World Transactions on Engineering and Technology Education
Vol.14, No.1, 2016

Reform of a Java curriculum based on a CDIO active project-driven method

Feng-hua Cao

Inner Mongolia University of Finance and Economics
Inner Mongolia, People's Republic of China

ABSTRACT: Problems in the teaching of Java courses include: knowledge isolation, the lack of an integrated course, lack of student practice ability and the lack of reflection on the teaching. In view of these problems, an active project-driven, CDIO-based method was proposed and applied to the Java curriculum. The reform covered the curriculum redesign, ability objectives matrix construction, project implementation, reflection on course teaching and student academic performance assessment. This work provides a reference for engineering teaching reform, particularly, for Chinese universities and colleges.

INTRODUCTION

Conceive-Design-Implement-Operate (CDIO) is an advanced teaching mode suitable for engineering education, which was developed at a number of universities, including the Massachusetts Institute of Technology. The CDIO method is an implementation of a product lifecycle, from R&D to operation, so that students can learn the actual engineering practice [1-3]. For some courses, however, there are no specific capability requirements and implementation methods. This requires the instructor to formulate the CDIO targets and implementation method. The CDIO mode is one where the project is taken as central, the teacher as guide, and the student as subject. The Conceive-Design-Implement-Operate method has an integrated curriculum, with learning driven by an active project; it also involves the setting up of an ability matrix, a *guarantee* and teaching reflection. Java course teaching reform is taken as an example, so as to elaborate on implementing CDIO core ideas in science and engineering teaching.

PROBLEMS IN TEACHING PREVIOUS JAVA COURSES

Java courses include Java programming and Java Web development, with a focus on theory and practice. Traditional teaching tends to lead to more theory and less practice. The author surveyed academic colleagues to identify problems with the traditional teaching methods, and has found:

- Lack of an integrated course leading to isolated knowledge:

  A teacher cannot simply impart knowledge mechanically during the teaching of a subject. The teaching must integrate the professional knowledge of the subject, taking note of the personal ability and interpersonal skills of the students.

- Lack of student practice ability:

  Traditional examination-based assessment methods focus on theoretical knowledge and not on its practical application. As a result, the students fail to have a complete understanding of the knowledge, and cannot flexibly apply the knowledge to the solving of a practical problem. In addition, they lack analytical and operational abilities. When encountering a large-scale and complicated software project, students cannot conceive, design, implement and operate in an effective and rapid way.

- No reliable basis to determine whether a learning goal has been reached:

  Most learning goals are simply a list of knowledge points that will be taught. Such teaching goals not only restrain course teaching, but weaken teacher responsibility, and are counterproductive to the cultivation of student ability.

In addition to mastering knowledge (i.e. the knowledge points), such knowledge is just the basis for its flexible application as the final goal. Whether the teaching goal has been achieved inevitably will be an unclear subjective judgment. So far, few courses at the Inner Mongolia University of Finance and Economics have set the cultivation of student ability as an objective.

- Lack of reflection on the teaching:

  Reflection on the teaching process refers to the dialogues and discussions about problems in the course teaching. Reflection is an interactive activity, which combines both the teacher and students as a group and pays attention to the successes, failures, co-operative learning and possibilities for improvement. The lack of effective reflection ultimately affects students' understanding and the completion of teaching goals. The lack of reflection also renders difficult any redesign of the teaching.

Noting the problems mentioned above, some teaching reform of Java courses has been undertaken, such as the project-driven method [4-6] and the case teaching method [7-9]. Despite this, deficiencies still exist in cultivating team spirit, engineering understanding, enterprise demands and understanding the engineering environment. Conceive-Design-Implement-Operate also has been applied to the teaching reform of Java courses [10-12]. Although such reforms lead to a better imitation of the engineering process, deficiencies still exist in the implementation of CDIO educational philosophy.

In the redesign of teaching of Java courses at the Inner Mongolia University of Finance and Economics, *project-driven* was transformed into *active project-driven student-oriented learning* in the hope of improving student initiative and creativity. The CDIO core spirit should be implemented, with the project as the principal line, the teacher as guide and the student as subject. The aim is to cultivate a student's individual and vocational ability in thought, expression, learning and co-operation, as well as to train a software engineer with endogenous capacity and practice ability.

