Research on Comprehensive Software Engineering Practical Training Course Based on Constructivism Learning Theory

Taotao Li, Huawei Huang, Mingzhi Mao*, Shude Lan, Zhuliang Lin, Yuzheng Li, Zhenbang Huang, Yuting Zhou, Zibin Zheng

School of Software Engineering, Sun Yat-sen University, No. 2 University Road, Zhuhai, China

{litt93, huanghw28, mcsmmz, lanshd, linzhliang6, liyzh255, huangzhb27, zhouyt69, zhzibin}@mail.sysu.edu.cn

*Corresponding author

Keywords: Constructivism, Software Engineering Practical Training, Student-Centered

Abstract: Based on the current issues in the student learning and course content of the software engineering practical training course at Sun Yat-sen University, we carry out a systematic course design by using constructivism learning theory and integrating the concept of industry-education integration. Further, we propose a comprehensive software engineering practical training course system, which also be called the *one center, two integrations, three guidance, four stages, and five objectives*. The proposed course content is aligned with real-world engineering and market demands. We share the implementation plan that focuses on teaching practices driven by industry-education integration. The implementation results demonstrate that the course reform enhanced students' practical engineering skills and their ability to meet market demands.

1. Introduction

In 2019, the Ministry of Education issues the "Opinions on Deepening the Reform of Undergraduate Education and Teaching to Comprehensively Improve the Quality of Talent Training", which stresses the importance of cultivating students' innovative spirit, entrepreneurial awareness, and innovation-entrepreneurship capabilities, and integrating entrepreneurship education into professional education [1]. Sequentially, Sun Yat-sen University issues the "Opinions on Continuously Deepening the Reform of Undergraduate Education and Teaching to Enhance the Quality of Talent Cultivation" [2], which explicitly emphasizes the "collaborative education model between industry, education, and research," aiming to build an innovative platform integrating scientific research and industry-education integration. This platform focuses on cultivating top innovative talents who can serve national strategic needs and lead future developments by combining education with practical applications at the forefront of both research and industry.

The Comprehensive Software Engineering Practical Training course is a core professional course designed for third-year software engineering students. The course requires these students to have foundational knowledge in C++ programming and data structures and algorithms [3]. This course is primarily offered to third-year software engineering students at top-tier universities in China [3]. The general learning profile of the students reveals *three strengths and three weaknesses*: i) strong programming fundamentals but insufficient awareness of software engineering practices, ii) strong self-awareness but weak collaboration skills, and iii) strong coding skills but underdeveloped programming thought processes. Traditional software engineering courses typically rely on lecture-based teaching, often failing to adequately develop practical skills and being disconnected from industry demands [4].

To address these issues, the course team, guided by constructivism learning theory, designs and implements a software development course centered around real-world and industry-related applications. Here the course can advance the integration of industry and education. The course is student-centered, and focuses on hands-on software engineering projects. In addition, the course also integrates on-campus applications with innovation and entrepreneurship competitions,

  DOI: 10.25236/icemeet.2024.012

emphasizing collaborative learning as the core teaching method. Through the stages of *training, understanding, reflection, and presentation*, the course not only facilitates deep knowledge internalization but also enhances students' engineering practice capabilities by integrating real-world industry applications. This reform effectively bridges the gap between teaching and industry demands.

2. Design of the Comprehensive Software Engineering Practical Training Course

2.1. Design Framework of the Comprehensive Software Engineering Practical Training Course with Industry-Education Integration

In the current software engineering curriculum, the depth of industry-education integration still needs improvement, as the connection between academia and industry demands remains insufficient [3].

Constructivism learning theory emphasizes student-centered learning, where students actively construct knowledge, and teachers serve as facilitators and guides[5-6]. The teaching methods based on this theory often involve creating real-world contexts and collaborative learning, allowing students to independently build and understand knowledge in specific environments. The software engineering practical training course plays a key role, with the lead instructor guiding the systematic redesign of course objectives, content, organization, and evaluation, embedding industry-education integration throughout the entire course. Following the constructivism learning theory, the course team develops a teaching design framework based on *One Center, Two Integrations, Three Guidance, Four Stages, and Five Competency Objectives*. This framework focuses on enhancing students' engineering practice skills and aligns their learning with industry needs. The framework can also facilitate a deep connection between knowledge and real-world application, as shown in Figure 1.

