Education Reform and Development, 2025, Volume 7, Issue 10
https://ojs.bbwpublisher.com/index.php/ERD
Online ISSN: 2652-5372
Print ISSN: 2652-5364

# Exploration and Research on the "AI +" Talent Cultivation Model for Computer Software Engineering majors in Higher Education Institutions

**Huiying Zhang***, **Haoyi Xie**, **Yule Sun**

Nanjing Tech University Pujiang Institute, Nanjing 211200, Jiangsu

**Author to whom correspondence should be addressed.*



**Abstract:** Against the backdrop of the advancement of the national "AI +" strategy and the intelligent upgrade of the software industry, enterprises have an increasingly urgent demand for compound talents who "understand development and are intelligent", but the traditional training model of software engineering in colleges and universities has obvious shortcomings and is difficult to meet this demand. This paper takes the computer software engineering major in application-oriented universities as the research object, systematically sorts out the problems existing in the current training mode, and proposes targeted four-dimensional solutions of "curriculum reconstruction, practice strengthening, faculty construction, and evaluation optimization". Practice has shown that this model can effectively enhance students' interdisciplinary application ability and job fit, providing a practical reference for the reform of software engineering education.

**Keywords:** Software Engineering; AI +; Talent development model; Curriculum integration; School-enterprise collaboration

**Online publication:** November 14, 2025

## 1. Introduction

With the implementation of policies such as the "Next Generation Artificial Intelligence Development Plan" and the "14th Five-Year Plan for Software Industry Development", the deep integration of artificial intelligence and the software industry has become the core direction of industrial upgrading, and the software industry has transformed from traditional coding to intelligent development. Enterprises are in urgent need of compound talents who master development technologies such as Java and Python and can apply machine learning and computer vision to solve practical engineering problems, such as AI-driven software systems [1].

However, there is a significant mismatch between the supply and demand of software engineering professionals in colleges and universities [2]. On the one hand, the curriculum system is dominated by single-discipline knowledge, and AI-related content is mostly elective and disconnected from core courses such as software design and testing, making it difficult for students to combine AI technology with software engineering practice [3,4]. On the other hand, practical teaching often uses simulated projects and lacks real "AI + software engineering" integration scenarios, resulting in students having a long time to adapt to their jobs after graduation. Therefore, exploring the "AI +" talent cultivation model has become the key to resolving the contradiction between educational supply and industrial demand [5,6].

## 2. Problems Existing in the cultivation of "AI +" talents in software engineering at present

### 2.1. Low integration of AI with software engineering courses

Core software engineering courses, such as Software Project Management and Software Development Technology, focus on coding and process management. AI courses such as Fundamentals of Machine Learning are mostly general elective courses and do not incorporate key elements of the software development life cycle. For example, Software Testing Techniques still mainly focus on manual test case design and do not introduce AI automated testing tools such as Applitools Eyes [4]. Software Design does not cover intelligent architecture optimization based on reinforcement learning, and students cannot translate AI technology into engineering capabilities. Meanwhile, the course lags behind industry development. Currently, AI technology has evolved towards large model development and low-code platform intelligent plugins, but most universities still mainly teach traditional machine learning algorithms, such as decision trees and SVM, and do not cover the application of AI-generated code tools such as ChatGPT-assisted development and CodeLlama. There is a gap between students' knowledge reserves and the demands of enterprises [7].

### 2.2. The practical teaching scenarios are virtual, and the collaboration between schools and enterprises is shallow

There are "two virtual" issues in the practical process: First, there is a "virtual project." Most practical projects are simulated scenarios, such as the development of library management systems, without integrating AI technology, which cannot train students' ability to solve engineering problems with AI [8]. Second, the cooperation between the school and the enterprise mostly remains at the level of lectures by enterprise experts and short-term visits by students, without establishing a deep collaborative mechanism of jointly designing courses, jointly guiding projects, and jointly evaluating achievements. Students have difficulty accessing real "AI + software" projects, such as the development of intelligent operation and maintenance systems and the design of industrial software AI modules. The practical skills are hard to meet the job requirements [9].

### 2.3. The teaching staff has weak interdisciplinary capabilities

Most of the current faculty members have a single background. Software engineering teachers are familiar with development techniques but lack practical experience in AI and are unable to efficiently teach integrated courses such as AI-driven software optimization [10]. AI teachers are good at algorithmic theory but not familiar with software engineering processes, making it difficult for them to guide students through the entire process from AI models to software implementation. In addition, teachers have few opportunities to practice in enterprises. Most of them have not participated in "AI + software" industry projects. Teaching cases are mainly based on textbook examples, lacking practicality and unable to provide students with teaching guidance that is in line with industry reality [11].

