**International Journal of Education and Social Development**
ISSN: 3078-9931 (Print), ISSN: 3078-2287 (Online) | Vol. 5, No. 1, 2025

# Design and Practice of a Three-Phase "Practice-Innovation-Entrepreneurship" Path for Software Maker Training

**Mingquan Zhang, Huawei Mei, 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 urgently needs to address issues such as the disconnection in traditional training paths and the disjunction between practice and innovation. Based on two decades of teaching practice, this paper constructs a three-phase maker training path of "Practice Foundation—Innovation Breakthrough—Entrepreneurship Incubation". This path consolidates the practical foundation through open laboratories, course-competition integrated curricula, and a student mentor and teaching assistant system. It achieves innovation breakthroughs by engaging in research projects, tiered competitions, and interdisciplinary workshops. Finally, it promotes the commercialization of outcomes through entrepreneurship courses, incubation platforms, and industry chain docking. Empirical evidence shows that this path can effectively enhance students' fundamental skills, innovative thinking, and entrepreneurial capabilities, providing a replicable path paradigm for software maker training.

**Keywords:** Software Maker Training, Three-Phase Path, Practice Foundation, Innovation Breakthrough, Entrepreneurship Incubation.

## 1. Introduction

### 1.1. Demand Background

The report of the 20th National Congress of the Communist Party of China explicitly called for "accelerating the implementation of the innovation-driven development strategy," requiring universities to cultivate high-quality talents with a complete capability chain of "practice, innovation, and entrepreneurship". In the era of the digital economy, software professionals not only need to master core technologies such as programming and algorithms but also require the ability to transform technology into commercial value [1]. However, traditional training paths often focus on single aspects, either concentrating solely on course experiments or conducting entrepreneurship training in isolation, making it difficult to support the progressive growth of student capabilities. There is an urgent need to build a coherent and systematic training path.

### 1.2. Current Situation and Problems in Training Paths for Software Majors in Universities

Current training paths for software majors in universities face three core problems:

Path Disconnection: Practical teaching is often confined to course experiments and short-term competitions, lacking a bridging mechanism from basic skills to innovative development, and further to entrepreneurship incubation. Student innovation outcomes mostly remain at the prototype stage, struggling to achieve commercial implementation [2].

Unclear Objectives: Training content has not formed a hierarchical and progressive system. Students with weak foundations and those with outstanding abilities receive homogeneous training, which neither solidifies the foundation nor stimulates innovative potential.

Value Disjunction: Training often aims at "completing tasks," lacking guidance on cultivating students' "national and societal" sentiment and awareness of social value. Student motivation primarily stems from external drivers such as awards and postgraduate recommendations, leading to insufficient autonomous development awareness [3, 4].

## 2. "Practice-Innovation-Entrepreneurship" Three-Phase Path Design Framework

Based on the patterns of student ability growth and industry needs, the three-phase path revolves around the core logic of "Foundation Consolidation—Ability Breakthrough—Value Transformation". The objectives, measures, and outcome orientations for each phase are clearly defined, forming a closed-loop education system. The specific framework is shown in Table 1.

**Table 1.** Three-Phase Path Framework

| Phase | Core Objective | Key Measures | Outcome Orientation |
|---|---|---|---|
| Practice Foundation | Master basic technologies, establish collaboration awareness | Open laboratories, Course-competition integrated courses, Student mentor and teaching assistant system | Ability to independently complete small-scale software projects, master basic skills like Git and the development toolchain |
| Innovation Breakthrough | Cultivate innovative thinking, enhance engineering capability | Research projects integrated into teams, Tiered cultivation for disciplinary competitions, Interdisciplinary workshops | Publish academic papers/apply for patents, win provincial-level or above competition awards |
| Entrepreneurship Incubation | Possess business thinking, achieve outcome transformation | Entrepreneurship courses, Incubation platforms, Industry chain docking | Incubate startup projects |

# 3. Specific Implementation Strategies for the Three-Phase Path

## 3.1. Practice Foundation Phase: Building a Solid Foundation in Technology and Collaboration

This phase focuses on "transitioning from knowledge application to skill consolidation," building a practical base through three major measures:

### 3.1.1. Open Laboratories: Creating Autonomous Practice Spaces

Laboratories operate on a "24-hour reservation system," divided functionally into "Basic Development Zone," "Smart Hardware Zone," and "Comprehensive Testing Zone," equipped with 3D printers, embedded development boards, cloud computing servers, etc., to meet practical needs combining software and hardware. When students develop a "Smart Temperature and Humidity Monitoring System," they can complete sensor and development board connection debugging in the "Smart Hardware Zone" and conduct 72-hour stability tests in the "Comprehensive Testing Zone." Simultaneously, the laboratory establishes a "student self-management system," where students are responsible for equipment registration, fault reporting, and project progress tracking, fostering a sense of responsibility and practical norms.

