

Research on Content Construction and Competency Progression System of "Competition-Education Integration" based on Software Maker Training

Huawei Mei, Mingquan Zhang, Junxian Zhao

North China Electric Power University, Baoding, China

Abstract: Against the backdrop of the deep integration of "Emerging Engineering Education" construction and innovation and entrepreneurship education, the cultivation of software professionals faces challenges such as lagging traditional teaching paradigms and strict disciplinary barriers, making it difficult to meet the industry's demand for "T-shaped talents". Focusing on software-related disciplinary competitions and centering on "Competition-Education Integration," this paper constructs a "Competency Map"-driven cultivation framework, designs three collaborative models-Project-Driven, Competition-Oriented, and School-Enterprise Dual-and establishes a three-tier competency progression system comprising the Basic, Advanced, and Excellence levels. Verified through over 20 years of teaching practice, this content construction scheme effectively consolidates students' technical foundation, fosters innovative thinking and engineering capabilities, addresses issues such as the fragmentation of practical teaching and insufficient interdisciplinary integration in software majors, and provides a replicable practical paradigm for optimizing maker training content in software programs at universities.

Keywords: Competition-Education Integration, Software Maker Training, Competency Progression System, Project-Driven.

1. Introduction

1.1. Policy Background and Era Demands

The report of the 20th National Congress of the Communist Party of China highlighted invigorating China through science and education, developing a quality workforce, and driving innovation-led development as fundamental, strategic supports for national development, stating clearly that "education is the foundation for talent emergence, and science and technology are the bedrock of national strength." Within the wave of "Emerging Engineering Education" construction, innovation and entrepreneurship education in universities needs to balance the consolidation of students' engineering fundamentals with the cultivation of their innovation capabilities. As a core pillar of the digital economy, talent cultivation in computer software majors must break away from the "knowledge indoctrination" model, position professional goals towards fostering self-directed learning abilities, and build a "practice-driven, innovation-led" education system^[1]. Simultaneously, it must align national needs with personal values, cultivating students' abilities in self-management, service, and development, and establishing correct ideals and values^[2].

1.2. Current Status and Existing Problems in Practical Teaching for Software Majors in Universities

While current practical teaching in software majors has seen progress, deep-seated contradictions remain prominent:

Lagging Teaching Paradigms: Traditional teacher-centered classrooms lead to passive knowledge acquisition, lacking training in the complete innovation chain of "design-practice-iteration". The "China University Innovation Education Development Report (2023)" [3] indicates that only 23% of students believe classroom teaching effectively enhances innovative thinking. Students in software majors are

often confined to code writing exercises, struggling to develop systemic engineering thinking.

Rigid Disciplinary Barriers: Courses in software majors often focus on single technical domains like programming languages and algorithm design. Students lack interdisciplinary knowledge in hardware design, user experience, business logic, etc., making them unable to handle "software + hardware + service" composite project demands^[3], thus becoming disconnected from the industry's requirement for transversal literacy in "T-shaped talents".

Insufficient Resource Integration: University laboratories often have low accessibility and slow equipment updates, industry-university cooperation tends to be superficial, and competitions are disconnected from the curriculum system. Student practice often consists of fragmented course designs or short-term competition preparation, making it difficult to form a progressive enhancement from basic skills to innovative capabilities^[4, 5], significantly reducing the effectiveness of practical teaching.

2. Content Construction Framework for "Competition-Education Integration" in Software Maker Training

2.1. Core Concept: Anchoring Cultivation Objectives with a "Competency Map"

Breaking away from the "course-oriented" mindset, we establish a "Competency Map"-driven cultivation framework, using software-related competitions (e.g., "Internet+" Competition, ACM International Collegiate Programming Contest, National College Student Software Design Competition) as "anchor points" for competency development. Aligning with the industry's demand for "T-shaped talents", it vertically focuses on professional technical capabilities (algorithm design, system architecture, cross-platform development, etc.) and horizontally covers interdisciplinary

literacy (product thinking, teamwork, engineering standards, etc.), ensuring high alignment between maker training content and industry needs.