### 2.4. The evaluation system emphasizes theory over application

The traditional evaluation system, which is based on test scores plus code submission, has obvious limitations. First, the evaluation dimension is single and does not include indicators such as the application effect of AI technology and the degree of project engineering, making it impossible to comprehensively measure students' interdisciplinary abilities. Second, the evaluation subjects are limited, scoring only by teachers within the school, assessing only the correctness of code operation, without evaluating core indicators of concern to enterprises such as concurrent processing capacity of the system, making it difficult to reflect students' true abilities. The evaluation results are disconnected from job requirements without the inclusion of enterprise experts' opinions [12,13].

## 3. Countermeasures for optimizing the cultivation of "AI +" talents in software engineering programs in colleges and universities

### 3.1. Reconstruct the integrated curriculum system of AI + software engineering

With the software development life cycle as the logical thread, break down AI technology into five modules: requirements analysis, design, development, testing, operation and maintenance, and embed them in core courses. Add an "AI Requirements Forecasting" module to the Introduction to Software Engineering to guide students to predict software functional requirements using regression analysis algorithms. In Software Design, add "Intelligent Architecture Design" and teach to optimize module division using reinforcement learning; Introduce the application of AI-generated code tools in Software Development Technology to train students to use AI to assist in development. In Software Testing Technology, incorporate "AI Automated testing" to enable students to develop test tools based on image recognition, and in Software Operation and Maintenance, add "intelligent Fault prediction" to implement software log anomaly detection using LSTM algorithms. At the same time, "AI + Software Engineering" elective directions such as "Intelligent Software Development" and "AI-driven Testing and Operations" are set up, along with cutting-edge courses such as "Large Models and Software Development" to fill the knowledge gap.

### 3.2. Strengthen practical teaching and deepen school-enterprise collaboration and platform construction

(1) Build practice bases, sign cooperation agreements with local IT enterprises, enterprises provide real projects such as system intelligent monitoring software development, send engineers as off-campus mentors, universities organize students to participate in development in the form of project teams, submit requirements documents, design plans, code implementation, test reports and other complete engineering results.

(2) Upgrade the on-campus experimental platform, build an "AI + software engineering" integrated laboratory, configure GPU servers supporting TensorFlow [14] and PyTorch [15], deploy industrial-grade development environments such as Jenkins and JIRA, and develop virtual simulation project libraries such as medical image analysis software, To make up for the long cycle and limited participation of enterprise projects.

(3) Implement progressive practice projects. At the basic level, focus on single technology integration

projects such as CNN-based software development. At the advanced level, focus on system module integration projects. At the advanced level, connect with real enterprise demands such as intelligent monitoring software for industrial equipment to achieve step-by-step improvement of students' abilities.

### 3.3. Build a dual-qualified, interdisciplinary teaching team

(1) Conduct in-school teacher training. Regularly organize software engineering teachers to participate in AI core technology training, such as deep learning and computer vision. Only those who pass the assessment can undertake integrated courses. Teachers are required to be on secondments to enterprises for 2–3 months every two years to participate in the research and development of "AI + software" projects, such as the development of intelligent testing tools, and convert practical experience into teaching cases.

(2) Adopt a combination of an introduction and a part-time approach. Introduce high-level talents with a doctoral degree in software engineering and experience in AI enterprise research and development to be responsible for teaching core integrated courses. Hire enterprise AI software research and development leaders as industry professors and give 2–3 special lectures each semester to guide students' advanced projects.

(3) Promote cross-university faculty exchanges, establish cooperation with top universities at home and abroad, send teachers to exchange experiences in the design of "AI + software engineering" courses, share teaching resources, and enhance the overall level of the faculty team.

### 3.4. Establish a diversified and dynamic evaluation system

(1) Expand the evaluation dimensions and set indicators in three aspects: knowledge, ability, and quality. The knowledge dimension includes integrated course examination results and AI technology theory defense. The ability dimension covers project achievements such as requirement document completeness, code quality, and AI algorithm effect. The quality dimension involves teamwork, project progress management, and document standardization to comprehensively measure students' comprehensive abilities.

