PCER: A Dynamic Feedback Model for Optimizing Graduate Student Training in Software Engineering

Bangchao Wang, Yuyang Dai, Hongyan Wan*, Xinrong Hu, Xiong Wei

Abstract: Under the national innovation-driven development strategy and the emerging need for new engineering education, the cultivation of high-quality software engineering talent faces increasingly stringent requirements. To address critical challenges in the current graduate training model at regional universities—including fragmented undergraduate-to-graduate transition, ambiguous competency development pathways, and structural deficiencies in teaching resources—this paper proposes a novel PCER (Plan-Carrying Out-Evaluate-Refine) dynamic feedback model for software engineering graduate education. The PCER framework establishes phased training objectives, develops an adaptive multi-level resource repository, and implements a multi-dimensional evaluation mechanism. Empirical results demonstrate that the model significantly enhances students' academic innovation capabilities and engineering practice competencies, providing a replicable reform path for the cultivation of software engineering talents. The PCER approach offers a transferable solution for engineering education reform, particularly for institutions facing similar resource constraints and quality improvement challenges.

Keywords: Software engineering; Training model; Graduate education; Teaching resources

1 Introduction

With the continuous development of China's society and economy, increasing demands have been placed on the quality of talent cultivation in higher education[1]. Especially in the field of software engineering, with the rapid development of technologies such as artificial intelligence and big data, the demand for professional talents is growing steadily and becoming increasingly diversified.[2]. However, the current training pathways for software engineering graduate students at regional universities are unclear, and the structural shortage of teaching resources limits the effectiveness of cultivating high-level talent[3]. To address these challenges, this study proposes a PCER (Plan-Carrying out-Evaluate-Refine) model inspired by quality management principles[4]. The PCER model represents an innovative adaptation for educational contexts, comprising four iterative phases that emphasize continuous improvement through dynamic feedback mechanisms. As shown in Figure 1, The model mirrors the essential stages of talent cultivation: (1) establishing clear training objectives (Plan), (2) implementing curriculum and research activities (Carrying out), (3) systematically assessing learning outcomes (Evaluate), and (4) refining the training process based on performance data (Refine).

Building upon the PCER framework, this study develops a novel software engineering graduate training model supported by an adaptive, multi-level resource repository and a multi-dimensional evaluation mechanism. The model specifically addresses three critical limitations in current training systems: (1) the disconnection between training phases, (2) insufficient feedback mechanisms, and (3) rigid resource allocation. By implementing the PCER cycle with its built-in refinement phase, the proposed model ensures continuous optimization of the training process while maintaining alignment with evolving industry requirements and technological advancements.

This study makes four significant contributions to software engineering graduate education:

(1) Propose an innovative PCER-based graduate training model that establishes a closed-loop mechanism for continuous improvement in talent cultivation.

(2) Develop a multi-level training ladder that implements differentiated cultivation strategies for distinct graduate program types (academic master's vs. professional master's).

(3) Develop an adaptive, multi-level resource repository that effectively bridges the gap between academic resources and industry requirements.

(4) Propose an integrated evaluation framework that combines quantitative assessment with systematic feedback mechanisms.

Collectively, the PCER model constructs an adaptive education ecosystem that addresses critical challenges in graduate training, particularly for institutions with

Bangchao Wang, Yuyang Dai, Hongyan Wan*, Xinrong Hu, Xiong Wei are with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, 430200, China. E-mail: wangbc@whu.edu.cn; 2415283041@wtu.edu.cn; hywan@wtu.edu.cn; hxr@wtu.edu.cn; wx_wh@wtu.edu.cn,.
* To whom correspondence should be addressed.

limited resources. The proposed approach demonstrates significant potential for enhancing training quality while maintaining responsiveness to evolving technological and industrial demands.

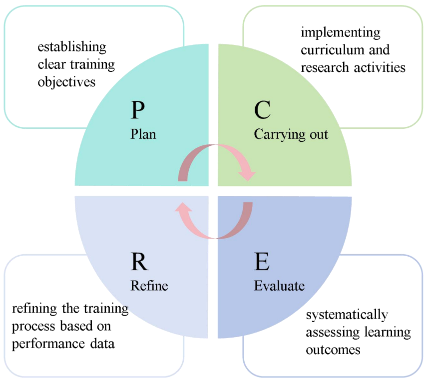

Figure 1: PCRE Cycle-Based Graduate Training Model

2 Challenges in Graduate Education

This chapter systematically examines the critical challenges confronting regional universities in software engineering postgraduate education. Focusing on three fundamental dimensions:(1) fragmented undergraduate-to-graduate transitions, (2) ambiguous developmental pathways, and (3) structural deficiencies in teaching resources. The analysis identifies systemic barriers to cultivating high-quality talent. These empirically observed challenges not only reveal the limitations of current training paradigms but also provide the foundational rationale for developing the proposed PCER-based training model.