### 3.1.2. Course-Competition Integrated Courses: Bridging Practice and Competitions

In courses like "Software Innovation Practice Technology" and "Innovation and Entrepreneurship Methods and Practice," "competition-style assessment" is incorporated into the overall grade. For example, student teams developing a "Campus Life Service APP" need to include core modules like lost and found, campus notifications, and second-hand trading, and demonstrate the technical solution and functional implementation through live presentations. Outstanding works are directly recommended for the university-level "Software Design Competition," forming a progressive channel of "coursework — university-level competition — provincial-level competition".

### 3.1.3. Student Mentor and Teaching Assistant System: Passing on Technology and Culture

Select outstanding senior students, such as competition award winners and project leaders, to serve as "Student Mentors," adopting a "one-on-one" model to guide junior students. Mentors conduct weekly technical sharing sessions covering Git version control, Markdown documentation writing, basic algorithms, etc.; and provide project guidance every two weeks, helping junior students solve problems like code debugging and requirements analysis. Approximately 100 junior students are trained annually through this system. Mentors also impart the team culture of "dedication and collaboration" during guidance, shifting student motivation from "interest-driven" to "value-driven". Participating junior students reported that mentor guidance significantly improved their learning efficiency.

## 3.2. Innovation Breakthrough Phase: Stimulating Technological and Thinking Innovation

This phase focuses on "transitioning from technical problem-solving to value creation," promoting students to break through their ability boundaries through three measures:

### 3.2.1. Integrating Research Projects into Teams: Promoting Innovation through Research

Faculty release research topics like edge computing and blockchain. Students form teams to apply independently and, after a defense, join the research. Teams must complete the entire process from literature review, algorithm design, simulation experiments, to paper writing. For instance, in an edge computing project, students designed a load balancing-based task scheduling algorithm, verified it via MATLAB simulation, reducing task response latency by 20%. Through project training, students' scientific research and innovation capabilities are significantly enhanced.

### 3.2.2. Tiered Cultivation for Disciplinary Competitions: Precise Need Matching

Differentiated training plans are developed for different types of competitions:

Algorithm Competitions (e.g., ACM): Form "Algorithm Task Forces" with daily programming problem training, weekly sessions on solution approaches, and monthly mock contests. Training focuses on algorithm efficiency optimization, requiring students to optimize sorting algorithms from $O(n^2)$ to $O(n \log n)$. The team has repeatedly achieved good results in ACM regional contests in recent years.

Application Competitions: Invite industry experts to guide from both "technical feasibility + market value" dimensions. Guided by experts, participating projects have won provincial third prizes.

Entrepreneurship Competitions (e.g., "Internet Plus"): Introduce tools like Business Canvas and SWOT analysis, organize student visits to enterprises for research, and design personalized learning plans, also achieving good results.

### 3.2.3. Interdisciplinary Innovation Workshops: Breaking Professional Barriers

Establish three types of workshops: "Software + Hardware," "Technology + Design," and "Engineering + Management." Each workshop consists of teams of students

from different majors. Taking the "Smart Hardware Development Workshop" as an example, computer science students are responsible for APP development and algorithm design, electronic information students handle hardware circuit design, and industrial design students work on product appearance and interaction design, collaborating to complete the project. Workshops are equipped with interdisciplinary supervisors who hold monthly progress seminars to address issues like "data exchange delays between software and hardware" and "conflicts between appearance design and functional implementation".

### 3.3. Entrepreneurship Incubation Phase: Promoting Commercial Transformation of Outcomes

This phase aims at "transitioning from prototype verification to commercial implementation," achieving value transformation through three major supports:

#### 3.3.1. Entrepreneurship Course Modules: Cultivating Business Thinking

Offer elective courses and lectures like "Innovation and Entrepreneurship Methods and Practice," inviting alumni entrepreneurs and partners from investment institutions to teach. Courses adopt a "case teaching + practical simulation" model, requiring students to use their innovation projects as prototypes to complete modules like market analysis and financial forecasting. "Roadshow Simulation Competitions" are organized, with investors providing on-site feedback to help students optimize their business logic.

#### 3.3.2. Incubation Platform Support: Providing Implementation Guarantees

Leverage university-level entrepreneurship parks and university science parks to provide projects with one-stop services including "office space + legal consultation + financial agency." The entrepreneurship park establishes a "Maker Space" providing office equipment and meeting venues free of charge for projects; the science park assists with company registration and intellectual property applications.

