Teaching software testing based on CDIO

Songhao Jia & Cai Yang

Nanyang Normal University
Nanyang, Henan, People's Republic of China

ABSTRACT: The CDIO (Conceive - Design - Implement - Operate) model was adopted for the teaching of software testing as part of engineering education. This new method combines classroom teaching with practical teaching by the use of projects. The application of this method allowed to improve the quality of training offered in software testing. Practice has shown that this teaching mode can arouse the enthusiasm of the students and, results indicate that it can advance their abilities in theory and practice.

INTRODUCTION

With the development of technology, computer software has been subject to rapid change. However, among the many challenges facing software development, there have been issues, such as poor quality and functions that are too complex. Software testing is the final means by which software quality can be ensured. Therefore, Nanyang Normal University has focused on reforming the software testing course. Issues within the software testing course have included a weak practical or practice component; a shortage of capacity for handling students' hands-on activities. These issues require the University to continue to reform the existing course, so as to offer excellent training standards, which will produce well-qualified students to meet the demands of society.

The Conceive - Design - Implement - Operate (CDIO) model is widely used in education. This model has been studied by many experts; for example, Karl-Fredrik Berggren analysed the relationship between the engineering education reform strategy of CDIO and co-operative working [1]. John Malmqvist developed a new system for CDIO in education, using the CDIO training scheme, curriculum system, teaching mode, environment and evaluation system [2]; and Mohan et al proposed that a project in education be carried out that leads to professional skills [3]. These studies mainly focus on the concept of CDIO, its application and case studies. There are few courses run that combine CDIO and software testing.

Therefore, the authors of this article have continued to explore the teaching of software testing courses based on the basic concepts of CDIO. The authors have developed a new method for teaching software testing based on CDIO, which has these basic characteristics: a project is the main focus, with the teacher as the guide and students are the target. This teaching method sets out to train students in thinking, expressing themselves, learning and collaborating. The method places emphasis on cultivating personal ability and practical skills. Experience has shown that this new teaching method can improve classroom teaching and boost students' expertise in practical work.

CDIO ENGINEERING EDUCATION THEORIES

The CDIO engineering education theory is the latest reform in international engineering education in recent years. A new educational philosophy and implementation of system engineering was developed jointly by the United States Massachusetts Institute of Technology, and the KTH Royal Institute of Technology, Sweden; Chalmers University of Technology and Linköping University, Sweden. Conceive - Design - Implement - Operate emphasises real world systems and processes by which to learn the theory and practice of engineering [4].

Consider the four phases of CDIO: conceiving, includes the requirements and the necessary technology, and the strategy, rules and regulations [5]; designing, determining the plan and the details of what will be implemented,

e.g. algorithms; implementing, that is, turning the project into real products, and which includes manufacturing, compiling, testing and verifying; and the final stage, operating, is the use of the products to determine if they perform according to the requirements and it includes, maintenance, updating and eventual decommissioning [6].

The CDIO theory is the embodiment of the principle of *learning by doing* and the *project oriented teaching*. There are 17 skills or abilities in CDIO. These are shown in Table 1. The CDIO engineering education theory emphasises that learning through a practical project will cultivate students' abilities in theory and practice, communication and collaborative learning ability [7].

Table 1: CDIO - student skills.

Skill level	Skills
1 Speculative knowledge	1.1 Basic knowledge 1.2 Basic knowledge of the core project 1.3 Basic knowledge of advanced engineering
2 Occupational skills and personal skills	2.1 Engineering reasoning and problem-solving skills 2.2 Experimentation and skills in finding knowledge 2.3 Systematic thinking 2.4 Personal skills and attitudes 2.5 Occupation skills and attitude
3 Teamwork and communication	3.1 Teamwork 3.2 Communication 3.3 Language skills
4 Conceive, Design, Implement and Operate in an enterprise and social environment	4.1 External and social environment 4.2 Enterprise and business environment 4.3 The conception and system engineering 4.4 Design 4.5 Implement 4.6 Operate

INSUFFICIENCY OF SOFTWARE TESTING COURSES

At present, in many college computer departments, their *software testing* courses are not entities in their own right, but are encapsulated within software engineering. Software testing courses at some universities are focused too much on explaining software testing theory and introducing a test method, rather than focusing on the practice of software testing. Hence, students lack systematic training [8].