2.1 Fragmented Transition Between Undergraduate and Graduate Education

The current higher education system exhibits a fundamental paradigm shift between undergraduate and postgraduate training objectives. At the undergraduate level, the predominant focus remains on fundamental knowledge acquisition and standardized exam-oriented proficiencies. In stark contrast, postgraduate education emphasizes research competency development, innovation thinking cultivation, and independent learning capabilities. However, due to a lack of connection between the undergraduate and postgraduate stages, many students enter graduate programs with weak research foundations and a limited understanding of basic research processes. Most are unfamiliar with research tools upon enrollment, and even lack knowledge of academic paper structures, citation standards, and submission procedures. This deficiency in research literacy not only delays the initiation of their research but also reduces the quality and efficiency of early-stage work. In addition, most postgraduates have had little or no experience with real-world engineering projects or work in enterprises prior to enrollment. As a result, they often underperform in research topics, project advancement, academic competitions, and industry collaborations, struggling to effectively apply theoretical knowledge to complex real-world engineering scenarios. This gap significantly affects both the depth of their research and the efficiency of translating research outcomes into practical applications.

2.2 Ambiguous Developmental Pathways in Graduate Training

In the cultivation of software engineering graduate students at regional universities, the ambiguity of training pathways is another major factor hindering the quality of talent development. With the continuous evolution of the software engineering discipline and the deepening of research, the field has become increasingly specialized. Subfields such as natural language processing, requirements engineering, and software traceability have developed into relatively independent research areas. However, existing training models generally lack a systematic top-level design, resulting in knowledge gaps and directional confusion during students' learning and research processes. Students often lack clear guidance in selecting research directions, planning research trajectories, and identifying areas for skill enhancement. The lack of structured guidance makes it difficult to establish a progressive and coherent development path, thereby undermining research efficiency and limiting their long-term academic and professional potential.

2.3 Structural Deficiencies in Teaching Resources

Under the background of rapid development in information technology, software engineering is evolving at a pace comparable to Moore's Law. Regional universities face significant structural contradictions in teaching resources for graduate education in software engineering. First, curricular content updates lag considerably behind technological progress. Many courses continue to employ outdated knowledge systems and methodologies, failing to incorporate cutting-edge research findings or industry developments. This disconnect is exacerbated when instructors utilize teaching materials developed years prior without substantive revisions, despite their ongoing research activities. Second, the translation of faculty research into pedagogical resources remains inadequate. While many professors maintain active research programs and industry collaborations, their scholarly outputs and practical experiences are rarely systematically adapted for classroom application. Third, the structural imbalance between theoretical and applied knowledge persists. Students consequently face limited exposure to contemporary technologies and research methodologies[5-6], leaving them ill-prepared for either academic research or professional practice. This resource gap is particularly acute in emerging domains like AI-driven software development and cloud-native architectures, where the disparity between academic instruction and industrial practice exceeds 3-5 years. These systemic deficiencies collectively undermine the quality of graduate education.

3 PCER Dynamic Feedback Model

To address the previously identified challenges in graduate education, this study proposes a novel PCER dynamic feedback training model. Inspired by the PDCA cycle theory, the model operates as follows: In the Plan phase, a multi-level training ladder is constructed by setting phased training objectives. This approach effectively alleviates the disconnect between undergraduate and graduate education, enhances students' foundational research and practical skills, and clarifies both long-term and interim goals for graduate training, ensuring that the training path is both scientific and executable. To address the structural shortage of teaching resources, the Carrying Out phase introduces a dynamic, tiered case resource repository. Faculty research outcomes and enterprise projects are transformed into hierarchical teaching cases, enriching the available teaching materials. In the Evaluate phase, the model establishes a multi-agent, multi-dimensional evaluation mechanism that comprehensively assesses the achievement of graduate students' phased training objectives through a combination of qualitative and quantitative methods. In the Refine phase, issues that emerge during the training process are systematically summarized and analyzed. Identified areas for improvement are fed back into the next Plan phase, enabling dynamic optimization and continuous enhancement of the entire training process in a closed feedback loop.

