*International Journal of Basic and Applied Sciences*, 14 (6) (2025) 237-241

International Journal of Basic and Applied Sciences

*Website: www.sciencepubco.com/index.php/IJBAS*
*https://doi.org/10.14419/15dr4033*
***Research paper***

# Exploring The Innovative Path of The Management Mechanism of Applied Talent Training in Computer Science in Colleges and Universities Based on The Integration of Industry and Education

**Haitao Fan [1] *, Dr. Rowena R. Abrea [2]**

[1] *College of Teacher Education, Batangas State University, the National Engineering University, Batangas, Philippines*
[2] *Doctoral Institution: College of Teacher Education, Batangas State University, the National Engineering University, Batangas, Philippines*
**Corresponding author E-mail: rowena.abrea@g.batstate-u.edu.ph*

*Received: July 30, 2025, Accepted: October 4, 2025, Published: October 12, 2025*

**Abstract**

This research explores the innovative pathways for the management mechanisms of applied talent training in computer science in universities, based on the integration of industry and education. By employing a combination of literature review, questionnaires, interviews, case studies, data analysis, field observations, policy analysis, and expert consultation, this study provides a comprehensive analysis of the status and challenges of applied talent training in computer science. The findings highlight the importance of aligning university curriculum with industry needs, strengthening practical teaching components, and enhancing university-enterprise cooperation through effective management mechanisms. The research proposes specific strategies to optimize the organizational management of university-enterprise cooperation, improve collaborative teaching models, and establish robust interest coordination and incentive mechanisms. These strategies aim to foster a more effective and sustainable model for applied talent training in computer science, ensuring that graduates are better prepared to meet the demands of the modern technology industry. The study concludes that a well-integrated industry-education model can significantly enhance the quality of applied talent training and contribute to the development of the computer science field.

***Keywords***: *Industry-Education Integration; Applied Talent Training; Management Mechanisms; Computer Science; University-Enterprise Cooperation*

## 1. Introduction

In the era of comprehensive digital economy penetration and explosive growth in computing power demands, computer science professionals have become the core variable driving industrial upgrading. However, while artificial intelligence, cloud-native technologies, and large models are rapidly iterating in industrial applications, universities maintain a four-year cycle for curriculum revisions. Enterprises crave engineers ready for immediate deployment, yet academic evaluations still rely on thesis submissions and written tests. This triple misalignment of "technological iteration, training pace, and job standards" has exacerbated the supply-demand gap for applied computer talents. More critically, traditional industry-academia collaborations often remain at the level of equipment donations or one-time lectures, lacking deep integration with cutting-edge scenarios like open-source ecosystems, DevOps pipelines, and cloud-native security, as well as dynamic adjustment mechanisms for courses and assessments driven by real-time data. Thus, industry-education integration is no longer about resource accumulation but requires systematic restructuring around governance frameworks, curriculum DNA, faculty expertise, practical scenarios, and quality closed-loop systems. Against this backdrop, this paper explores: When the half-life of computer technology has shortened to twelve months, how can universities and enterprises jointly build a management mechanism with sensitive "technology radar," pure engineering culture, and rigorous evaluation metrics? This would enable the running of iterative curriculum updates with monitored pilots, staged rollouts, and reversions like production releases. By analyzing the innovative paths of five dimensions: school-enterprise collaborative governance, micro-course modules, two-way faculty–industry exchange, cloud-native practice base, and multi-data closed-loop, this paper tries to present a new paradigm of computer application-oriented talent training that can not only respond to the immediate needs of the industry, but also adhere to the long-term value of education.