Existing problems in the teaching of software testing are:

- Disconnection between theory and practice leads to less interest by students;
- The classroom teaching and the content of the work in business are not the same. This disadvantages students with their employers;
- Students' lack of testing experience may lead them to being unable to grasp the process and steps of testing;
- Students' development experience may be insufficient for them to understand test-driven development;
- The software testing laboratory is limited, and so classroom teaching and practice are not closely related [9].

SOFTWARE TESTING TEACHING METHODS BASED ON CDIO THEORY IN ENGINEERING EDUCATION

Noting the present situation of software testing teaching, a method is presented in this article, based on the CDIO theory, for the teaching of software testing that is innovative and integrated. Improving students' practical ability in software testing will better meet the needs of industry. The teaching method is shown in Figure 1.

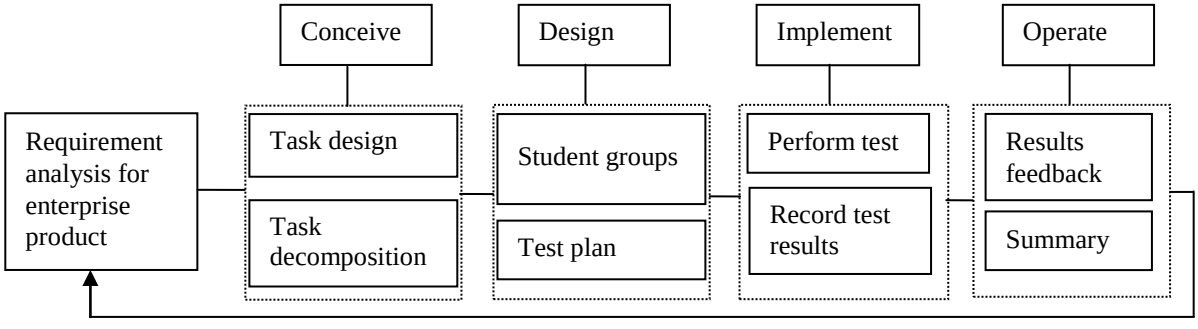

Figure 1: Software testing education based on CDIO.

According to the engineering education ideas of CDIO, the new course will fuse software testing for each practical project, and will integrate the contents of the curriculum and strengthen experimentation as part of the training process. The practical project used for the teaching will give students actual experience of a project, and so the students will accumulate work experience at college. At the same time, students' collaborative learning, interpersonal communications and innovation during the project will be cultivated, by focusing on their soft skills [10]. The course has the following characteristics.

Students Play the Main Role

In constructivist learning environments, as compared to traditional teaching, the status held by teachers and students has changed. During class, teaching is now student-centred. Teachers must give full expression to students' initiatives, and let the students be actively involved in the process of learning. This can be very good for improving classroom teaching.

Interest is the best teacher; so the students' interest is of great significance in teaching. In situ teaching can be effective in stimulating students' interest. For example, to explain a test plan, it is best to set up more test cases for students to address.

Improving Curriculum Content

Because the teaching of software testing is practical and demanding, much research has been carried out by the authors on an appropriate curriculum. This shows that, during classes, it is best to include real-world examples and case studies to underpin knowledge. Students would, then, not only master the basic concepts but also accumulate real-world experience.

Project Training

At present, in teaching software testing, many universities use projects, but problems remain. For example, the choice of project may not be appropriate; the practical sessions may not be relevant to the project. Different projects may require different approaches. However, students may then not acquire an overall understanding of software testing by focusing on specific issues for specific projects.

The complete teaching and training of software testing by using a project is considered and discussed in this article. This enables students to have a complete understanding of software testing in project development. Through the project training approach, students will learn software testing and integrate knowledge and theory.

Strengthening Teaching

In addition to classroom teaching, practical teaching is most important in a software testing course. There are two components: practical teaching and practice in an enterprise. Practical teaching requires the following:

- *Planning*. Each computer class should have a clearly defined task or tasks and a plan. The task selection must be targeted at the requirements. Unit testing deals with the detailed formulation of the test suite which, for example, includes parametric tests.
- *Strengthen communications with students*. Students are formed into study groups. These groups enhance learning by promoting communication and information exchange.
- *Teacher overview*. When students are carrying out the practical content of the course, teachers should check their progress, and summarise progress. At the same time, teachers can deepen business experience and strengthen the co-operation skills expected by enterprises.

Furthermore, students should have the opportunity to understand the actual software testing business and software testing staff. Hence, the students are sent to Beijing each year to work in a company to improve their practical abilities.