3.1 Plan Phase

In response to the differentiated training needs of academic and professional master's students, as well as individual differences in research foundation and practical abilities, this model centers on a hierarchical design. As shown in Figure 2, this study proposes a multi-level training ladder, which assists in establishing both short-term and long-term training objectives for students at different levels, clarifying their training pathways and achieving systematic cultivation.

3.1.1 Level 1: Classification of Training Types
Based on the fundamental differences in training objectives, academic master's and professional master's programs correspond to two distinct types of training. The academic master's program is research-oriented, with a long-term goal of cultivating academic talents who possess the ability to conduct independent research and produce systematic research outcomes within various fields of software engineering. In contrast, the professional master's program is application-oriented, aiming to develop applied talents capable of solving complex technical problems in industry. These graduates are expected to translate theoretical knowledge into practical engineering methods and independently undertake project development and implementation tasks.

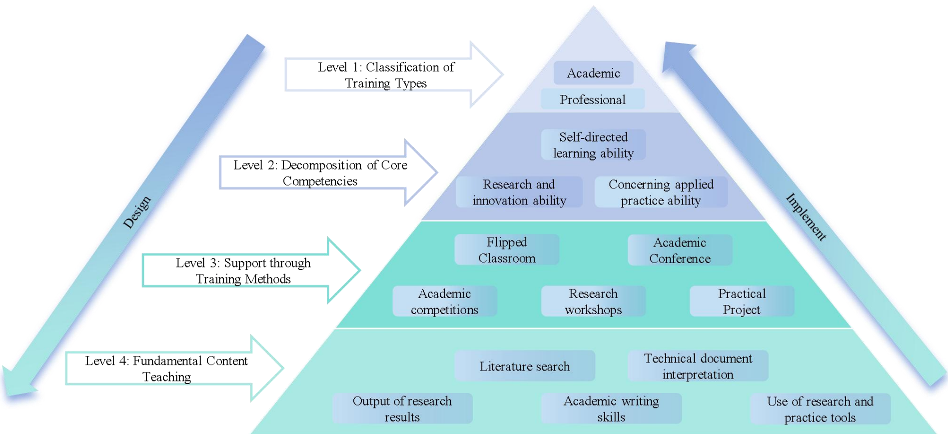

Figure 2:multi-level training ladder

3.1.2 Level 2: Decomposition of Core Competencies
Focusing on the training orientations mentioned above, the cultivation of the two types of master's students revolves around three core competencies: In terms of self-directed learning ability, both types of master's students need to master the ability to dynamically track the forefront of their fields. However, academic master's students focus on theoretical learning and domain knowledge integration centered on top-tier journals and conferences, while professional master's students emphasize the rapid application of new technologies and tool adaptation aligned with industry technology iterations. Regarding research and innovation ability, academic master's students center

on principled research, emphasizing theoretical innovation, methodological breakthroughs, and the expansion of knowledge systems, highlighting "scientific rigor, theoretical depth, and originality." Professional master's students focus on technological improvements and scenario innovation, emphasizing the optimization of technical solutions and the proposal of new application scenarios in practical projects, prioritizing "practicality, adaptability, and engineering orientation."Concerning applied practice ability, academic master's students focus on validating research outcomes, emphasizing data verification through experimental models to support theoretical innovation, whereas professional master's students focus on engineering implementation, including requirements analysis, system development, and project operation and maintenance. It is important to emphasize that these two training pathways are not completely separated or mutually exclusive in training the three core competencies, but differ in emphasis based on their respective training objectives.

3.1.3 Level 3: Support through Training Methods

In response to the core competency training objectives and considering the faculty strength and teaching resource conditions of the college, a diversified training approach covering the entire research and practice chain has been constructed: The flipped classroom transforms the core course content into pre-class learning tasks, requiring students to conduct preliminary knowledge exploration through literature retrieval and review of open-source resources. Classroom time is then focused on in-depth interactive sessions such as debates on academic controversies and the breakdown of complex problems. Regarding academic conferences, the college regularly holds graduate academic forums where students are required to present their stage-wise research achievements as presenters. Faculty members provide guidance to help improve their ability to communicate results and engage in academic exchanges. Meanwhile, the college actively encourages and funds graduate students to participate in various academic conferences to broaden their academic horizons and enhance their communication skills. In terms of academic competitions, the college encourages students to participate in various contests, incorporating competition results into scholarship evaluations and graduation requirements, further motivating student engagement in research competitions. For research workshops, students are required to regularly report their work progress and engage in discussions, implementing a process tracking mechanism to promote the efficiency of their research practice output. Regarding practical projects, leveraging the advantage of university-enterprise cooperation, the college jointly establishes graduate practice bases with enterprises, decomposes real enterprise projects into several manageable sub-tasks, and assigns these to individual students or groups for hands-on training.