## 4. Path Implementation Effectiveness and Verification

### 4.1. Significant Improvement in Student Capabilities

In terms of basic skills, 92% of students participating in the path training were able to independently master the development toolchain and version control technologies.

In terms of innovation capability, the number of provincial-level and above competition awards won by students increased by an average of 20% annually.

Outstanding Employment Quality: Tracking of past graduates shows that the employment rate of students who participated in the path training reached 98%; their average starting salary was also higher than that of ordinary students, with many graduates joining leading internet companies like ByteDance and Alibaba.

Career Development: The average time for club members to get promoted to technical backbone or management positions was shortened by 1.5 years. For instance, a 2021 graduate, Ms. Zheng, was promoted to Technical Director at a tech company within three years, responsible for a development team of over 10 people.

### 4.2. Typical Case: The Growth Path of the "Image Geek" Project

Practice Foundation: Founder, Ms. Lu, joined the interest group in her freshman year, completed basic projects on film and television image processing in the open laboratory, mastering tools like Photoshop and Premiere.

Innovation Breakthrough: In her sophomore year, she participated in the "University-level Digital Media Competition," developed an "Intelligent Video Editing Plugin," won the first prize, subsequently optimized the algorithm, and applied for software copyright.

Entrepreneurship Incubation: Learned business model design through entrepreneurship courses, secured investment through industry chain docking meetings. The company is now it has taken shape, serving clients in education, cultural tourism, and other fields.

## 5. Challenges and Improvement Directions

### 5.1. Existing Problems

Unsmooth Path Transition: Some students struggle to transition from the practice phase to the innovation phase due to unconsolidated basic skills; outcomes from the innovation phase often fail to effectively enter the entrepreneurship phase due to a lack of business guidance.

Uneven Resource Distribution: High-quality equipment and mentor resources tend to be allocated more towards the innovation and entrepreneurship phases, while students in the practice phase receive limited guidance.

Incomplete Evaluation System: Ability evaluation across path phases remains outcome-oriented, lacking tracking and feedback on students' procedural growth.

### 5.2. Improvement Directions

Add "Bridge Projects": Set up "Basic Skill Standard-Reaching Projects" between the practice and innovation phases, requiring students to complete small-scale system development and pass a defense before entering the innovation phase; establish a "Business Pre-incubation" phase between the innovation and entrepreneurship phases, inviting alumni mentors to intervene early to assess the commercial viability of outcomes.

Optimize Resource Allocation: Increase the number of mentors in the practice phase, establish a "Basic Skills Special Fund" to purchase more entry-level development equipment; establish a resource rotation mechanism to ensure students at all stages can access core equipment.

Construct a Dynamic Evaluation System: Introduce ability growth portfolios to record students' progress in technical mastery, collaboration skills, innovative thinking, etc. Adopt a multi-dimensional evaluation method combining "student self-assessment + mentor evaluation + enterprise evaluation" to achieve a combination of procedural and outcome-based assessment.

## 6. Conclusion

The "Practice-Innovation-Entrepreneurship" three-phase path, through its layered design and interlocking implementation strategies, effectively addresses issues like path disconnection, weak foundations, and difficulty in outcome transformation in software maker training. It facilitates the leap in student capabilities from "knowing

technology" to "being able to innovate" and further to "being adept at entrepreneurship". In the future, it is necessary to further optimize path transitions and resource allocation, improve the evaluation system, and provide stronger support for cultivating innovative and entrepreneurial talents in software majors adapted to the demands of the digital economy era.

## Acknowledgements

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

## References

[1] Ministry of Education of the People's Republic of China. Opinions on Accelerating the Construction of High-Level Undergraduate Education and Comprehensively Improving Talent Training Capacity [Z]. 2018.

[2] Chen Jun, Xiao Yuan, Wang Fengping. Exploration and Research on the Cultivation of Innovative Talents in Computer Majors for Emerging Engineering Education [J]. Education and Teaching Forum, 2025, (17): 153-156.

[3] Huang Zhitong, Ji Yuefeng, Yin Changchuan. Exploration and Practice of First-Class Undergraduate Course Construction for Innovative Talent Cultivation in Electronic Information Majors [J]. China University Teaching, 2021, (10): 43-48.

[4] Zhou Yan, Zeng Fanzhi, Yang Guangfa. Innovation Talent Training Model in Computer Science Driven by Technology Competitions and Integrating Multiple Knowledge Points [J]. Computer Education, 2012, (16): 19-22.