## 2. Literature Review

Over the past four years, the scholarly conversation around nurturing industry-ready computer-science graduates has converged on a single, urgent insight: when large-language models, cloud-native stacks, and zero-trust frameworks iterate faster than academic calendars, the traditional four-year frozen curriculum becomes a liability rather than a safeguard. Researchers now argue that the only viable response is to adopt open-source governance and cloud-native engineering practices in curriculum management. Zhang, Zhou & Luo (2022) first articulated a “depth–breadth–novelty” competency cube, insisting that students must simultaneously master low-level system depth, cross-domain breadth, and frontier-shattering novelty. Su et al. (2023) corroborated this model with GitHub telemetry, showing that merged commits to CNCF (open-source cloud-native software under the Cloud Native Computing Foundation used as de facto industry standards in modern infrastructure) projects predict starting salary better than GPA ever could. Li & Wu (2021) replaced static syllabi with a micro-service-style “course-package plus hot-swappable micro-modules” architecture that shadows the quarterly CNCF Landscape releases; Dong et al. (2025) demonstrated a 27 % boost in creative-problem-solving scores when large-language-model APIs were injected into an otherwise conventional C++ course. Huawei Cloud’s joint experiments with several universities have gone further, turning syllabus changes into GitOps pull requests (managing course and lab changes via version-controlled repositories and peer review, analogous to software release workflows) reviewed by both faculty and corporate committers; the median time from RFC to classroom deployment now stands at 21 days. Governance has likewise been re-engineered: a university-industry technical steering committee modeled on the Apache PMC gives corporate committers veto power over capstone designs, while faculty Slack channels stream real-time Prometheus metrics (time-series telemetry from student-deployed services used to monitor availability, latency, and error budgets for instructional feedback) from student-run micro-services—triggering just-in-time remediation workshops whenever SLAs breach their error budgets. The 2025 National Computer Basic Education Summit crystallized these threads into the notion of a “generative-AI talent flywheel,” in which large models continuously mine job-skill heat-maps to steer the next micro-module refresh. Yet significant gaps remain: dynamic governance metrics, longitudinal alumni telemetry, and ethical oversight of AI-accelerated curricula have yet to be systemically addressed—precisely the territory this study occupies.

Global models and international partnerships

International exemplars further demonstrate durable pathways:

MIT’s long-standing industry-partnership labs link capstone design to sponsor-defined constraints and shared IP frameworks, compressing the loop from requirements to deployable prototypes and aligning assessments with production-grade deliverables.

Germany’s dual education system provides a co-op cadence in which learners alternate between academic blocks and paid enterprise placements under joint curricular governance, with recent technology-sector variants illustrating how dual-mentoring and co-assessment can accelerate role-ready competency acquisition in software and embedded systems.

Recent case work such as Siemens & TU Munich (2023) highlights dual education adaptations for tech roles, including synchronized competency maps and co-developed assessment tasks that mirror enterprise toolchains, offering transferable design patterns for university–enterprise co-governance in computing programs.

Ethical governance in AI-accelerated curricula

As curricula increasingly employ large-language models and AI-driven analytics, ethical oversight must address bias in content recommendations, unequal tooling access, model hallucination risks in assessment contexts, and data privacy in telemetry-informed interventions. Governance mechanisms can include an ethics review board spanning faculty, student, and enterprise representation; mandatory bias and safety audits for AI-assisted courseware; model cards and data sheets for educational models; and red-teaming exercises for prompt and output reliability in graded activities.

Embedding required modules on AI ethics, fairness, accountability, transparency, security, and privacy—aligned to role pathways—helps ensure that agility does not erode educational integrity or equity outcomes.

## 3. Method

This research employs a combination of literature review, questionnaires, interviews, case studies, data analysis, field observations, policy analysis, and expert consultation. The literature review helps to outline the research background and theoretical foundations. Questionnaires and interviews are used to gather first-hand data from universities, enterprises, and students. Case studies provide an in-depth analysis of university-enterprise cooperation models and their effectiveness. Field observations verify the actual conditions of practical teaching components. Policy analysis interprets the policy support for industry-education integration. Expert consultation ensures the scientific validity and practical applicability of the research. These methods complement each other, ensuring the comprehensiveness and reliability of the research findings and providing robust support for the innovation of talent training mechanisms in computer science at universities.