3.1.4 Level 4: Fundamental Content Teaching

The training objective at this stage is to effectively bridge the learning content between the undergraduate and graduate levels. Through a systematic foundational teaching design, it aims to fill the knowledge gaps in students' research and practical abilities, laying a solid foundation for subsequent hierarchical training. All graduate students, regardless of training type, are required to master the following core content: First, in the literature search and technical document interpretation, students must acquire systematic literature retrieval and reading strategies, efficiently using databases such as Google Scholar and Web of Science. They should be able to rapidly extract the innovations from the literature. Additionally, they must possess the ability to accurately interpret technical documents, including open-source project manuals and API interface specifications. Second, regarding the use of research and practice tools, students are taught practical tools such as reference management software, data analysis tools, and Git, to facilitate efficient initiation and detailed management in their subsequent research activities and engineering projects. Third, in academic writing skills, students learn basic academic conventions, structural frameworks, and citation formats, helping them establish a standardized academic expression system that ensures clear logic and proper formatting when writing research outputs. Finally, concerning the output of research results, students are required to understand the submission processes for conferences and journals, master key procedures such as target journal selection, manuscript submission, and response to reviewers' comments, and clearly comprehend the standard pathways and requirements for transforming academic achievements into published works.

3.2 Carrying Out Phase

The Carrying Out phase is the core stage where the hierarchical training objectives set during the Plan phase are transformed into concrete actions. To address the common structural shortage of teaching resources faced by regional universities at this stage, this study proposes a dynamic, tiered case repository. This repository breaks down research outcomes from enterprises and faculty, as well as industrial projects, into individual resource cases, effectively supplementing the case inventory. In terms of resource design, the case repository is divided into three levels: basic, intermediate, and advanced.

Basic-level cases focus on bridging knowledge between undergraduate and graduate stages, selecting content such as classic algorithm implementations and standardized research procedures to help students quickly solidify foundational research and practical methods. Intermediate-level cases are derived from faculty research projects and real enterprise needs, decomposed into manageable sub-tasks, and supported by corresponding code libraries and literature databases to aid students in deeply understanding specific knowledge points and independent module functions. Advanced-level cases are categorized by different research fields, integrating top conference and journal results as well as industry technology bottlenecks, providing precise support for graduate students in accessing relevant materials during their research activities.

3.3 Evaluate Phase

The evaluation phase is conducted after the Carrying Out phase, as shown in Figure 3. By establishing a multi -agent, multi-dimensional evaluation mechanism and adopting a combination of qualitative and quantitative methods, this model comprehensively assesses the achievement of training objectives, providing data support and directional guidance for the next round of optimization. This mechanism involves multiple evaluators both inside and outside the university, including academic advisors, industry experts, and the students themselves, to conduct a comprehensive assessment of the stage-specific goals set during the planning phase. The evaluation dimensions cover the students' process, abilities, and outcomes achieved during the implementation phase. Specifically, academic advisors assess students based on task completion, quality of outcomes, and performance throughout the process, with a focus on their research capabilities, project management skills, and depth of knowledge. Industry experts evaluate students from the perspective of real-world engineering projects, emphasizing the completion and applicability of their work. Students engage in self-reflection and stage summaries to review personal growth and identify areas for improvement. In terms of evaluation dimensions, distinctions are made based on training type. For academic-oriented master's students, the focus lies on the scientific and innovative quality of research plans, the completeness and effectiveness of research outcomes, literature review ability, and academic writing skills. For application-oriented master's students, the evaluation emphasizes capabilities in problem modeling and analysis, technical implementation ability, and the applicability and transfer ability of their technical outcomes in practical projects.