Internships

- An internship with a company or organisation offers an opportunity for a student to engage in practical work as agreed between the university and the company. Internships for students require specifications and documentation of the work undertaken, which allow the students to use their skills outside the institution and according to the standards expected in the marketplace.
- The company or organisation chosen should provide a good training environment and have a good supervisor. The individuals who supervise the students should ensure they understand the work and make appropriate ongoing adjustments to plans.
- During training, the institution can give the enterprise or organisation guidelines as regards supervision, as well as ongoing suggestions to ensure the students are progressing satisfactorily.
- After the training period, a project evaluation of the students' training should be carried out, and an assessment completed.

Soft Skills

Software testing skills are much in demand in industry. It not only requires the applicant to have theoretical knowledge and practical ability, but also an appropriate aptitude and *soft skills*; current teaching methods are lacking in this area. Training students in attitude and *soft skills* increases their chances of employment:

- Pay attention to teamwork and communication skills - for the software testing industry, communication skills are very important. Software testing requires team co-operation and good communication skills, enabling a new member to integrate quickly and effectively into a team. A good engineer should be able to express his/her thoughts clearly and effectively. The practical teaching methods at Nanyang Normal University are oriented to developing students' skills in communications, team working, and awareness of the work environment. A good test engineer should have a well-developed sense of responsibility, as well as patience and insight into the technical possibilities.
- Encouraging students to doubt the system - the purpose of software testing is to find faults in the system being tested. Therefore, students are encouraged to question all and to use all possible means to verify software. Even the simplest function may have user significance and may require, for example, boundary value testing or other means to remedy any problems.

CONCLUSIONS

As an engineering education model, CDIO (Conceive - Design - Implement - Operate) emphasises that students should have basic knowledge, but it also cultivates abilities of team co-operative and the application of theory. An active project-driven learning method for teaching software testing is outlined in this article, based on the CDIO engineering education model, with CDIO as the guiding principle and active project learning as the main enabler.

The teaching of software testing based on CDIO focuses on encouraging students' all-round skills. The new methods of teaching puts students' future occupational roles at its heart. Attention is also paid to theory and practice. The teaching of practical sessions shows that this kind of teaching cultivates students' creative thinking and practical abilities.

REFERENCES

1. Berggren, K-F., Brodeur, D., Crawley, E.F., Ingemarsson, I., Litant, W.T.G., Malmqvist, J. and Östlund, S., CDIO: an international initiative for reforming engineering education. *World Transactions on Engng. and Technol. Educ.*, 2, **l**, 49-52 (2003).
2. Malmqvist, J., Edström, K., Gunnarsson, S and Östlund, S., The application of CDIO Standards in the evaluation of Swedish engineering degree programmes. *World Transaction of Engng. and Technol. Educ.*, 5, **2**, 361-364 (2006).
3. Mohan, A,, Merle, D., Jackson, C., Lannin J. and Nair, S.S., Professional skills in the engineering curriculum. *IEEE Transactions on Educ.*, 53, **4**, 562-571 (2010).
4. McBriar, I, Smith, C., Bain, G., Unsworth, P., Magraw, S. and Gordon, J.L., Risk, gap and strength: key concepts in knowledge management. *Knowledge-Based Systems*, 16, **1**, 29-36 (2009).
5. Ichiro, K. and Mai, S., A study on information recommendation system that provides topical information related to user's inquiry for information retrieval. *Proc. 2006 IEEE/WIC/ACM Inter. Conf. on Web Intelligence and Intelligent Agent Technol.*, 385-388 (2006).
6. Jafari, M., Akhavan, P. and Nouranipour, E., Developing an architecture model for enterprise knowledge: An empirical study based on the Zachman framework in Iran. *Management Decision*, 47, **5**, 730-759 (2009).
7. Bourque, P., Oligny, S., Abran, A. and Fournier, B., Research and practice of dual professional and compound software engineering curriculum. *The Computer Educ.*, 9, 64-68 (2010).
8. Benlian, A. and Hess, T., Opportunities and risks of software as a service: findings from a survey of IT executives. *Decision Support System*, 52, **l**, 232-246 (2011).
9. Schnittka, C., Bell, R. and Richards, L., Save the Penguins: teaching the science of heat transfer through engineering design. *Science Scope*, 34, **3**, 82-91 (2010).
10. Mitchell, J., Canavcon, V. and Smith, J., Problem-based learning on education communication systems: student perceptions and achievement. *IEEE Transactions on Educ.*, 53, **4**, 587-594 (2010).