## 4. Results and Discussion

### 4.1 Investigation of the Current Status of Applied Talent Training in Computer Science in Universities

Firstly, a comprehensive analysis of the curriculum and teaching content in computer science programs is conducted. The curriculum structure forms the foundation of talent training, and a well-designed curriculum can provide students with a systematic knowledge framework. Through surveys conducted across multiple universities, it is observed that there is a certain variation in the proportion of theoretical and practical courses. For instance, in some universities, theoretical courses account for as high as 80% of the total, while practical courses only make up 20%. This ratio discrepancy may lead to students having a solid grasp of theoretical knowledge but relatively weaker practical skills. Further analysis of the course module distribution reveals that the allocation of hours for basic courses, core professional courses, and cutting-edge technology courses also varies. For example, in a certain university, basic courses account for 30%, core professional courses for 40%, and cutting-edge technology courses for 30%. Whether this module distribution is reasonable needs to be assessed in conjunction with industry demands.

**Root causes of persistent misalignment**

- The persistence of curriculum–industry misalignment stems from institutional inertia in curriculum governance cycles, funding constraints that slow lab and equipment refresh, and incentive systems that prioritize publications over practical alignment with evolving job standards.
- Accreditation and committee-based revision processes frequently operate on 3–4-year cadences, which lag the 12–18-month half-life of key technologies, further widening the timing gap between academic syllabi and production tooling changes.

- Faculty workload models and procurement policies also constrain rapid adoption of new stacks, making it difficult to pilot and scale modules tied to current industry pipelines without earmarked budgets and streamlined approvals.

**Integrating emerging trend lines**

- To reduce this lag, emerging threads such as AI ethics, responsible AI engineering, secure-by-design practices, and foundational quantum computing concepts can be embedded as micro-modules aligned to role-specific skill maps in Table 1, ensuring agility without destabilizing the core sequence."
- "For example, programs can prioritize model evaluation, prompt engineering, data governance, and MLOps literacy for software development engineers; advanced statistics, causal inference, and model interpretability for data analysts; and secure model pipelines, adversarial robustness, and privacy-preserving analytics for cybersecurity engineers, mapping each to Table 1's high-frequency skill dimensions.

Meanwhile, the alignment between teaching content and enterprise needs is also a crucial factor affecting the quality of talent training. Through surveys conducted in the computer industry, it is found that there is a significant difference in skill demands across different positions. For example, software development engineers require a high level of proficiency in programming languages (such as Python and Java) and development tools, while data analysts place more emphasis on data analysis tools and data visualization skills (see Table 1). Program directors can use Table 1 as a prioritization instrument to allocate contact hours and assessment weight to the highest-frequency skills by role, while reserving 10–15 percent of hours for fast-refresh micro-modules on frontier topics such as AI ethics and quantum-aware algorithms where appropriate. However, there is a disconnection between the course content of some universities and the needs of enterprises. For example, in artificial intelligence courses, universities may focus more on theoretical algorithm explanations, while enterprises are more concerned with the practical application and optimization of models. This disconnection may lead to students being unable to meet the demands of enterprises for applied talents after graduation.

**Table 1:** Frequency of Skill Demands for Different Positions in the Computer Industry

| Skill Dimension | Software Development Engineer | Data Analyst | Cybersecurity Engineer |
|---|---|---|---|
| **Programming Languages** | High (90%) | Medium (70%) | Medium (60%) |
| **Development Tools** | High (80%) | Low (30%) | Medium (50%) |
| **Data Analysis Tools** | Low (20%) | High (90%) | Low (10%) |
| **Project Management Skills** | Medium (50%) | High (80%) | Medium (40%) |