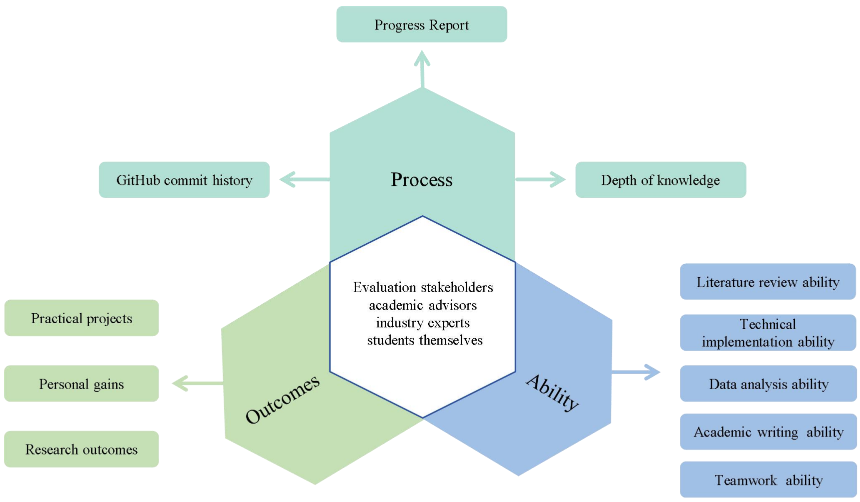

Figure 3:multi-dimensional evaluation mechanism

3.4 Refine Phase

The Refine phase serves as the concluding stage of the PCRE cycle. Its core function is to summarize and categorize problems encountered during the training process, consolidate successful practices into institutionalized standards, and provide adjustment references for the next Plan phase. This ensures the dynamic optimization and closed-loop improvement of graduate education. In this phase, the identification and classification of problems are critical. Accordingly, this study divides the main issues revealed during the training process into two categories: short-term solvable problems and long-term systemic problems, each addressed with corresponding handling strategies. Driven by the dual mechanisms of problem attribution and correction, this phase not only promotes precise management of existing issues but also provides practical foundations and reform directions for the subsequent Plan phase. Thus, it facilitates the efficient closed-loop operation of the PCRE cycle within the graduate training system.

4 Survey On Undergraduate-to-Graduate

Transition

To gain a deeper understanding of the current status of research foundations and practical abilities among software engineering graduate students at regional universities, this chapter designs and implements a structured questionnaire. Through quantitative analysis, the survey provides data support for the design of the training model.

4.1 Questionnaire Design

To comprehensively analyze the differences in research foundations and practical abilities faced by software engineering graduate students at regional universities during their initial enrollment, this study designed a questionnaire consisting of four main sections: basic information, research foundation, practical foundation, and overall evaluation and suggestions. Closely aligned with the previously identified issues—such as the undergraduate-to-graduate transition gap and weak research and practical foundations—the questionnaire adopts a structured design to collect sample data quantitatively. Due to space limitations, the full questionnaire can be accessed at: https://github.com/WTU-intelligent-software-development. The specific questionnaire items are integrated into the corresponding result figures.

4.2 Questionnaire Analysis

In this study, first-year postgraduate students were selected as the survey subjects, with a total of 47 valid responses collected, including 11 academic masters and 36 professional masters. The current status of undergraduate-postgraduate articulation was analyzed from three dimensions, and the specific dimensions are discussed as follows:

1. Analysis of Research Foundation (Q2 and Q4–Q8): As shown in Figure 4, 48.9% of graduate students did not participate in any research-related activities during their undergraduate studies. Additionally, 44.6% of respondents indicated that they were unclear about how to determine their research direction. For Q5 through Q8, since these items reflect the fundamental competencies required for conducting scientific research, students who rated themselves at level 3 or below were categorized as having a weak research foundation. The results reveal that the majority of students fall into this category, highlighting the necessity of incorporating relevant foundational research training into the curriculum.

Figure 4: Survey Results on Research Foundation

2. Analysis of Practical Foundation (Q3 and Q9–Q11): As shown in Figure 5, 72.3% of graduate students reported having no prior experience with real-world engineering projects or industry-related work. Additionally, 55.3% of respondents indicated that they were unable to independently build a complete project, reflecting a general lack of hands-on project experience. However, results from Question 9 show that 68.1% of the students are proficient in more than three programming languages, and findings from Question 10 indicate that 57.4% are familiar with version control tools. This suggests that while most students possess a certain level of technical competence, they lack opportunities to engage with real-world projects. In response, the college aims to strengthen university – enterprise collaboration, enabling graduate students to gain industry exposure and enhance their practical experience and project execution capabilities.