(2) Introduce multiple evaluation subjects: in-school teachers evaluate theoretical mastery and code standardization, enterprise mentors evaluate the degree of project engineering and job fit, and students evaluate team collaboration performance to form a "three-party evaluation" mechanism to ensure that the evaluation results are in line with job requirements

(3) Improve the dynamic feedback optimization mechanism. At the end of each semester, collect student learning feedback, enterprise employment evaluation, and teacher teaching reflection, and adjust the course content in a timely manner (such as adding the "large model software application" module) and optimize the practical projects to achieve continuous iteration of the training model.

## 4. Conclusion

This paper addresses the pain points of "AI +" talent cultivation in software engineering programs in colleges and universities and proposes a four-dimensional optimization strategy of curriculum, practice, faculty, and evaluation. The core is to break the traditional model of the separation of AI and software engineering, and cultivate compound talents who understand development, are intelligent, and can be implemented through curriculum integration, school-enterprise collaboration, faculty upgrading and evaluation innovation. In the future, the scope of school-enterprise cooperation can be further expanded, industrial colleges can be established to achieve “order-based” training, and the curriculum content can be dynamically updated to adapt to the evolution of AI technology, continuously improving the quality of talent cultivation and providing intellectual support for the upgrading of the software industry.

## Funding

2025 China University Computer Basic Education Society Project (Project No.: 2025-AFCEC-113); Nanjing Tech University Pujiang College Young Backbone Teachers Project(Project No.: 2025PJYQ04); Key Natural Science Project of Nanjing Tech University Pujiang College 2024 (Project No.: njpj2024-1-01).

## Disclosure statement

The authors declare no conflict of interest.

## References

[1] Yang C, Chi R, Li C, 2020, Research on the Training Mode and Strategy of Software Engineering Technology. Journal of Shenzhen Polytechnic University, 24(5): 99–104.

[2] Dong Z, Lin Y, Cong R, 2025, New Engineering Curriculum Optimization and Evaluation System Reconstruction from the Perspective of Systems Theory: A Case Study of Computer Vision Course. Computer Education, 2025(6): 168–172 + 177.

[3] Lang C, Qian H, Wu S, et al., 25, Research on the Training Mode of Computer-related Compound Talents in the Context of New Engineering. Electronic Components and Information Technology, 9(1): 234–237.

[4] Liu Y, Wei J, Li J, 2014, Research and Application of Intelligent Testing Tools and Processes. Oil and Gas Well Testing, 23(2): 71–74 + 78.

[5] Zhang H, 2023, Research on the Training of Computer Talents in Private Colleges. Electronics Science Technology and Application, 2023(10): 71–73.

[6] Zhang H, Mu Y, 2024, Research on Curriculum Construction of Artificial Intelligence under the Background of New Engineering. Education Reform and Development, 2024(4): 108–111.

[7] Tian C, Chen L, Wang S, et al., 25, Macrocell Layout Method for Intelligent Chip Based on Graph Convolutional Network and Reinforcement Learning. Shanghai Aerospace (Chinese and English), 42(5): 167–177.

[8] Zhang H, 2024, Research on the Application-oriented Talent Cultivation System of Software Engineering in Independent Colleges. Internet Weekly, 2024(1): 65–67.

[9] Harris N, 2025, AI Technology in the Application of Computer Software Development. Information Recording Materials, 26(10): 121–123.

[10] Zhang H, 2022, Research on the Reform of Database Engineering Teaching in Applied Undergraduate Education. Jiangsu Science and Technology Information, 39(17): 53–56.

[11] Cai Y, Zeng A, 2025, “12345” Integration of Government, Industry, Academia, Research and Application for Innovative Talent Cultivation in Software Engineering. Computer Education, 2025(9): 131–135.

[12] Chen H, Du J, 2025, Exploration of Teaching Reform of “Time Series Analysis” Course Based on “SARIMA-BP/

SVM/RF" Joint Model and R Language. Chinese Journal of Health Statistics, 42(4): 632–636.

[13] Shen H, Chen Y, Gu P, et al., 2020, Case Study and Practice of "Disaster Monitoring and Early Warning Technology" Course Based on LSTM Model: Taking Rainfall Prediction as an Example. Industrial Control Computers, 38(8): 106–108.

[14] Gao Z, Tang L, Gu R, 2023, Exploring the Development of an Innovative Case Curriculum Standard for "TensorFlow Basic Practice". Knowledge Library, 2023(3): 37–39.

[15] Lu J, Wang L, Hou M, 2022, Application and Prospect of Deep Learning-based PyTorch in Flight Safety Teaching. Educational Teaching Forum, 2022(46): 137–140.