Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Incorporating Innovation-Oriented Education into the Software Engineering Course

Yu Gao
Department of Computer
Zhejiang Ocean University
Zhoushan, China 316000
gaoyu@zjou.edu.cn

Xiangzhong Feng
Department of Computer
Zhejiang Ocean University
Zhoushan, China 316000
fxzfeng@yahoo.com

***Abstract*—The strong sense of innovation represents the prominent characteristics of the software engineering course in contrast with other major courses of computer. It decided that the course itself was the best carrier of implementing innovation-oriented education. In order to incorporate innovation-oriented education into software engineering courses and foster students' sense of innovation, new thoughts of teaching this course was proposed.**

***Keywords*—Software Engineering, Innovation-Oriented Education, Carrier, New Thoughts**

I. INTRODUCTION

As one of the major courses of computer, software engineering course [1] plays an important role in courses system of computer major. It introduces the knowledge concerning the development, maintenance and management of software. Aim of this course is making students acquire the knowledge, so as to lay a solid foundation for their future career.

Innovation-oriented education is of particular importance to the students from computer major. As computers is developing at an amazing speed, the related knowledge keeps on renewing and would become old-dated several years later. Then new problems would come out, unable to solve with present methods and technology and new ones were thus called for. It is, therefore, necessary to instill innovation awareness into the students and help them cultivate new ideas so that new problems from their work can be solved with fresh methods and technology.

It can be concluded that how to incorporate innovation-oriented education into software engineering course deserve our research.Related technology

II. A STRONG SENSE OF INNOVATION REPRESENTS ONE OF THE PROMINENT CHARACTERISTICS OF SOFTWARE ENGINEERING COURSE

Being a field full of vitality, software engineering needs constant new knowledge to make perfect with its fast renewing speed [2, 3]. The strong sense of innovation represents the prominent characteristics of the software engineering course in contrast with other major courses of computer.

Viewing software engineering from the perspective of its development history, we can clearly find that innovation run through the whole process of software engineering and propel the development of it. For example, the concept of software engineering itself was innovative because engineering was exclusively referred to hardware building such as civil engineering, water engineering and so on before the concept of software engineering, as a work of program writing, was put forward. A second example is the new concept of software (referring to the complete set of programs, data and related document). In people's eyes software is equal to program, excluding data and related document, people do not consider other work as that of developing software, in the early software development activities. A third example is the concept of software life circle model represented by waterfall model. People developed software according to their own needs before with no standards to follow. Waterfall model changed the situation. A forth example is the concept of structured programming. Before the concept was put forward, programming was limited to skills and at peoples' will, so the programs was unclear and difficult to read and understand. Now people can write programs in accordance with three basic structures as the concept was put forward, thus improving the quality of programs. A fifth example is the object-oriented programming. The traditional method was to divide data and operations on data into two separate parts, however, object-oriented programming integrated them together, then the concept of class and object came into being so that the natural mapping from problem space to solution space was achieved and the independence of module was reinforced. It is of great significance to the maintenance and development of large software. A sixth example is the UML [4], which unified diverse modeling language. A seventh example is agile method [5, 6]. To the problems of the traditional developing process which proved to be complex, resist changes and ignore human’s initiatives, it bring up a solution. A eighth example is Agent-oriented method [7, 8],

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

which represented peoples' good will that computers can serve people automatically. The last but not the least example is the concept of CMM [9], which provided the standards of assessing software mechanism and the effective strategies of improving it.

III. Software engineering course is the best carrier of fostering students' sense of innovation

All it is better to conduct innovation-oriented education and instill the sense of innovation into students via cases, which claims to be the best carrier to spread new ideas and thoughts. For students of computer majors, they are offered abundant cases displaying the sense of innovation as shown in the second part of this paper because the course software engineering reflects a strong awareness of innovation in contrast with other courses of computer major. This implies that software engineering course is the best carrier of implementing innovation-oriented education. Therefore, by taking full advantage of it, teachers should help cultivate students' sense of innovation, incorporate innovation-oriented education into the course and instill innovative ideas and thoughts while they are teaching the course.

IV. New thoughts for teaching software engineering course

The traditional software engineering course taught students nothing about innovation. It is, therefore, a new issue before us as to how instill sense of innovation into students via this course.

If the course were still taught in the traditional way with the sole aim to instill knowledge, fostering students' awareness of innovation would turn out to be a failure. Therefore, it demands new thoughts for teaching this course since we try to instill into students major knowledge as well as the awareness of innovation.

