Journal of Contemporary Educational Research, 2024, Volume 8, Issue 11
http://ojs.bbwpublisher.com/index.php/JCER
ISSN Online: 2208-8474
ISSN Print: 2208-8466

Exploration of the Integration of Software Engineering Thinking into Embedded System Design Course

Jinyan Hu*, Haihua Yu, Yumei Gong, Shaojing Song

School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 200030, China

Corresponding author: Jinyan Hu, jyhu@sspu.edu.cn

Abstract: Engineering practice is the key bridge between college education and actual work in the industry. In order to deliver qualified talents with engineering quality to the industry, this paper explores integrating software engineering thinking into the Embedded System Design course. A practical and effective teaching mode is designed consisting of immersive learning, case-based learning, progressive practice, interactive learning, and autonomous learning. Through this teaching mode, multi-levels of closed-loop have been established including final project cycle closed-loop, testing cycle closed-loop, and product cycle closed-loop. During this process, students gradually transition to putting forward product requirements, carrying out design and development, thinking and solving problems, collaborating, and assuring quality from the perspective of software engineering. The practice results show that students' engineering quality has been significantly improved.

Keywords: Embedded System Design; Software engineering; Engineering practice; Teaching mode

Online publication: November 27, 2024

1. Introduction

With the development of science and technology, colleges and universities are facing increasingly severe challenges in cultivating engineering and technical talents. As an important part of industries such as aerospace, automotive, medical, communication, industrial automation, and other industries, embedded systems have seen a growing demand for their development and applications. However, the current teaching and practice of most Embedded System Design courses are disconnected from practical engineering applications and cannot meet the requirements of the industry [1]. Software engineering thinking not only provides students with a complete set of development processes and methodologies, but more importantly, it cultivates students' systematic thinking, quality control awareness, team cooperation ability, continuous improvement spirit, and proficiency in the application of tools and technologies. These abilities are critical to the success of students in their future engineering practice [2]. This paper aims to explore the introduction of software engineering into the teaching of the Embedded System Design course through the design and implementation of appropriate teaching modes and strategies, cultivating students' engineering quality and improving their engineering practice ability.

2. Instructional design

Embedded System Design is a practical course in electronic information engineering. Before this course, students have learned the fundamentals of embedded systems and performed some basic experiments. In this course, STM32F10x is taken as the microcontroller of the embedded system. The contents of the course are organized through the design and development of a complete embedded system, which runs through all stages of the course. Firstly, we propose a design requirement for an embedded system and introduce the development process of embedded products, corresponding job positions, and the basic knowledge and application capabilities of embedded systems that should be possessed in actual work. Next, we will delve into the fundamental practices of embedded systems, and then carry out more comprehensive applications including single-module development and multi-module integration tasks [3]. Finally, according to the design requirement proposed at the beginning of the course, the final project design is carried out using embedded system design methods.

Fundamental practices involve clock system, interrupt and events, and peripherals such as input/output interfaces, general purpose timers, USART, ADC, etc. By using simple hardware such as push buttons and LEDs, various applications can be implemented. Through these practices, students become more familiar with the embedded system, from hardware architecture and interfaces to software programs and control logic.

The comprehensive practice stage focuses on two aspects of applications, one is user interface, and the other is serial bus communication. For consumer electronics products, the user interface, especially the graphical user interface, is crucial for the product experience. Serial bus is the communication method adopted by most sensors and execution devices. The most commonly used serial buses including the IIC bus, SPI bus, and CAN bus are involved in this section. In the process of comprehensive practice, basic tasks and extended tasks are set to meet the needs of students with different backgrounds. For example, in the User Interface module, the basic tasks are user-defined font and basic graphic display, and the extended tasks include but are not limited to color image display, animation display, transparent effect display, etc.

The final project is to realize a complete embedded system application that is treated as a real product, such as an automobile dashboard system [4]. The implementation process includes product definition, requirement analysis, system design, software development, integration test, etc. Finally, students should submit a design report and conduct product demonstration and oral defense. Through this project, students gain a comprehensive engineering experience of a complete embedded system, and their hands-on ability is further improved.

3. Teaching modes and strategies

3.1. Immersive learning

In the teaching team of this course, more than 80% of the teachers are dual professional teachers with rich industrial practical experience. At the initial stage of the course, the corresponding job positions and responsibilities of embedded product development are introduced from the perspective of actual work, as shown in **Figure 1**. Students immerse themselves in the embedded product development process in various positions (project manager, hardware engineer, software engineer, testing engineer, field application engineer) during comprehensive applications and the final project design of embedded systems. Through immersive learning, students not only improve their design and development abilities but also develop their systematic thinking and team cooperation ability.