CDIO ACTIVE PROJECT-DRIVEN LEARNING

Integrated Curriculum

Guided by modern education theory and based on the requirement of cultivating Java software engineers, the Java curriculum integrates the inter-related courses of the computer specialty into an organic curriculum. The curriculum is composed of three types of courses, viz. preparation, current and follow-up courses. More specifically, the preparation course covers computer engineering and the foundations of programming. The current course can be divided into Java programming, Java Web development and data structures. A follow-up course refers to database principles, computer networks and software engineering.

After the preparation course, students will be equipped with the ability to learn Java programming and Java Web development. When students take Java as an elective course, they can also take data structures at the same time.

The interaction of the two courses will cultivate the ability to develop complicated programs. In active project-driven learning, students are required to learn the relevant knowledge of database principles, computer networks and software engineering, independently. The Java-related courses have been integrated into an organic whole.

Through the association of knowledge points among courses, the learning on the current course will enhance a student's understanding of the preparation course, and lay a foundation for the follow-up courses. Thus, it can be seen that the learning of Java and Java Web courses is an important key to the preparation courses and follow-up courses. The curriculum has integrated the subject knowledge and cultivated student abilities.

Learning Objectives

The CDIO ability matrix sets standards for learning from an engineering perspective. These standards list the abilities that should be possessed by a qualified engineer. During the teaching reform, the CDIO ability matrix should be formulated to reflect the learning objectives of the course and confirm which standards are followed in CDIO. In this way, the designer of the course can consider the detailed learning objectives and provide a reliable basis for assessment.

The curriculum for the Java courses will have to give consideration to preparation for the follow-up courses. Under CDIO, the learning objectives of the Java course can be divided into two parts: technology objectives and educational objectives. Technology objectives focus on the improvement of the student's technological level. In technology objectives, there are six components:

- Document editing;
- Platform development (e.g. Android, Eclipse);
- System modelling, i.e. UML;
- Technology of Java and Java Web;
- Analysis and code design principles.

Training objectives in Java courses are in 28 micro-levels, under four major levels. The objective matrix is shown in Table 1.

Table 1: CDIO ability matrix for Java courses.

CDIO Ability		Name	Description
Technology knowledge and reasoning	1.2	Basic knowledge of core engineering	Related tools involved in basics of computer engineering and basics of applied computing
	1.3	Basic knowledge of professional engineering	Basics of programming
Personal ability, occupational ability and attitude	2.1.1	Problem identification and presentation	Assess project feasibility, grasp overall goal
	2.1.4	Probabilistic analysis	Engineering cost-benefit analysis and risk analysis
	2.1.5	Solution and suggestion	Solution, summative suggestions, room for improvement
	2.3.1	Comprehensive thought	Identify and define a system, system behaviour and system unit; identify systematic society, enterprise and technological background; identify the interaction between the system and the outside environment, and the influence of system behaviour
	2.3.2	Systematic presentation and interaction	Discuss the abstraction required by definition and system modelling; identify the behaviour and function characteristics presented by the system (intentional and non-intentional), recognise the important connector between system units; evolution of the system in time
	2.3.3	Confirmation of primary-secondary factors and emphasis	Find out the driving factor of the integrated system; explain the resource allocation required to solve the problem
	2.3.4	Assistance, judgment in solving problems	Balance various factors in the system, eliminate strained relations, optimise system
	2.42	Persistence and flexibility	Difficulty encountered in solving a problem
	2.43	Creative thinking	Creative design of mobile game
	2.44	Critical thinking	Analyse the stated questions
	2.47	Management of time and resource	Order of task schedule
	2.5.1	Professional ethics, honesty, sense of responsibility	Dare to adhere to the courage of undertaking risks; understand and accept mistakes; undertake due obligations; recognise co-operator's work; be responsible for work
Interpersonal communication, teamwork and exchange	3.1.1	Team organisation	Confirm the members' roles and responsibilities
	3.1.2	Teamwork operation	Confirm the objective and schedule; carry out effective exchange; realise project planning; organise and implement; formulate problem solution
	3.2.6	Oral expression and interpersonal communication	Language and non-language exchange; effective question answering
Conceive, design, implement and operate system under the enterprise and in the social environment	4.2.3	Technology entrepreneurship	Meet the opportunity of technology entrepreneurship; identify technology that can innovate new products and new systems
	4.2.4	Work in an organisation	Participate and organise
	4.3.1	Systematic goals and requirement setting	Establishment objectives and requirements
	4.3.2	Definition function, concept and structure	Confirm necessary system functions; make use of the right technological level
	4.3.3	System modelling and goal completion	Build model, discuss life cycle and cost; accept or reject; iteration
	4.3.4	Management of development project	Project management
	4.4.1	Design process	According to the objectives of the integrated system, as well as the requirement of each module and component, select initial design solution; implement suitable optimisation under constraint conditions; iterate until constriction; reach the final design; can adjust to the changes if demanded

