Student involvement in principled change: Understanding the student experience. (paper is available at www.cdio.org).

[7] Liping Li, Shiming Zhang, Yueping Wu. Teaching Reform and Practice on Software Project Management Curriculum. ICCSE 2015, Cambridge, England. July 22-24, 2015.

[8] Gabriele Bavota1, Andrea De Lucia etc. Teaching Software Engineering and Software Project Management: An Integrated and Practical Approach. ICSE 2012, Software Engineering Education. Zurich, Switzerland. 2012, pp 1155-1164.