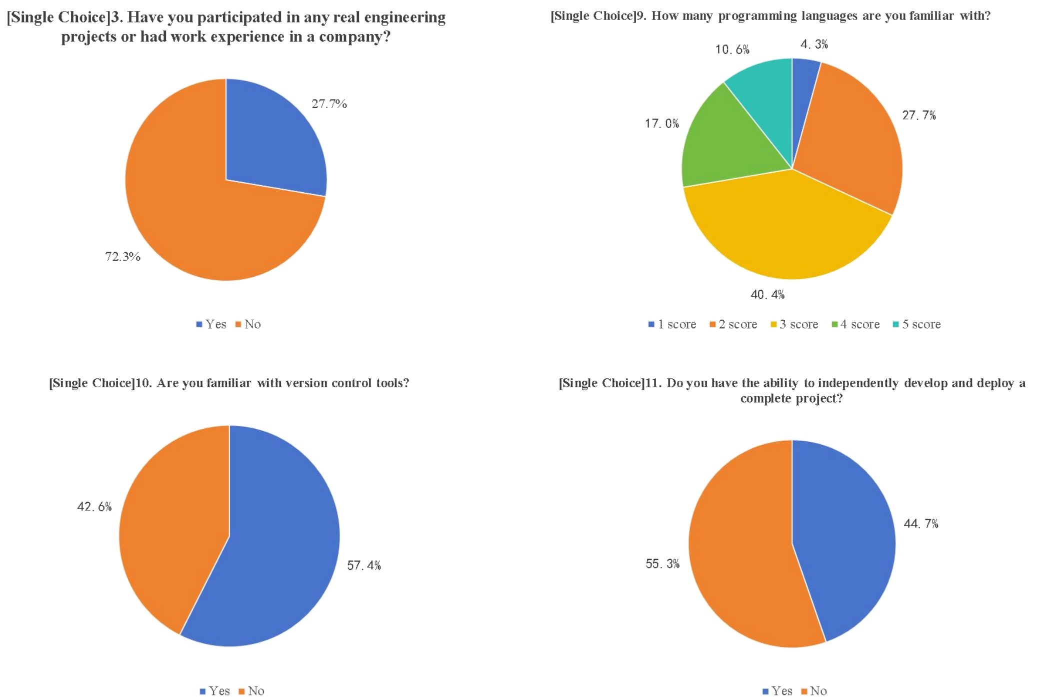

Figure 5: Survey Results on Practical Foundation

3. Analysis of Student Suggestions (Q12): As shown in Figure 6, the majority of students expressed support for the content included in the fourth level of the planning phase, namely "Foundational Content Instruction." Notably, 89.3% of students indicated that they hoped the university would strengthen training in the use of research and practical tools. This clearly reflects a high level of recognition and a positive attitude among graduate students toward this type of instructional content.

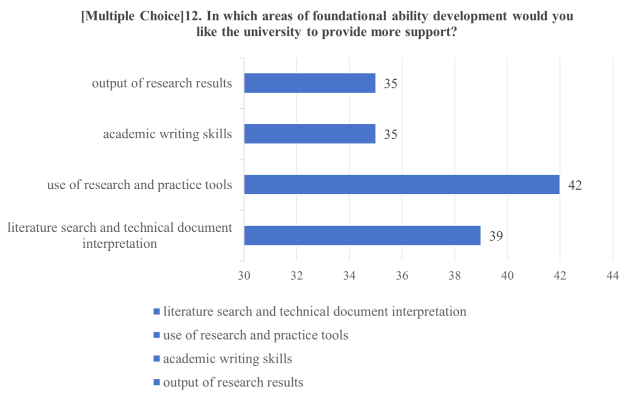

Figure 6: Survey Results on Student Suggestions

5 Teaching Effectiveness Analysis

As illustrated in Figure 7, since the implementation of the PCRE-based training model, the research capabilities of software engineering graduate students in the college have improved significantly. Graduates have achieved notable results in two key areas: academic paper publications and subject competitions. In contrast, there has been a slight decline in the number of patents and software copyrights over the past two years. This decline is primarily due to adjustments in the college's training approach — specifically, removing the mandatory requirement for publishing patents or software copyrights—thus allowing students to devote more time to scientific research and industry internships. Figures 8 and 9 present the number of academic papers published and the number of awards won in competitions by graduates from 2020 to 2025. It is worth noting that the data for 2020 and 2021 represent the period prior to the implementation of the reformed training model, Due to the fluctuations in the number of papers published by current students, it is difficult to accurately reflect the actual effectiveness of graduate education. Therefore, it is necessary to evaluate the quality of education more comprehensively by analyzing the achievements of graduates.