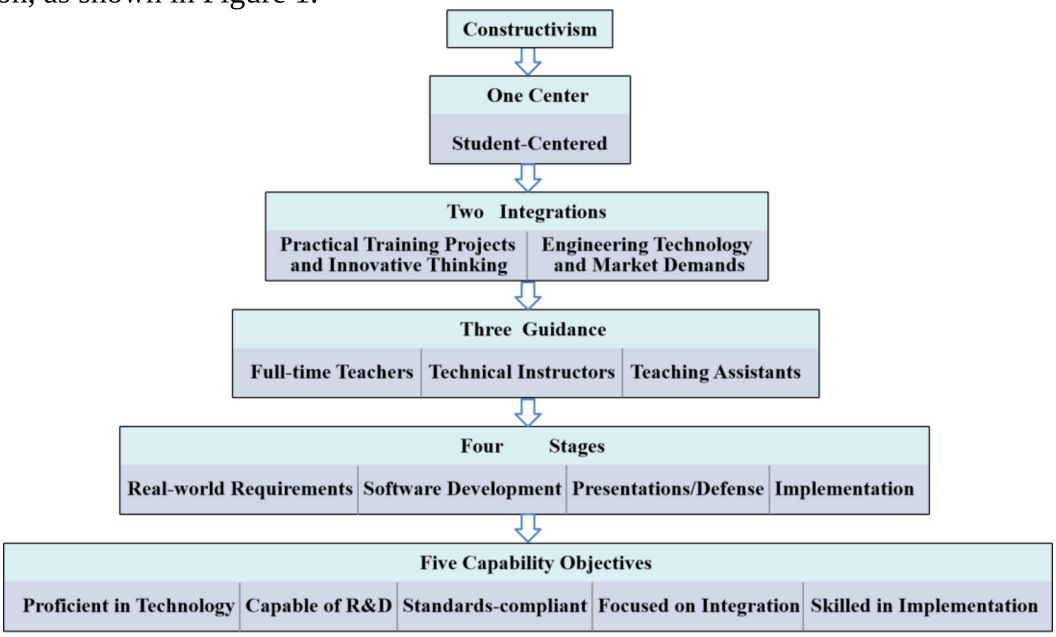

Figure 1 Design Approach for the Software Engineering Practical Training Course Framework.

2.2. System Design of the Comprehensive Software Engineering Practical Training Course

The comprehensive software engineering practical training course aims to cultivate outstanding software engineering talent. To achieve this, the course team designs the system around two key integrations. As shown in Figure 2, the course first integrates software engineering training projects with innovative and entrepreneurial thinking. Under this integration, students are guided to engage in software development and project implementation based on real-life and industry application scenarios. Besides, this integration also enables students to apply software development theories and technical solutions effectively in collaborative projects, thus improving their problem-solving skills.

Additionally, the course integrates elements of innovation and entrepreneurship into professional training projects. Under this integration, students are encouraged to focus on national development strategies and regional economic needs, developing both engineering thinking and business model design skills. By combining engineering technology with market demands, students are equipped with the practical skills needed for employment or entrepreneurship, laying a solid foundation for their future careers.

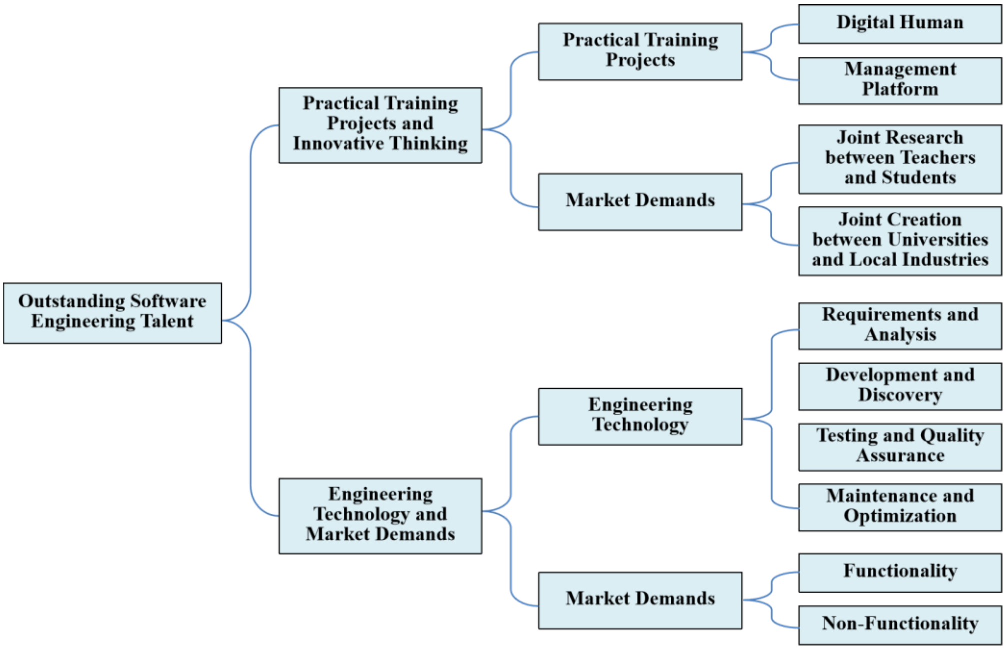

Figure 2 System Design Diagram of the Comprehensive Software Engineering Practical Training Course.