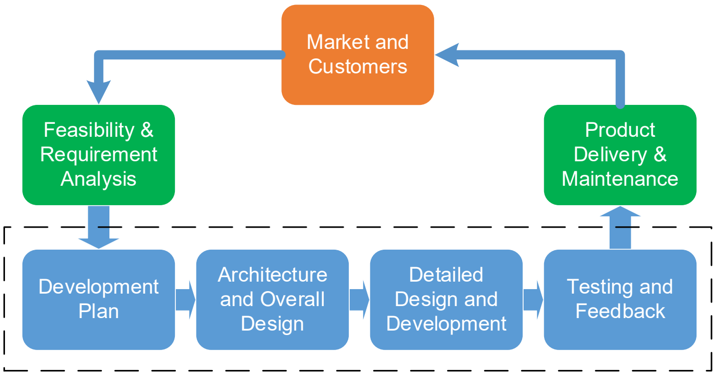

Figure 1. Embedded product development process

3.2. Case-based teaching

By selecting appropriate application cases as the teaching content, students can understand the application fields of embedded systems intuitively [5,6]. In the comprehensive applications stage, application development cases that are close to students' lives are designed, such as stick figure animation, electronic hygrometer, Peppa's USB disk, etc. **Figure 2** shows two examples of case-based teaching. The projects should be sufficiently challenging and can stimulate students' interest in learning. Case-based teaching not only helps students master explicit knowledge, methods, and skills but also guides them to think in specific situations, thus deepening their understanding and mastery of embedded systems. At the same time, students' interest, client awareness, and innovation awareness can be greatly enhanced.

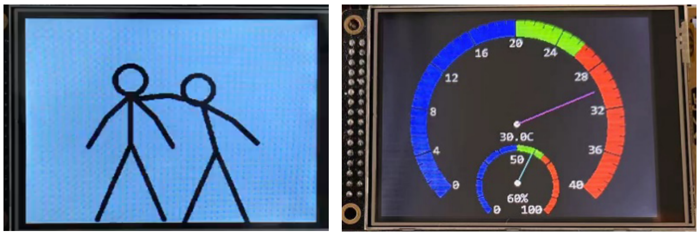

Figure 2. Examples in the case-based teaching. Left: Stick figure animation; Right: Electronic hygrometer

3.3. Progressive practice

Progressive practice includes upward progressive practice and downward progressive practice. Basic and extended tasks have been set for comprehensive applications, and students can choose the task level according to their own situation. For example, the basic task of a single bus module is real-time temperature and humidity detection + LCD display, and the extended task is a color screen hygrometer with GUI. This is an upward progressive practice. Through this level of practice, students can establish a preliminary sense of optimization on the basis of completing basic tasks, that is, the spirit of continuous improvement. For comprehensive applications and the final project, students can choose to create new APIs, and improve and perfect related driver programs based on their own situation within the existing program architecture. For example, in the LCD display driver program, students create an API that can draw user-defined fonts of any size according to the programming specifications. This is a downward progressive practice. Through this level of practice, students will have a better understanding of embedded software architecture and establish a sense of standardization and craftsmanship.

3.4. Interactive learning

In interactive learning, two students study together in groups, exchanging R&D developer and SQA tester identities during the comprehensive applications and the final project stage. The R&D developer should ensure quality during development, while the SQA tester should try to identify issues during testing. They discuss the reasons and solutions for the issues together. **Figure 3** shows the working flow of R&D and SQA. In addition, students are encouraged to communicate more with teachers in or out of class. The teaching staff arranges office time and provides prompt answers to students' questions face-to-face, or through online methods using the course website. Through interactive learning, students can improve their ability to identify and solve problems and establish a sense of quality control and teamwork.