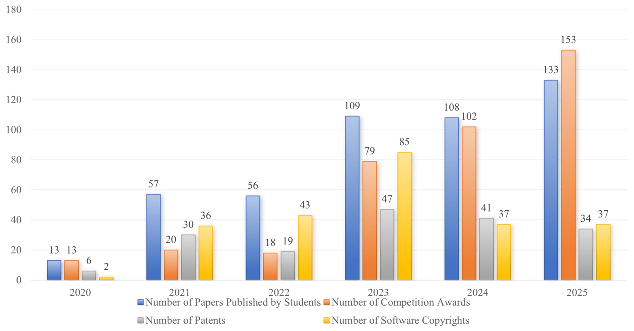

Figure 7: Research Output of Graduating Cohorts (2020-2025)

Regarding academic paper publications, as shown in Figure 8, graduates from the 2022 to 2025 cohorts have achieved remarkable results. During this period, graduate students published a total of 463 academic papers, including 166 papers indexed by SCI/EI (accounting for 35.8%) and 142 papers in CCF-recommended conferences and journals (accounting for 30.6%). The number of publications in 2025 increased by 137.6% compared to 2022, the year when the reform began, reaching 133 papers. Moreover, the proportion of high-level publications significantly increased, with SCI/EI and CCF papers together accounting for 92.4% (123 papers) of the total publications that year.

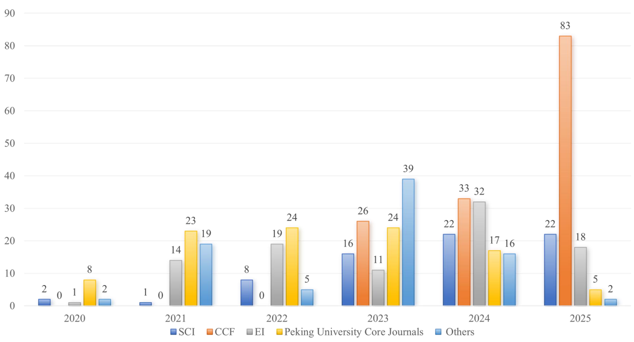

Figure 8: Academic Papers Published by Graduating Cohorts (2020-2025)

In terms of academic competitions, as shown in Figure 9, graduates from the classes of 2022 to 2025 have also performed outstandingly. The graduate students have won a total of 352 awards in various competitions, including 143 national-level awards (accounting for 40.6%), such as the National Second Prize in the China Graduate Mathematical Contest in Modeling and the National Second Prize in the China Robotics and Artificial Intelligence Competition. Compared with the period before the reform of the training model, the number of awards won by the 2025 graduates has achieved a breakthrough increase, strongly demonstrating the significant improvement in the students' practical application abilities.

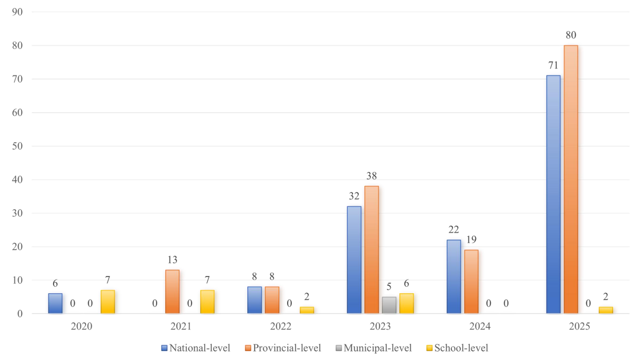

Figure 9: Awards-Winning Competition by Graduating Cohorts (2020-2025)

These achievements would not have been possible without the support of the proposed training model. In terms of cultivating graduate students' research abilities, the college has focused on building foundational research literacy, helping students quickly understand the research process, master literature review skills, and learn academic writing methods, thereby laying a solid foundation for producing high-quality papers. Meanwhile, regularly held academic paper activities have greatly broadened students' academic horizons; according to data, graduates of the 2025 class participated in over 20 academic seminars on average. Furthermore, to improve teaching quality and establish high-quality research workshops, the college has introduced more than thirty young PhD scholars in recent years and conducted multiple teaching ability training sessions for all faculty members, thereby enhancing the overall teaching level and providing strong support for the training model. With these combined efforts, students have been able to better present their research results in the form of high-quality papers. Regarding the cultivation of practical abilities, the school-enterprise cooperative training approach has effectively enhanced graduate students' engineering practice skills. At the same time, the involvement of enterprises has allowed students to access cutting-edge industry concepts and technologies, laying a solid foundation for achieving excellent results in academic competitions.