3. Implementation of Comprehensive Software Engineering Practical Training

The primary goal of the comprehensive software engineering practical training course is to master programming, excel in tools, understand principles, and focus on hands-on practice [7-8]. Combining constructivism learning theory with industry-education integration elements, the course is designed to align with industry demands, and enhances programming skills, tool proficiency, and the refinement of engineering thinking.

3.1. Overall Course Implementation Plan

The implementation of the comprehensive software engineering practical training course is divided into two steps: Engineering Practical Training Project Review Committee, and Project Implementation and Outcome Conversion. In the first step, prior to the course start, a practical training project review committee is established, following the $1+N$ model, which consists of the course instructor and several external mentors. The committee's primary goal is to ensure that the projects generated from the practical training course are aligned with real-world application demands and meet the standards of innovation and entrepreneurship competitions.

In the other hand, the course team provides resource support to outstanding projects, promoting deeper development and real-world implementation. Additionally, students are encouraged to participate in various innovation and entrepreneurship competitions, with assigned mentors to guide them through the process.

3.2. Detailed Course Implementation

The comprehensive software engineering practical training course is implemented in three stages: *Preliminary Preparation, Midterm Development, Final Evaluation*. In the preliminary preparation stage, students are required to form teams before the course starts, and each group selects a development task. Teaching assistants (abbr. TAs) are also recruited for each group. The evaluation criteria, defense methods, and presentation formats are then announced. During the midterm development stage, students are required to submit a requirements analysis and a prototype design document. Teaching assistants must regularly organize meetings with team members. In the final evaluation stage, each group must select a representative to participate in the defense, and groups may also voluntarily participate in the presentation defense. The presentation serves as a bonus opportunity for students. A detailed implementation process is outlined below.

3.2.1. Preliminary Preparation

- **Student Grouping.** To facilitate team formation for the entire class, Tencent Docs is used for management. By importing the class list, students can fill in their team status, and the document automatically tracks which students have formed teams and who have not yet participated. This ensures the accuracy and completeness of the team information, significantly improving the efficiency of team organization.
- **Teaching Assistant Recruitment.** The recruitment of teaching assistants is managed through Tencent Docs, specifically for the Comprehensive Software Engineering Practical Training course. The document gathers various project topics for students to view and select, ensuring a transparent and open selection process, aiding in recruiting suitable teaching assistants for the course.
- **Grading Evaluation.** The grading process is managed via a Tencent Docs, which collects and records each student's performance. The document provides a convenient evaluation system, ensuring transparency and efficiency in the grading process, and allows students to access their feedback at any time, facilitating further learning improvements.
- **Roadshow Format.** For the roadshow, each team is required to submit a PPT, formatted to display two teams' results per screen. The PPT supports animated content to enhance the visual presentation of the projects, ensuring that each team's presentation is clear, concise, and impactful.

3.2.2. Midterm Development

- **Requirement Analysis, Prototype Design, and Development Phase Meetings.** During the development process, a major meeting is held at each phase—requirement analysis, prototype design, and development. Participants include the business mentor, project TAs, team members, and representatives from the previous cohort. These meetings ensure smooth project progress and integration with industry needs, helping students understand real-world application scenarios. Additional meetings are not held unless necessary, to maintain efficiency.
- **Teaching assistants Communication and Feedback.** Teaching assistants (TAs) are required to i) maintain close communication with the group leader and members, ii) provide regular feedback on issues and needs, and iii) coordinate with the business mentor to ensure the project stays on track. Each semester, TAs must organize at least 8 business meetings with their teams, including 3 major meetings. Through industry-education integration, TAs help bridge theory and practice, supporting project implementation. TAs must provide meeting records and photos at the end of the semester for verification.
- **Group Leader Communication with TAs and Business Mentor.** The group leader can request meetings with the TA or business mentor to discuss project progress or resolve issues. These meetings count towards the business meeting quota and ensure that, within the framework of industry-education integration. These meetings' goal is to ensure that students can address real-world problems promptly.

