Article

Optimization of Course Content and Exploration of Teaching Pathways for the JavaEE Framework Technology Course Oriented Toward Innovative Talent Cultivation

Lihua Luo [1,*] and Qian Wu [1]

[1] Department of Computing, Guangdong University of Science and Technology, Dongguan, China
* Correspondence: Lihua Luo, Department of Computing, Guangdong University of Science and Technology, Dongguan, China

Abstract: With the continuous advancement of engineering education reform, the role of software engineering programs in higher education has evolved to shoulder the critical responsibility of cultivating high-level, innovation-driven professionals who can meet the demands of the rapidly changing digital era. Among the core practical courses offered in applied undergraduate programs, the JavaEE Framework Technology course plays a pivotal role in enhancing students' engineering literacy, hands-on development competence, and capacity for innovation. However, traditional teaching models often fall short in nurturing these competencies due to their limited focus on theoretical knowledge and framework usage. This study, grounded in the innovative ability structure model, proposes a systematic course content optimization strategy centered around a three-tier structure—foundation, application, and expansion. The redesigned curriculum aims to bridge the gap between theoretical learning and real-world application, fostering students' ability to undertake enterprise-level software development tasks. In addition, the paper introduces a diversified teaching pathway that integrates classroom instruction, project-based learning, and competition-oriented practice, thereby aligning the instructional approach with the principles of outcome-based education (OBE). By constructing a holistic teaching framework that emphasizes capability development and knowledge transfer, this research offers a practical and scalable reference model for the reform of software engineering curricula. It not only supports the development of students' technical proficiency and creative thinking but also contributes to the formation of comprehensive professional qualities required by modern engineering industries.

Keywords: JavaEE framework technology; innovative talent cultivation; course content optimization; teaching path design

Published: 06 August 2025

1. Introduction

With the accelerated advent of the digital economy era, the industrial landscape is undergoing profound transformation, placing increasingly higher demands on the quality and structure of talent cultivation in higher education. In particular, the expectations for software engineering graduates have shifted dramatically—from mere technical proficiency to the ability to innovate, solve complex problems, and adapt to rapidly evolving technologies [1]. This paradigm shift requires educational institutions to rethink and redesign their curriculum systems to foster innovative, application-oriented talents who can meet the dynamic needs of the software industry.

Modern engineering education advocates a comprehensive talent training model that emphasizes not only technical knowledge, but also engineering thinking, collaborative problem-solving, and the capacity for continuous learning and innovation [2]. In this context, the JavaEE Framework Technology course serves as a pivotal component within the

software engineering curriculum. It is not only a bridge between foundational programming and enterprise-level software development, but also a critical platform for cultivating students' project-based learning ability and innovation consciousness. The design of course content and teaching pathways in this subject directly influences students' transition from mastering basic syntax to acquiring full-stack development capabilities suitable for real-world enterprise systems [3].

However, the current instructional model for JavaEE Framework Technology remains largely traditional. It primarily emphasizes framework usage and API familiarity, while overlooking the cultivation of higher-order skills such as system integration, architectural thinking, and innovative application design. Furthermore, conventional teaching methods, dominated by lecture-based instruction and summative assessments, fail to create opportunities for students to engage in authentic, problem-driven learning experiences. The lack of interactive, task-based, and student-centered strategies has become a major bottleneck in nurturing innovative capabilities in software engineering students [2].

Addressing these challenges necessitates a systematic reform of both course content and pedagogy. This study responds to this educational imperative by proposing an optimized and restructured JavaEE Framework Technology course. Anchored in the goal of cultivating high-level innovative software professionals, the reform introduces a three-dimensional instructional pathway that integrates modular knowledge delivery, project-based application, and innovation-driven competition. This comprehensive teaching framework aims to realize the deep integration of knowledge acquisition, skill enhancement, and professional literacy development. It also aligns with the broader vision of engineering education reform, which emphasizes student-centered, outcome-based, and competence-oriented teaching models for the development of next-generation software engineers [3].

2. Teaching Challenges of the Java EE Framework Technology Course

With the continuous advancement of engineering practice and the evolving demands of the software industry, the teaching of the Java EE Framework Technology course in universities faces significant challenges that restrict its effectiveness in nurturing innovative talents. These challenges primarily revolve around two main aspects: limitations in course content and the inadequacies of the teaching methodology.