The low demand for data analysis tools among cybersecurity engineers reflects role specialization: security operations, incident response, identity and access management, and secure architecture emphasize threat modeling, network telemetry, vulnerability management, and compliance controls over BI-style analytics stacks, explaining the 'Low 10' entry for data analysis tools in that column. Curricular implications include emphasizing secure coding, network defense, identity systems, cryptography, and cloud security posture management for cybersecurity tracks, with only targeted analytics coverage (e.g., log parsing, SIEM queries) rather than broad business analytics training, thereby freeing capacity to increase hands-on labs in red–blue team exercises and incident response drills. Secondly, practical teaching is an essential component for cultivating students' practical and innovative abilities. The arrangement of experimental and training courses directly affects students' practical operation skills. For example, in a certain university, the proportion of experimental hours accounts for 35% of the total hours, and there is a relatively large number of experimental projects. However, the update frequency of experimental equipment is relatively low (see Table 2). In such cases, students may encounter problems such as outdated equipment and software versions during experiments, which can affect the experimental outcomes. The construction of off-campus internship bases is also an important part of practical teaching. Through surveys conducted in some universities, it is found that the industry distribution of internship bases is quite extensive, but the scale and quality of the bases vary greatly. For example, a certain university has established internship bases with many Internet companies, but some of these bases are relatively small in scale and offer limited internship positions (see Table 2). In addition, the application of project-driven teaching models is also worth paying attention to. Some universities cultivate students' practical and innovative abilities through project courses, but there are differences in project completion rates and student participation. For example, in a certain university, the project completion rate is 80%, and students have a high level of satisfaction with the project-driven teaching model (see Table 2).

**Table 2:** Data on Practical Teaching in a University

| Aspect of Practical Teaching | Data Indicator | Data Value |
|---|---|---|
| Proportion of Experimental Hours | Percentage of total hours | 35% |
| Number of Experimental Projects | Total number of projects | 50 |
| Update Frequency of Experimental Equipment | Update cycle (years) | 3 years |
| Industry Distribution of Internship Bases | Internet: 50% | - |
| Scale of Internship Bases | Average number of employees | 200 |
| Project Completion Rate in Project-driven Teaching Model | Project completion rate | 80% |
| Student Satisfaction with Project-driven Teaching Model | Satisfaction percentage | 85% |

**Implications of a 3-year equipment cycle**

- A 3-year update cadence raises the risk that student lab environments lag enterprise baselines by 1–2 major versions, reducing transferability of operational skills and increasing onboarding time post-graduation.
- Mitigations include containerized lab stacks pinned to supported LTS images, cloud credits for managed services, and a rolling update policy for mission-critical tooling each semester, preserving stability while exposing learners to current versions and security patches.

**Project completion and outcomes**

- An 80 percent project completion rate coupled with 85 percent satisfaction suggests high engagement, but triangulation with external performance (e.g., code review rubrics, SLA adherence in service labs) is recommended to ensure depth and production realism rather than form-only completion.

The construction of a teaching staff is the key to improving teaching quality. The practical experience and industrial background of teachers directly affect the applicability and relevance of teaching content. Through surveys conducted in some universities, it is found that the proportion of teachers with industrial work experience is relatively low, at only 30% (see Table 3). This may lead to teachers lacking real-world cases and project experience in the teaching process, which can affect students' understanding and application of knowledge. At the same time, the involvement of enterprise mentors should not be overlooked. Some universities have hired enterprise mentors to participate in teaching, but there are differences in the number of enterprise mentors and the allocation of teaching tasks. For example, a certain university has hired 10 enterprise mentors, who mainly undertake practical guidance and project guidance tasks (see Table 3). The participation of enterprise mentors can provide students with real industrial experience and project practice opportunities, but how to ensure the teaching effectiveness of enterprise mentors and student satisfaction is still a problem that needs to be solved.

**Table 3:** Data on Faculty Development in a University

| Indicator of Faculty Development | Data Value |
|---|---|
| Proportion of Teachers with Industrial Work Experience | 30% |
| Average Duration of Teachers' Industrial Work Experience | 5 years |
| Number of Enterprise Mentors | 10 |
| Allocation of Teaching Tasks for Enterprise Mentors | Practical guidance: 60% |