	4.4.3	Application of knowledge in design	Apply technology and scientific knowledge (e.g. psychology and aesthetics) to carry out creative and critical thinking, and solve problems; discuss the acquisition of design knowledge
	4.5.3	Software implementation procedure	Modular decomposition; discuss algorithm; system construction
	4.5.5	System testing	Test system
	4.6.1	Operating design and optimisation	Describe operational performance; the objectives; cost and value

Implementation

Apart from the teaching of theory in the classroom, active project-driven learning is an important means by which to guarantee the completion of course learning objectives. The implementation process of CDIO active project-driven learning is composed of the following six basic links. Each link includes a specific goal. Link I is the traditional theory teaching link; in Link II, tasks will be given and students will form a team. Link III to Link VI are the complete life cycle, from product R&D to operation.

Because traditional teaching plays a small role in ability cultivation, it does not cover the ability objectives effectively. The cultivation of ability in CDIO can only be realised by the project. Students training in the project cycle can both develop their technological ability and cultivate internal non-technological abilities. Different abilities are the focus of each link. Although emphasis is laid on the objectives in each link, overlaps may exist.

- Link 1

  Teaching of theory for Java courses, including background introduction, introduction to development tools, systematic construction methods, basic knowledge of course theory and philosophy, new technology related to Java, and introduction to new applications. (Matching objectives: 1.2, 1.3, 4.2.3).

- Link II

  The teacher's tasks can be divided into project introduction, exploration of project objectives, exploration of target abilities, and team organisation. Student tasks can be divided into organising the team independently, knowledge required in the project and supplementary technology. (Matching objectives: 3.1.1, 4.2.4).

- Link III

  Systematic conception includes the analysis of project objectives, feasibility analysis, budget analysis and cost control (budget, consumable items, personnel allocation, software resources), project report I (project objectives and feasibility exploration and assessment, system creativity exploration and assessment, budget exploration and assessment). (Matching objectives: 2.1.1, 2.1.4, 2.4.2, 2.4.3, 2.4.4, 3.1.2, 4.3.1, 4.3.2, 4.3.4).

- Link IV

  System design covers the formulation of a solution, developing plans, project report II (project design solution elaboration and assessment, project risks elaboration and assessment). (Matching objectives: 1.2, 1.3, 2.1.5, 2 3.1, 2 3. 2, 2 3.3, 2 3.4, 2 4.2, 2.4.4, 2.4.7, 3.1. 2, 4.3.3, 4.4.1, 4.4.3, 4.5.3)

- Link V

  System implementation can be divided into project reflection, project iterative construction and project report III (mid-term checking). (Matching objectives: 1.2, 1.3, 2.4.2, 2.4.4, 2.4.7, 2.5.1, 3.1.2, 4.3.4, 4.4.1, 4.4.3).

- Link VI

  Operation and assessment includes project test and acceptance inspection, project report IV (system demonstration, system cost analysis, system profitability analysis and multi-level analysis); multi-level assessment includes student assessment, student of the same group assessment, student of different group assessment and simulation user assessment. (Matching objectives: 1.3, 2.3.1, 2.4.4, 3.1.2, 3.2.6, 4.2.3, 4.5.5, 4.6.1).

Reflection on Course Teaching

Reflection is one of the most important aspects of teaching. For students, reflection can inspire them to rethink answers to problems and, thereby, enhance their learning. It also helps students in deep learning and strengthens their understanding of theoretical knowledge.

For the course teacher, reflection is an effective means by which to tease out problems in the teaching process. From reflections on the teaching, teachers can understand whether the knowledge they teach is accurate; whether the knowledge is required by students; whether the students understand the knowledge; and whether the teaching objectives have been achieved.

By reflecting on the teaching, teachers can conduct an objective and rational assessment and evaluation on their teaching. This provides a basis for teachers by which to improve their performance.