2.1. Limitations in Course Content and Innovation Guidance

The existing curriculum for the Java EE Framework Technology course primarily focuses on the basic application of Java EE frameworks and fundamental API operations. While this foundation is necessary for students to acquire technical skills, it falls short in cultivating deeper engineering innovation abilities. The lack of emphasis on advanced concepts such as architectural design, system integration, and innovative problem-solving limits students' capacity to develop creative and critical thinking skills required for complex software development. Moreover, the course content has not kept pace with the rapid evolution of software technologies. Despite the proliferation of emerging frameworks and tools, the curriculum remains anchored in traditional Java EE frameworks, resulting in a disconnect between academic learning and current industry practices. This gap hinders students from gaining exposure to cutting-edge technologies and best practices, thereby reducing their readiness for real-world software engineering challenges.

2.2. Inflexibility of Teaching Methods and Insufficient Support for Active Learning

Traditional teaching methods for this course largely rely on lecture-based instruction, often presented through static PowerPoint slides, and concluded with final examinations. This rigid approach provides limited opportunities for students to actively engage with the material or apply their knowledge in meaningful ways. The lack of project-driven learning, collaborative exercises, and interactive classroom activities restricts students'

ability to construct knowledge independently and develop problem-solving skills in practical contexts. Additionally, the absence of diversified teaching pathways and support mechanisms inhibits personalized learning and fails to accommodate varying student needs and learning paces. Consequently, students may struggle to internalize theoretical knowledge and translate it into practical competencies essential for innovative software development.

To address these challenges, there is an urgent need to comprehensively optimize the course content, diversify teaching methods, and establish a supportive learning environment. Such reform should integrate up-to-date technological content, promote project-based and student-centered teaching approaches, and foster a dynamic platform where theoretical knowledge and practical skills are seamlessly connected. This will better prepare students to meet the demands of the digital economy and cultivate the innovative talents that the modern software industry requires.

3. Building a Capability Model for Innovative Talents

In the ongoing transformation of engineering education across colleges and universities worldwide, cultivating innovative talents has become the paramount objective driving curriculum reform and pedagogical innovation. The traditional educational paradigm, which primarily focuses on knowledge transmission and skill training, is increasingly insufficient to meet the complex and rapidly evolving demands of the modern software industry. Consequently, educational institutions are called upon to develop competency-based models that not only emphasize foundational knowledge but also foster creativity, critical thinking, problem-solving abilities, and practical engineering skills that are essential for innovation and sustained professional growth.

As a core course within the software engineering curriculum, particularly oriented toward enterprise application development, the Java EE Framework Technology course must be strategically aligned with the logic of student ability development. This necessitates a clear definition of the capability dimensions and growth trajectories that the course intends to support, ensuring that learning outcomes extend beyond basic programming skills to encompass higher-order competencies such as system architecture design, software integration, security management, and project collaboration. The course thus serves as a vital platform for students to transition from mastering theoretical constructs to applying integrated knowledge in complex, real-world development environments.

Building on a comprehensive review of international and national frameworks for innovative talent cultivation—such as the CDIO (Conceive-Design-Implement-Operate) initiative and industry-driven competency models—along with specific job competency requirements in the software engineering field, this study constructs a three-dimensional competency framework (Figure 1). This structural model integrates core technical skills, practical engineering experience, and cognitive innovation capabilities, emphasizing the cultivation of interdisciplinary talents who combine solid technical foundations with adaptive thinking, creativity, and collaborative problem-solving skills. These three dimensions are mutually reinforcing and form the backbone of the innovative talent development process.