2.2. Multi-Collaboration Models: Three Driving Models Supporting Content Implementation

2.2.1. Project-Driven Model: Connecting Real Needs, Designing Progressive Projects

Transform genuine enterprise demands into a three-tier progressive project system: "Course Project – University-level Competition Project – Provincial/National-level Competition Project", achieving deep integration between practical content and industrial reality. For example, collaborating with an internet company to develop a "Smart Campus Management System": during the course project phase, students complete user requirement analysis and core module coding; the university-level competition phase requires system integration and functional optimization; the national-level competition phase involves adding advanced features like data visualization and high-concurrency processing, along with completing business value analysis. This entire process covers the full workflow: requirement analysis, architecture design, code implementation, testing deployment, and roadshow defense, enabling students to master complete engineering methods through project advancement.

2.2.2. Competition-Oriented Model: Centering on Competition Needs, Building Closed-Loop Training Content

Establish a closed-loop mechanism of "Pre-competition Training – Simulated Competition – Actual Participation", with training content designed specifically for competitions. Pre-competition offerings include "Algorithm Bootcamp", "UI/UX Design Workshop", and "Business Plan Writing Course". The "Algorithm Bootcamp" focuses on high-frequency test points for the ACM-ICPC, featuring daily programming exercises and error review sessions. The "UI/UX Design Workshop" invites industry designers to

explain user research methods and interface design standards. During competitions, cross-grade student teams are formed, with senior students guiding juniors in completing project module development, fostering knowledge transfer. Post-competition debriefings are held to feed identified technical shortcomings (e.g., inadequate distributed system debugging skills) and process issues (e.g., chaotic project schedule management) back into curriculum optimization, updating case libraries for courses like "Software Innovation Practice Technology".

2.2.3. School-Enterprise Dual Model: Introducing Industry Resources, Enriching Content Dimensions

Invite enterprise mentors to participate in cultivation plan design and co-build a "Innovation and Entrepreneurship Mentor Database", integrating cutting-edge industrial technologies and engineering standards into maker training content. Enterprise engineers deliver multiple "Industry Frontier Lectures" per semester, covering hot topics like the application of AIGC in software development and compliant data security design. They also guide students in solving real-world engineering problems. For instance, for an "Intelligent Customer Service Robot" project, an enterprise mentor proposed a dialogue request scheduling solution for high-concurrency scenarios, supplementing industrial-grade solutions not covered in the curriculum. Furthermore, enterprises provide internship positions and technical documentation resources, converting internal training materials into elective content for maker training, thereby broadening students' technical horizons.

2.3. Three-Tier Competency Progression System: Systematically Designed Training Content

Based on the pattern of student competency growth, a three-tier competency progression system-Basic, Advanced, and Excellence-is constructed. Each tier has its specific focus while being interconnected, as detailed in Table 1.

Table 1. Three-Tier Competency Progression System

Tier	Goal Positioning	Core Content	Supporting Platform
Basic Tier	Consolidate technical foundation and collaborative awareness	Programming languages, data structures, software development toolchain training; Pair programming, small-scale system projects	University laboratories, Open-source communities
Advanced Tier	Cultivate innovative thinking and engineering capabilities	Algorithm optimization, system architecture design, cross-platform development; Provincial / National competition projects (e.g., "Smart Car Navigation System")	University-level innovation bases, School-enterprise joint laboratories
Excellence Tier	Strengthen entrepreneurial ability and holistic vision	Business model design, investment pitching, product iteration management; Venture project incubation (e.g., commercial operation of campus service APPs)	University science park, Business incubators

3. Practical Effectiveness and Verification

3.1. Significant Improvement in Students' Technical Capabilities

Through the "Competition-Education Integration" content training, students demonstrated outstanding performance in technical mastery and application skills. Over the past three years, the certification pass rate for programming languages among participating students was significantly higher than that of average students; in competitions such as the "China Software Cup" and the "Blue Bridge Cup," the award rate for technical categories exceeded 40%. Furthermore, students showed a stronger awareness of engineering standards in course design, with 85% able to independently complete comprehensive project development documentation, a marked increase compared to before the reform.

3.2. Effective Cultivation of Interdisciplinary Collaboration Skills

The interdisciplinary content integrated into the three-tier system broke down the disciplinary barriers of software majors. In the Advanced tier's "Intelligent Hardware Development" project, students needed to collaborate with Electronic Information majors on hardware circuit design and work with Industrial Design students to optimize product aesthetics. Participating students reported acquiring cross-professional communication skills and demand translation methods through the projects. The proportion of interdisciplinary teams increased in national competitions. For example, in the "Intelligent Orthopedic Rehabilitation Medical System" project, software students were responsible for algorithm development, while medical students provided rehabilitation needs guidance, demonstrating the practicality of the interdisciplinary content training.