New thought I: Approach of Dispersion

In contrast with the traditional in-classroom teaching where knowledge points of each chapter is taught to the students in the order arranged by the textbook, the approach of dispersion is carried out as follows: when the knowledge points of each chapter is taught, any innovative thought or idea concerning these knowledge points is analyzed from the perspective of innovation so that the students can realize that the introduced method, technology or concept of these knowledge points is innovative which has resolved the problems in the way of software engineering development. Then the students may comprehend the process of innovation and come back with new ideas. For example, while teaching the first chapter "profile of software engineering", which involves some knowledge points of innovation in each section, teachers should reveal the implied new ideas according to the content and feature of the knowledge point. By adopting the approach of dispersion, each step they take is considered as the analysis of innovation. Specifically, the section software process has such knowledge points as "waterfall model", "rapid prototyping model", "incremental model", "spiral model", "fountain model" and "agile process" [10], then teachers should analyze the process of innovation while teaching each point.

New thought II: Approach of concentration

The approach of concentration should be implemented as follows: All of the knowledge points in each chapter is firstly introduced in accordance with the traditional teaching method, then a new section is added to each chapter where all of the knowledge points concerning innovation is put together and analyzed. To put it more simply, analyzing knowledge points in terms of innovation in each section of a chapter is uncalled-for, but conducted in the added the section to the chapter so as to foster students' sense of innovation. For example, when teaching the first chapter "profile of software engineering" teacher should focus their explanation purely on such knowledge points as "waterfall model", "rapid prototyping model", "incremental model", "spiral model", "fountain model" and "agile process", rather on innovation, then he should add a new section to concentrate their effort on the analysis of the knowledge points in terms of innovation.

To put thought I and thought II into comparison, we could find that the former is more natural in teaching software engineering course since innovative concept is instilled into students simultaneously. The author of this paper just employs the former approach. However, students are likely to ignore its importance as analysis of innovation is added after the knowledge of software engineering is taught. In thought II, teachers put the knowledge concerning innovation together and concentrate their explanation on it, the advantage of which is to intensify innovation and impress students so as to enhance their awareness of innovation.

As traditional textbook of this course excludes anything about innovation, teachers are required to rearrange their in-classroom teaching and prepare a new teaching plan in terms of innovation. It does deserve further research when it comes to compiling a new textbook characterized by innovation so as to incorporate innovation-oriented education into software engineering course and achieve the new thoughts for teaching.

V. Conclusion

Software engineering depended on innovation for its development in the past and will still depend on it in the future. Closely related to innovation, software engineering course should help foster students' sense of innovation and cannot be replaced by other major courses. It is a long-term subject deserving research to incorporate innovation-oriented education into software engineering course, the goal of which is to help students establish new ideas and cultivate their ability of innovation. There is still a long way to go before we finally achieved it.

Acknowledgment

This research has been supported by Science Plan Project of Zhejiang Province, 2008C31009, and by science research foundation of Department of Education of Zhejiang Province for special purpose, 20050136

REFERENCES

[1] Roger S. Pressman , Software Engineering: A Practitioner's Approach , McGraw-Hill, 2001.

[2] Yang FQ , "Think on the development of software engineering technology", *Journal of Software*, Beijing , 2005,16(1):pp. 1-7.

[3] [3]Yin Feng , "Development and revelation of software engineering", *China Information Review* , Beijing , 2005,(9): pp.59-61.

[4] James Rumbaugh,Ivar Jacobson,Grady Booch, *The Unified Modeling Language User Guide,* Addison-wesley, 2001.

[5] J. Highsmith, A. Cockburn, "Agile software development: the business of innovation", *Computer* , 2001,34(9):pp. 120-121.

[6] Zhang Jingzhou,Qian leqiu, Zhu Sanyuan, "Overview of Agile method", *Journal of Computer Applications and Software* ,Shanghai, 2002,(6) :pp. 1-9+54.

[7] Jennings N R, "On agent-based software engineering", *Artificial Intelligence*, 2000,117(2):pp. 277-296.

[8] Mao Xingjun, Chang Zhiming, "Software Engineering of Agent-Oriented: Current and challenge", *Journal of Computer Research and Development*, Beijing ,2006 , 43(10) : pp. 1782-1789.

[9] He Xinggui, *Capability Maturity Model of Software*, Tsinghua University Press, Beijing ,2000.

[10] Zhang Haifan , *Introduction to Software Engineering* (fifth edition) , Tsinghua University Press, Beijing , 2008, pp. 14-31.