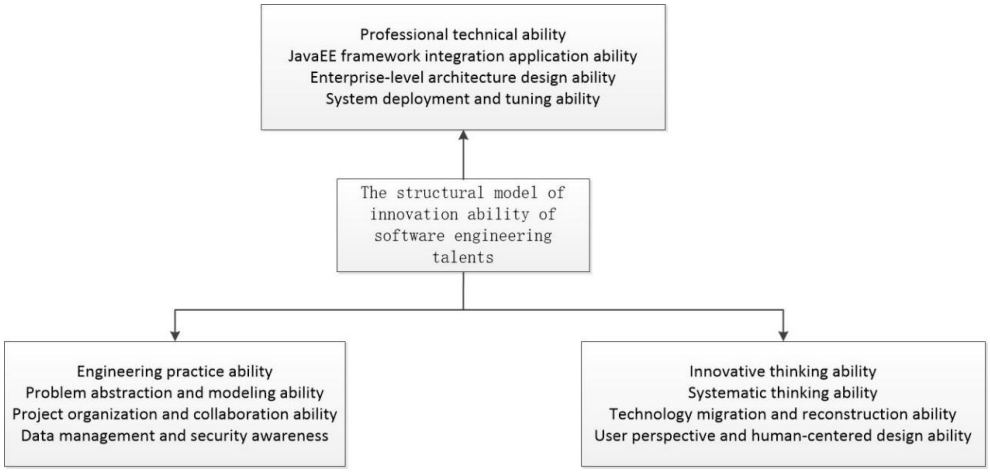

Figure 1. Structural model of innovation capability of software engineering talents.

By adopting this capability model, the Java EE Framework Technology course is positioned not only as a conduit for technical knowledge but also as a catalyst for cultivating comprehensive engineering competencies. The course design facilitates students' ability to analyze complex system requirements, implement scalable software solutions, and adapt flexibly to emerging technologies. In this way, the course plays a critical role in bridging the gap between academic learning and professional practice, equipping students with the competencies required to excel in dynamic enterprise environments and contribute meaningfully to technological innovation [4].

4. Optimization Design of Course Content System

In order to effectively implement the Outcome-Based Education (OBE) concept and enhance the overall effectiveness of the JavaEE Framework Technology course in cultivating high-level innovative talents, the course content system has undergone a systematic and comprehensive optimization. This reform centers on constructing a three-tiered teaching framework, which is organized around the progressive development stages of foundation, application, and extension. This layered structure serves as the main axis to strategically align the JavaEE Framework Technology curriculum with the overarching goals of innovative talent cultivation within the OBE educational paradigm [5].

By integrating the course's pre-requisite knowledge components with its subsequent advanced learning modules, the JavaEE Framework Technology course occupies a pivotal position within the broader competency development pathway of software engineering students. It acts as a critical transitional platform bridging foundational programming skills and advanced system-level engineering capabilities. Accordingly, this study proposes an optimized course content system that highlights the interconnection and progressive deepening of knowledge and skills through the core dimensions of foundation, application, and expansion, as illustrated in Figure 2. This framework not only structures the curriculum content in a coherent and logical manner but also facilitates the targeted development of students' innovative abilities and practical competencies in a stepwise fashion.

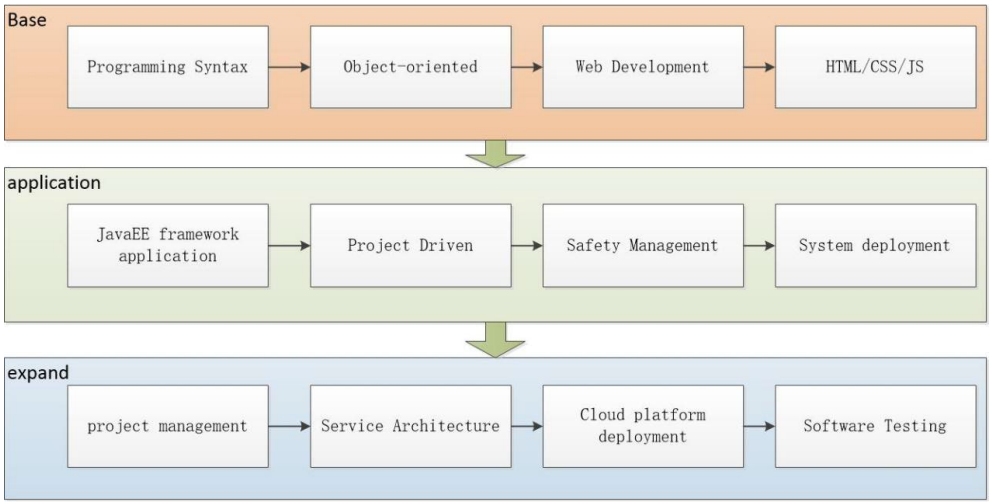

Figure 2. Course content optimization system diagram of JavaEE framework technology.