3.3. Typical Case Analysis

Case: Student Wang (graduated in 2021 with a degree in Software Engineering, currently a Backend Development Engineer at an internet company)

Training Path: Joined the Basic tier training in freshman year, completed Python programming and data structure studies, developed a personal blog website; entered the Advanced tier in sophomore year, participated in the university-level innovation training project "Campus Second-hand Trading Platform," responsible for backend interface development and learning microservices architecture design; joined the Excellence tier in junior year, participated in the "Intelligent Waste Sorting and Recognition System" project, enterprise requirements, and optimized image recognition algorithms.

Feedback on Capability Improvement: "The toolchain training in the Basic tier helped me quickly adapt to the development environment at work. The competition projects in the Advanced tier taught me how to decompose complex requirements. The enterprise collaboration project in the Excellence tier enabled me to master industrial-grade algorithm optimization methods. This content training laid a solid foundation for my career development."

4. Challenges and Improvement Directions

4.1. Existing Problems

Insufficient Integration of Courses and Training Content: Some professional courses still focus on theoretical teaching and have not integrated engineering cases from maker training (e.g., distributed system fault diagnosis) into the classroom, resulting in a disconnect between "course content and practical content."

Content Update Pace Lags Behind Industry Development: Although cutting-edge industry technologies like AIGC and large model development have been incorporated into the Excellence tier, relevant teaching cases and training projects are insufficient, making it difficult to meet students' deep learning needs.

Unbalanced Resource Allocation in the Basic Tier: High-quality instructors and equipment are disproportionately allocated to the advancing and excellence tiers. Students in the Basic tier receive limited support in tool usage guidance and project selection, affecting learning outcomes at the introductory stage.

4.2. Improvement Directions

Deepen the Integration of Courses and Training Content: Promote the restructuring of professional courses by transforming project cases from maker training into course lab content. For example, introduce database design cases into courses, requiring students to complete table structure design, index optimization, and transaction processing programming, achieving "integration of course and competition content."

Dynamically Update Training Content: Establish an industry demand research mechanism. Jointly update the competency map with enterprises each semester, add new training modules such as AIGC application development and large model fine-tuning, and develop new projects like "AI-assisted Software Development Tools" and "Intelligent Document Generation System" to ensure content remains synchronized with the industry frontier.

Optimize Resource Allocation for the Basic Tier: Establish a "Basic Tier Mentor Pool" by selecting outstanding senior students and junior enterprise engineers as basic mentors. Increase the opening hours of basic laboratories to ensure the learning needs of Basic tier students are met.

5. Conclusion

The "Competition-Education Integration" content construction framework and the three-tier competency progression system established in this paper, through multiple driving models and systematic content design, effectively address issues such as content fragmentation, insufficient interdisciplinary integration, and disconnection from industry in software maker training, significantly enhancing students' technical capabilities and interdisciplinary collaboration skills. Future efforts should further deepen course integration and content updates, optimize resource allocation, and provide more robust content support for cultivating innovative talents adapted to the digital economy era in software majors.

Acknowledgments

This work was supported by the 2023 Hebei Province Innovation and Entrepreneurship Education Teaching Reform Research and Practice Project (2023cxcy211) and the Innovation and Entrepreneurship Curriculum Project.

References

- [1] Ministry of Education of the People's Republic of China. Guiding Opinions on Accelerating the Construction of High-Level Undergraduate Education and Comprehensively Improving Talent Cultivation Ability [Z]. 2018.
- [2] Wang Bowen. Innovation and Entrepreneurship Education in Universities Empowering the Development of New Quality Productive Forces: Internal Mechanism, Realistic Obstructions, and Practical Paths [J]. *Education Theory and Practice*, 2025, 45(15): 3-7.
- [3] China Association of Higher Education. China University Innovation Education Development Report (2023) [R]. 2023.
- [4] Wang Shiqiang, Ge Lili, Song Yan. Exploration and Practice of Academic Evaluation Reform in Higher Education under the Background of Top-Notch Innovative Talent Cultivation [J]. *China University Teaching*, 2024, (12): 43-49.
- [5] Chen Jing, Lu Ying. The Practical Education Path of the University Innovation and Entrepreneurship Education Ecosystem--Based on a Four-Dimensional Synergy Framework of Society, Family, School, and Individual [J]. *Higher Engineering Education Research*, 2025, (S1): 108-114.