During the project report meetings I–IV of the project implementation links III, IV, V, VI, questions can be proposed to members of each group, to stimulate reflection on the answers. The question list for Java course students is shown in Table 2.

Some of the questions are open ended and the answers are diverse. The point is to lead students to reflect actively. The reflection question list for Java course teachers is shown in Table 3. The reflection is divided into three phases, viz. the early phase of teaching, middle phase and end phase.

Table 2: Question list for student reflection.

Phase point	Question	Matching ability objectives
Project report meeting I	Question 1: How do you understand the system requirements? How do you analyse customer requirements? What is your preparation work? How do you decide the budget for the preparation at the early stage?	2.1.1, 2.1.4
	Question 2: How do you select your partners?	2.1.1
	Question 3: As to innovation in the system, what iss your design inspiration? How do you think that your system is creative and attractive?	2.4.3
	Question 4: What is the expected earning potential of the system?	2.1.4
Project report meeting II	Question 1: How do you select your development tools and development technology? Have you considered the programming using the development tools and technology?	1.2, 1.3
	Question 2: What is the core part of the system? Have you considered system extendibility when designing the system?	2.3.1, 2.3.3, 2.3.4
	Question 3: How do you determine the project's budget and time management? Have you a project schedule and plan? Do you act according to this schedule?	2.4.7, 3.1.2
Project report meeting III	Question 1: How do you understand the requirements for making good software and files? Will you construct the system according to the documentation? If you construct the system by changing certain regulations in the documentation, what is the reason?	4.3.4
	Question 2: Have you encountered current development difficulties? If yes, what's your solution? If no, what is your solution to the next step? Have you ever considered overturning the whole system and starting again?	2.4.2, 2.5.1, 4.4.1
	Question 3: How important is the preparation work, is it worth it?	4.3.4
	Question 4: What is the association between the Java course and other courses of the curriculum?	1.3, 4.3.4
Project report IV	Question 1: Compared with other technology of the same kind, what is the advantage of your technology? Is it convenient for your system development?	4.2.3
	Question 2: What is the reason for success or failure of the analysis? How does it relate to the problem? Where do you think great effort should be made to ensure a successful project?	2.4.4
	Question 3: What have you learned? How do you summarise and apply this knowledge to other projects?	4.2.3
	Question 4: Based on others' assessment, do you think the system you have developed has achieved the expected outcomes?	4.6.1
	Question 5: What is the competitive advantage of your product? Is it the algorithm, better functions, or others?	2.3.1

Table 3: Question list for teacher reflection.

Phase Point	Question	Objective
Early phase of course teaching	Question 1: Whether redesign is required given the experience of course teaching of the last semester and the teaching plan and objectives of this year?	Teaching redesign and improvement
	Question 2: Whether the teaching experience of last semester is used to strengthen weak links in the courses?	Teaching redesign and improvement
	Question 3: Whether new technology and knowledge have been added to the course?	Teaching redesign and improvement
	Question 4: How to set up effective project categories? How to attract students to participating in projects?	Ensure the completion of course objectives
Middle phase of course teaching	Question 1: Based on the learning result of the current phase, whether the course schedule and teaching plan should be adjusted?	Ensure the completion of course objectives
	Question 2: Are the students interested in participating in the project? Whether to increase the difficulty for a group with excellent performance?	Ensure the completion of course objectives
	Question 3: What is the major difficulty faced by the groups? Do they have methods to solve the problems?	Ensure the completion of course objectives
End phase of course teaching	Question 1: Discuss with other teachers, whether the teaching result has reached the teaching objective? If yes, what is the reason for success? If no, what is the main reason?	Teaching redesign and improvement; improve education and teaching level; develop educational concept with unique characteristics
	Question 2: Based on the student's assessment, which CDIO ability has been trained? Which ability is required for further improvement? Combined with students' and teachers' understanding, whether adjusting the ability matrix is required?	Teaching redesign and improvement
	Question 3: What is the success and deficiency of the teaching?	Teaching redesign and improvement; improve education and teaching level; develop educational concept with unique characteristics

Students' academic performance and project implementations are effective measurements that reflect student ability. However, these two measurements are rigid and incomplete evaluation indicators. Thus, an evaluation indicator (or indicators) that can reflect students' overall ability is needed. The result of reflection is one such. Although the answer to a student's reflection question is non-unique, the expected central idea and emphasis of the answer to each question is clear. Then, based on whether a student's answer deviates from the central idea and emphasis, the student's ability can be measured. Some possible emphases to measure a student's ability for the question list for student reflection are provided below. If the ability objective has been reached, mark Y, if no, mark N.