6 Conclusion

This study proposes an innovative PCER cycle-based training model to address key challenges in software engineering graduate education at regional universities. The model establishes a systematic framework for continuous improvement through four phases: (1) The implementation of a multi-level training ladder during the Planning phase, which clarifies competency milestones for both academic and professional master's students, ensuring structured progression aligned with their specialization tracks. (2) The development of a dynamic hierarchical case repository in the Carrying out phase, bridging the academia-industry gap by transforming faculty research and industrial projects into modular teaching materials, thereby alleviating resource constraints. (3) A multi-dimensional evaluation mechanism in the Evaluation phase, integrating quantitative metrics and systematic feedback to assess training outcomes objectively, with data-driven insights for iterative refinement. (4) The Refinement phase's closed-loop optimization, which enables timely adjustments to curriculum, resources, and mentorship strategies based on performance analytics.

The model demonstrates effectiveness in enhancing both research and practical competencies and provides regional universities with a scalable framework to improve graduate education quality in alignment with technological and industry developments. Future work will focus on enriching case studies, improving personalized learning pathways, and strengthening industry collaboration to further optimize the training system.

Acknowledgments

This research was supported by the following projects: the Provincial Teaching Reform Research Project of Higher Education Institutions in Hubei Province (No. 2024346), the National Natural Science Foundation of China (No. 62102291), the Graduate-Level Quality Course "Advanced Software Engineering" of Wuhan Textile University (2024), the Practical Platform Construction Project for Algorithm Design Courses (No. 231002405072709), the Industry-University Collaborative Education Program of the Ministry of Education (No. 220606008213849), and the University-Level Teaching Research Project of Wuhan Textile University titled "A Study on the Teaching Model of Software Engineering Courses Integrating Research Resources into Classroom Content". We gratefully acknowledge their support.

References

[1] Tan Dekun, Zhao Jia, Han Longzhe, et al. Research-Case-Driven Inquiry-Based Teaching Model for Introduction to Artificial Intelligence Course [J]. Computer Education, 2023(12): 356-360.

[2] Ma Hua, Liu Hong. Personalized Training of High-Quality Compound Software Engineering Talents [J]. Computer Education, 2025(1): 76-80.

[3] Song Tian, Huang Tianyu. New Thinking on Teaching Cases for Python Programming Language [J]. Computer Education, 2017(12): 11-1419.

[4] Li Chaoling, Wang Fuqiang, Liu Minghua. Reform and Practice of Computer Technology Basic Teaching Based on PDCA Cycle [J]. Computer Education, 2019(11): 108-111.

[5] Su Shuzhi, Fang Xianjin, Yang Gaoming. Cultivating Research Thinking in Computer Teaching [J]. Education Progress, 2019, 9(3): 208-212.

[6] Zhao Jinshuai, Zhu Yi, Feng Xuan, et al. Reform of Computer Course Teaching Model in Universities under the Background of Deep Integration of Industry and Education [J]. Computer Education, 2025(1): 50-54.

Bangchao Wang, male, Associate Professor, Ph.D., Master's supervisor. His research interests include intelligent software engineering and natural language processing. Email: wangbc@whu.edu.cn

Yuyang Dai, male, Master's student. His research interests include computer vision and deep learning. Email: 2415283041@wtu.edu.cn

Hongyan Wan, female, Lecturer, Ph.D., Master's supervisor. Her research interests include natural language processing, requirements tracing, and machine learning. Email: hywan@wtu.edu.cn

Xinrong Hu, female, Professor, Ph.D., Doctoral supervisor, and Dean of the School. She is a Distinguished Teaching Professor of Hubei Province. Her research interests include image processing and media computing, 3D computer graphics, computer vision, virtual reality, and machine learning. Email: hxr@wtu.edu.cn

Wei Xiong, male, Professor, Ph.D., and Master's supervisor. His research mainly focuses on parallel computing, edge computing, and software reliability, as well as their applications in areas such as the industrial Internet. Email: wx_wh@wtu.edu.cn