In the basic module, the course focuses on consolidating students' technical foundation, covering programming syntax, object-oriented programming, Web development, and front-end technologies such as HTML/CSS/JavaScript, helping students master the basic methods and thinking patterns of development. Through practical links such as basic grammar exercises and small functional module development, students can form a relatively solid programming ability and lay a good foundation for subsequent course learning.

In the application module, as a core link between the upper and lower levels, it focuses on the integration and project practice of the mainstream JavaEE framework, including the practical application of technologies such as Spring, Spring MVC, and My Batis. At this stage, the course guides students to complete real or simulated project development through a project-driven approach and further enhances their understanding and mastery of system architecture, security management, permission control, and deployment processes. This module emphasizes the combination of theory and practice to enhance students' comprehensive development and system implementation capabilities.

In the expansion module, the course further enhances students' engineering thinking and professional ability, covering project management methods, service architecture design, cloud platform deployment, and software testing. By simulating enterprise-level development processes, students can be trained to collaborate and adapt in complex engineering environments, and their comprehensive professional literacy and technology transfer capabilities can be strengthened.

On the basis of course content optimization, several implementation strategies are proposed: First, focus on the systematic connection between modules, and use unified project tasks to run through the three-layer course structure to achieve the coherent integration of knowledge points and skill points. Secondly, introduce a multi-dimensional evaluation mechanism with learning outcomes as the core, combine process evaluation with outcome evaluation, and comprehensively measure students' achievement in knowledge mastery, ability improvement and comprehensive literacy; thirdly, promote the digitalization and modularization of course resources, and use the resource system of "micro-course + task + case + project" to facilitate the implementation of hybrid teaching and flipped classrooms. Finally, by utilizing competency profiling and learning trajectory analysis, the course provides students with personalized learning paths and development suggestions to stimulate their autonomous learning potential.

By constructing a course content optimization system with foundation-application-extension as the main line, not only the coordinated cultivation of knowledge, ability and literacy is achieved, but also the student-centered and outcome-oriented teaching goals

under the OBE concept are met. This optimized design will transform the JavaEE Framework Technology course from a traditional teaching model into a modern education model based on competency, and help cultivate high-quality, innovation-driven, application-oriented professionals.

5. Exploration and Practice of Multiple Teaching Paths

In response to the modularization and ability-oriented optimization of the JavaEE Framework Technology course content, the design of diverse and integrated teaching pathways has become a crucial factor in achieving course objectives and enhancing instructional effectiveness. Guided by student-centered and competency-based educational principles, the curriculum reform has transitioned from a traditional, lecture-centric approach to a multi-faceted teaching model. This model systematically combines classroom instruction, project-based learning, and competitive practice, forming a robust threefold teaching practice system that fosters comprehensive student development.

5.1. Classroom Teaching Path: Modular and Task-Driven Learning

The classroom teaching path emphasizes transforming traditional instruction by adopting modularized content delivery paired with task-driven pedagogies. Educators employ micro-lectures, detailed knowledge explanations, and interactive real-time coding sessions to guide students through the architectural principles and development patterns underpinning the JavaEE core framework. To deepen comprehension, students engage with framework source code analysis and hands-on case studies that simulate real development scenarios. Innovative teaching strategies such as flipped classrooms and collaborative group work are integrated to stimulate active participation and enhance classroom dynamics. The curriculum is meticulously organized according to the three-tier structure of foundation, application, and expansion, enabling students to systematically build a coherent knowledge base while progressively developing essential technical skills.

5.2. Project Practice Path: Progressive and Scalable Development Experience

The project practice path is designed to provide students with authentic, multi-phase development experiences that bridge theory and practice. By assigning expandable project tasks aligned with the application and expansion modules, students are encouraged to undertake end-to-end software development processes. This includes stages such as system architecture design, database modeling, functional module implementation, and deployment. The staged approach not only cultivates technical competence but also reinforces project management skills, teamwork, and iterative problem-solving abilities. Through continuous feedback and instructor guidance, students develop a holistic understanding of software lifecycle management, preparing them for the complexities of enterprise-level system development.