- Project report meeting I

  Question 1: if the demand is not clear or the preparation at the early stage is insufficient, mark N; does not think there is a requirement in the enterprise or society, mark N;
  Question 2: no clear partner selection, mark N;
  Question 3: no innovation, highlight or attractiveness in the project, mark N;
  Question 4: no expected learning from the project or no business considerations in the project design, mark N.

- Project report meeting II

  Question 1: can explain the core ideas, can understand the importance of design, mark Y;
  Question 2: improve the extensibility of the system by adopting a target-oriented idea, mark Y;
  Question 3: no budget, no project schedule, mark N.

- Project report meeting III

  Question 1: update the software documentation during the iteration process, mark Y;

Question 3: understand the importance of demand analysis, mark Y.

- Project report meeting IV

Question 2: understand the key factors of project implementation, mark Y.

Student Academic Performance Assessment

In the active project-driven CDIO-based engineering education mode, the performance assessment of Java courses can be divided into four: theoretical examination (30%), project implementation (40%), reflection (20%) and daily performance (10%). From the grading of these four parts, it can be seen that project implementation is the priority, and theory assessment is subsidiary. Together with reflection and daily performance, the student's ability is comprehensively evaluated.

CONCLUSIONS

In this study, an active project-driven learning method has been outlined, based on the Conceive-Design-Implement-Operate standards (CDIO). It takes CDIO as the central guiding philosophy and active project-driven learning as the main means. A Java curriculum ability matrix, according to CDIO, was set up to elaborate the specific objectives in detail.

This guides the teacher's teaching direction and provides a clear definition of the completion of teaching objectives. The active project-driven teaching mode has changed the situation, whereby, students in class accept knowledge passively. This mode has confirmed a student's dominant position, increased student enthusiasm and improved student's independent learning ability.

The student can search for materials according to the project requirements, and lead the project implementation process independently, to turn passive into active learning. In this way, a student's subjective initiative is stimulated. In addition, during the process of project implementation, the student's ability in project management, communication and interaction are improved. Thus, a sound foundation is laid for their work in the years ahead.

REFERENCES

1. Berggren, K., Brodeur, D., Crawley, E., Ingemarsson, I., Litant, W. and Malmqvist, J., CDIO: an international initiative for reforming engineering education. *World Trans. on Engng. and Technol. Educ.*, 2, **1**, 49-52 (2003).
2. Lin, G. and Tang, Y., Application of CDIO in practice and training of an engineering automation specialty. *World Trans. on Engng. and Technol. Educ.*, 13, **3**, 407-411 (2015).
3. Chen, Y., Evaluation method for CDIO project-based teaching, with total-process multi-assessment. *World Trans. on Engng. and Technol. Educ.*, 13, **3**, 291-295 (2015).
4. Gao, J., The application of case teaching method in the Java programming language teaching. *Hunan Tax College Academic J.*, 4, 53-55 (2015).
5. Zhao, C. and Liu, X., The practice and exploration of case teaching in the teaching of Java programming course. *Times Educ.*, 11, 17-18 (2015).
6. Liu, C. and Zhang, Y., Research and practice on the teaching of JAVA programming course in combination with practical cases. *Computer CD Software and Application*, 06, 221-222 (2014).
7. Peng, B., Jiang, L. and Luo, H., Application of project driven teaching in Java language programming. *J. of Mudanjiang Institute of Educ.*, 11, 92-93 (2014).
8. Li, Y., The Java programming curriculum teaching design based on the project driven method. *Asia Pacific Educ.*, 16, 90-91 (2015).
9. Li, J., Research on Java teaching model based on task driven and project leading group. *J. of Liuzhou Teachers' College*, 03, 120-123 (2015).
10. Jiang, D. and Sun, H., On CDIO-based active project-driven learning method journal of cases. *J. of Shantou University (Humanities & Social Sciences Edn.)*, 4, 79-82 (2012).
11. Yang, H., Feng, J. and Li, S., The Java project training course teaching reform - based on CDIO model. *J. of Liaoning Police Academy*, 06, 109-112 (2012).
12. Ma, X., Han, N. and Zhang, J., Java language curriculum design under the concept of CDIO. *Higher Educ.*, 12, 12-14 (2010).