3.2.3. Final Evaluation

- **Material Collection.** Before the project defense, each group must submit the defense PPT, roadshow poster, peer evaluation scores, and TA grading scores. After the defense, teams must provide the project report and code, which will be uploaded to the college's Gitee platform. By utilizing industry-education integrated project management tools, the project’s completeness and presentation quality are ensured, promoting subsequent project incubation and deployment.

- **Defense Notification.** The time and location for the defense and roadshow are determined in advance. Each group must designate a representative to present the project for 8 minutes, followed by a 2-minute Q&A session. The presentation should cover project functionality, unique features, and team member roles, ensuring the project is aligned with industry demands and practical scenarios.
- **Defense Requirements.** Each group must arrive 20 minutes early for sign-in, and PPTs will be preloaded onto the presentation computer. The presentation duration is 10 minutes, and students are expected to dress formally. The project presentation should demonstrate practical application scenarios and the team's engineering thinking. Teams must exhibit proper division of labor, tightly connected with industry practice, showcasing the results of industry-education integration.
- **Roadshow Arrangement.** The roadshow is conducted concurrently, showcasing the unique features of each project. Participating teams must prepare a one-page presentation PPT, and attendees will cast votes to rate the projects. Based on the vote count, teams may earn bonus points, encouraging the promotion and practical application of projects while enhancing students' entrepreneurial awareness through project exhibitions.

4. Conclusion

According to the current state of software engineering education at Sun Yat-sen University, where there is a need to enhance the connection between theoretical learning and industry practice, this comprehensive software engineering practical training course integrates constructivism learning theory and industry-education integration. The integrated course is centered around solving real-world problems and industry-relevant scenarios, offering students the opportunity to develop skills through hands-on projects. The practical training course system has been designed with a structured framework that includes setting clear course objectives, organizing content, and evaluation processes[9-10]. A systematic redesign was applied, focusing on bridging the gap between students' knowledge and practical industry needs. This redesign encourages collaborative learning, allowing students to take a leading role in the learning process while receiving guidance from faculty and industry experts.

The implementation process follows a phased approach. Students are grouped into teams, tasked with developing solutions to real-world problems, and are supported by teaching assistants and external mentors. The evaluation includes a project defense and roadshow, where students present their work, allowing for both academic and industry feedback. This integration of practical application and collaboration strengthens students’ engineering and problem-solving capabilities, which ensures their readiness for future career opportunities. By combining software engineering training with entrepreneurship and innovation, the course nurtures students' technical and business skills, equipping them to meet both national strategic needs and market demands.

Acknowledgements

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012360; and in part by the National Natural Science Foundation of China under Grant 62272496; and in part by the 2023 Project of Computer Education Research Association of Chinese Universities (CERACU2023R02).

References

[1] Ministry of Education of the People's Republic of China.(2019)Opinions on Deepening the Reform of Undergraduate Education and Teaching to Comprehensively Improve the Quality of Talent Cultivation.http://www.moe.gov.cn/srcsite/A08/s7056/201910/t20191011_402759.html.

[2] Liu,L.(2024)Sun Yat-sen University Promotes the Independent Cultivation of Top Innovative Talents. Ministry of Education Bulletin. https://hudong.moe.gov.cn/jyb_sjzl/s3165/202409/t202409t20240903_1148679.html.

[3] Mao,M.Z.,Liu,Z.(2022)Exploration of Ideological and Political Education in IT Project Management Course Under the Background of New Engineering.Computer Education,4,21-23.

[4] Wang,L.Q.,Chen,J.G.,Mao,M.Z.,Zheng,Z.B.,Wang,Y.L.(2024)Exploration of Ideological and Political Education in Software Engineering Practical Training Course Based on Constructivism Theory. Computer Education,4,101-104.

[5] Wang,L.Q.(2020)Research on Curriculum Reform of Programming Design Under the Gold Course Standard.Software Guide,19(12),213-216.

[6] He,K.G.(1997)Teaching Models,Methods,and Design Based on Constructivism Theory.Journal of Beijing Normal University(Social Sciences Edition),143(5),74-81.

[7] Mao,M.Z.(2000)Reflections and Experiences on Offering an Elective Course in Computer Numerical Algorithms.Modern Computer,11,96-98.

[8] Mao,M.Z.,Liu,Z.(2023)Research on Software Project Management Teaching Driven by Ideological Education and Product Development.Software Guide,22(6),235-239.

[9] Wang,L.Q.(2017)Research on Curriculum Reform of Object-Oriented Programming for Flipped Classrooms.Software Guide,16(9),222-225.

[10] Wang,L.Q.,Zhou,W.Q.(2015)Design and Production of Micro-Courses in Fundamental Programming for Universities.Software Guide,14(10),177-179.