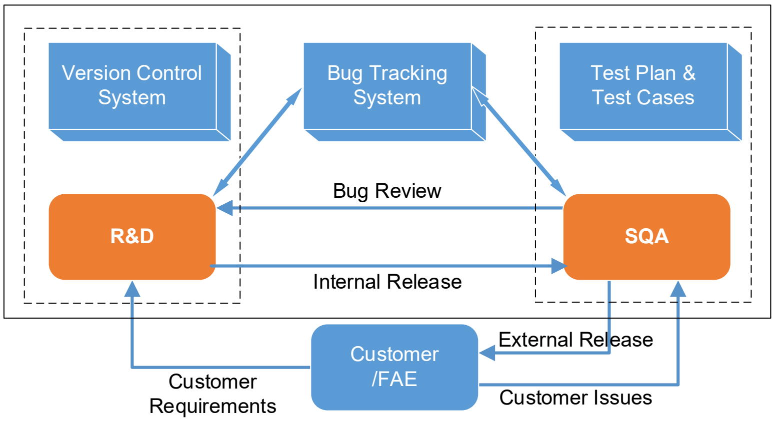

Figure 3. R&D-SQA working flow

3.5. Autonomous learning

In order to achieve a significant improvement in embedded system design ability through this course, it is insufficient to rely on limited class hours, and students need to practice in their extracurricular time. Autonomous learning requires preparation and support in many aspects. In terms of learning resources, a large number of software and documentation resources have been uploaded to the course website for students to refer to and download. In terms of hardware systems, a pocket development board is provided to each student for practice after class. In terms of technical support, teachers set clear goals for tasks, regularly check students' learning progress, and provide personalized guidance and suggestions. Through autonomous learning, the learning time of the course has been extended, allowing students to realize their ideas at any time, which helps to promote the improvement of proficiency in the application of hardware and software tools and technologies.

4. Innovation and effects

In the oral defense of the final project, students' project designs and reports are evaluated. Students can improve their design and paperwork according to the feedback within the specified time before the deadline. Thus, the final project closed-loop is formed. In interactive learning, the R&D developer and SQA tester cooperate to find bugs, fix bugs, and perform version upgrades. Thus, the testing cycle closed-loop is formed. Introducing the entire process of product development from requirement to delivery into conventional teaching allows students to immerse themselves in various roles in the product development process. Thus, the product cycle closed-loop is formed.

In the past three years, after taking this course, students majoring in Electronics and Information Engineering at Shanghai Polytechnic University (SSPU) have won a total of six university-level innovation projects, eight municipal (Shanghai) innovation projects, and three national (China) innovation projects. The proportion of high-level innovation projects is relatively high. In terms of participating in competitions, six groups of students participated in the TI Cup National Undergraduate Electronic Design Competition, among which one group won the second prize and two groups won the third prize.

5. Conclusion

This paper explored integrating software engineering thinking into the Embedded System Design course oriented to engineering quality cultivation and has achieved remarkable results. Through immersive teaching, students can master the life cycle of product development, clarify their future job positioning, and improve their systematic thinking and team cooperation ability. The use of case-based teaching has significantly improved students' learning interest and innovation awareness. In addition, through progressive practice, students can not only achieve upward application progress but also be driven downward to establish the standardization sense and craftsman spirit. Through interactive learning, students' problem-solving skills and quality control awareness can be improved. Through autonomous learning, the improvement of students' proficiency in the application of tools and technologies has been further promoted. The students' performance in the college student innovation projects and high-level competitions has proved the effectiveness of integrating software engineering thinking into embedded system design teaching.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Gao J, Huang L, 2021, On the Practical Teaching System of Embedded System Courses for Integration Between Industry and Education. Research in Higher Education of Engineering, 69(3): 39–43.

[2] Cico O, Jaccheri L, Nguyen-Duc A, et al., 2021, Exploring the Intersection Between Software Industry and Software Engineering Education—A Systematic Mapping of Software Engineering Trends. Journal of Systems and Software, 172(2): 1–28.

[3] Liu C, Deng Q, Chen G, et al., 2023, Experimental Teaching Reform and Practice of Embedded Systems in Internet of Things Specialty Under the background of New Engineering. Research and Exploration in Laboratory, 42(10): 196–199.

[4] Chen J, Han Q, Yang A, et al., 2022, Exploration and Practice of Project Team Teaching Method in Embedded Microcontroller Class. Research and Exploration in Laboratory, 41(4): 232–235.

[5] Pasricha S, 2022, Embedded Systems Education: Experiences with Application-Driven Pedagogy. IEEE Embedded Systems Letters, 14(4): 167–170.

[6] Rashid M, 2020, An Undergraduate Course on Model-Based System Engineering for Embedded Systems. Computer Applications in Engineering Education, 28(3): 645–657.