5.3. Innovation Competition Path: Enhancing Learning Through External Challenges

Recognizing the value of competitive environments in stimulating creativity and problem-solving skills, the course actively promotes student participation in a variety of professional competitions both within and beyond the university. Competitions such as the Blue Bridge Cup, the College Student Innovation and Entrepreneurship Competition, and the China College Student Computer Design Competition serve as important platforms for applying theoretical knowledge to practical challenges. Through these events, students are motivated to develop software projects that combine technical depth with real-world application value. The teaching team plays a supportive role by providing expert guidance, technical consultation, and resource assistance throughout competition preparation, thereby enhancing students' understanding and flexible application of the JavaEE framework. This pathway not only enriches the learning experience but also cultivates students' innovation capacity and solution-oriented mindset.

6. Conclusion

This study systematically designed and implemented a comprehensive teaching reform for the JavaEE Framework Technology course under the guiding principles of innovation and competency-based education. A modular content system was proposed, structured around a three-layer progressive framework—foundation, application, and extension—coupled with a multi-path teaching model integrating classroom instruction, project practice, and competitive activities. This three-in-one practice path effectively aligns the course objectives with broader graduation requirements, ensuring a coherent and targeted educational experience.

Through the modularization and capability-oriented reconstruction of the course content, the reform has achieved a systematic optimization and gradual advancement in the teaching material. Beginning with the cultivation of fundamental programming skills, the curriculum progressively advances toward complex project development and engineering-level expansions, thereby enabling students to construct a comprehensive learning trajectory that bridges theoretical knowledge mastery and practical skill transfer. The diversified and integrated teaching pathways significantly enhance student engagement and foster active learning and application, cultivating not only technical proficiency but also innovation-driven problem-solving capabilities.

Furthermore, the deep integration of teaching content, pedagogical methods, and evaluation mechanisms realized through the classroom-project-competition multi-path approach promotes a fundamental shift from traditional knowledge transmission to competency development. This transformation reorients the learning process from a teacher-led model to a student-centered framework, encouraging autonomous learning and collaboration. As a result, the curriculum reform not only advances students' technical abilities and engineering practice skills but also supports the holistic development of their overall professional competence and innovative thinking. This lays a robust foundation for nurturing high-quality, application-oriented software engineering talents capable of meeting the evolving demands of the digital economy and industry innovation.

Funding: The work was supported by the 2024 Guangdong Undergraduate College Teaching Quality and Teaching Reform Project No. 1112 (Teaching Reform and Practice of JavaEE Framework Technology Course Based on OBE under Industry-Education Integration) and the Quality Engineering Project of Guangdong University of Science and Technology (GKZLGC2024149).

References

1. T. Kong, L. Feng, and Q. Guo, "Innovative research on digital talent cultivation mode of higher vocational innovation and entrepreneurship education promoted by industry-teaching integration," *Appl. Math. Nonlinear Sci.*, vol. 10, no. 1, 2025, doi: 10.2478/amns-2025-0044.
2. X. Xiao, Y. Zhu, Z. Guo, et al., "An innovation talent cultivation mechanism for robotics in the digital-intelligent era: Exploration and practice at Wuhan University," *Front. Digit. Educ.*, vol. 2, no. 1, p. 12, 2025, doi: 10.1007/s44366-025-0048-9.
3. G. Wu and Q. Zhang, "Integration of science and education: Reform and practice of innovative talent cultivation model in the field of communication engineering," *Eng. Educ. Rev.*, vol. 2, no. 2, pp. 72–76, 2024, doi: 10.54844/eer.2024.0619.
4. X. Meng, C. Xu, and X. Sun, "Research and practice on innovative talent cultivation model in the field of educational technology from the perspective of education artificial intelligence," *Adv. Voc. Tech. Educ.*, vol. 6, no. 3, pp. 7–14, 2024, doi: 10.23977/avte.2024.060302.
5. J. Liu and J. Wu, "The development logic and dilemma resolution of innovative talent cultivation models in universities from the perspective of localization," *J. Humanit. Arts Soc. Sci.*, vol. 9, no. 4, 2025, doi: 10.26855/jhass.2025.04.015.