Finally, the evaluation of talent training quality is an important means to measure the effectiveness of talent training. The scientific and rational nature of the evaluation index system directly affects the accuracy and reliability of the evaluation results. Through surveys conducted in some universities, it is found that there are differences in the weight allocation of evaluation indicators such as professional knowledge, practical ability, innovation ability, and professional quality. For example, in a certain university, the weight allocation of evaluation indicators is as follows: professional knowledge 40%, practical ability 30%, innovation ability 20%, and professional quality 10%. Whether this weight allocation is reasonable needs to be assessed in conjunction with enterprise demands and students' actual performance. Meanwhile, the employment quality and career development of students are also important indicators for evaluating the quality of talent training. Through analysis of the employment data of graduates from some universities, it is found that there are significant differences in employment rate, employment unit nature, job positions, and salary treatment among graduates from different universities. For example, in a certain university, the employment rate of graduates is 90%, and most of them are employed in the Internet industry, with job positions mainly concentrated in software development and data analysis, and an average salary of 8,000 yuan. By tracking the career development of graduates, the quality of talent training in universities can be further assessed.

**Curriculum prioritization guide (by role)**

- Software development: emphasize programming languages, development tools, and collaborative engineering practices; reserve micro-modules for AI-assisted coding and secure SDLC.
- Data analyst: emphasize statistics, data analysis tools, visualization, and data ethics; add MLOps awareness for production handoffs.
- Cybersecurity: emphasize secure architecture, identity, cloud security, incident response, and adversarial testing; provide targeted analytics for SIEM and log analysis.

## 4.2 Exploration of Innovative Pathways for Management Mechanisms under Industry-Education Integration

Under the industry-education integration model, the organizational management mechanism of university-enterprise cooperation is the key to ensuring the smooth progress of cooperation. The establishment of a university-enterprise cooperation management committee can effectively coordinate the cooperation affairs between the two parties. The composition of the management committee should include university leaders, enterprise executives, professional teachers, and enterprise technicians to ensure that the interests of all parties are fully considered. For example, in the management committee established through cooperation between a certain university and an enterprise, university leaders account for 30%, enterprise executives for 30%, professional teachers for 20%, and enterprise technicians for 20%.By clarifying the working system and process of the management committee, decision-making efficiency can be improved to ensure the smooth implementation of cooperation projects. For example, the management committee holds a meeting once a month to discuss the progress of cooperation projects and solutions to problems.

The establishment of a joint office or project team can further strengthen the synergy of university-enterprise cooperation. The personnel configuration of the joint office or project team should include relevant personnel from both the university and the enterprise to ensure close cooperation between the two parties. For example, in the project team jointly established by a certain university and an enterprise, university personnel account for 40%, and enterprise personnel account for 60%. By clarifying the working responsibilities and process of the joint office or project team, the efficiency of project implementation can be improved. For example, the project team is responsible for project planning, organization and implementation, process management, and outcome evaluation.

Moreover, it is crucial to clarify the responsibilities and division of labor between the university and the enterprise in the talent training process. A well-defined distribution of responsibilities can prevent issues such as unclear responsibilities and shirking of duties during cooperation. For example, the university is responsible for formulating the teaching plan, organizing classroom teaching, and providing teaching venues, while the enterprise is responsible for providing practical projects, arranging enterprise mentors, and offering internship bases. Through agreements, the rights and obligations of both parties in talent training, technology research and development, and resource sharing should be clearly stipulated to ensure smooth cooperation. For example, a university-enterprise cooperation agreement can specify the specific terms of cooperation in these areas to avoid misunderstandings and disputes.

In addition, the collaborative management mechanism in the talent training process is an important aspect of innovation. The university and the enterprise should work together to develop a curriculum system that meets the needs of the industry. For example, regular seminars involving experts and teachers from both the university and the enterprise can be organized to adjust the curriculum and update teaching content based on the development trends of the computer industry and the demands of enterprises. Joint development of textbooks by university teachers and enterprise technicians can also be explored to incorporate real-world cases and project experience from enterprises into the teaching materials, enhancing their practicality and relevance.

The collaborative management of practical teaching is another key area. For example, enterprises can provide students with real project practice opportunities, and university teachers and enterprise mentors can jointly guide students in completing projects. This collaborative approach can effectively cultivate students' practical and innovative abilities. Furthermore, the university and the enterprise can jointly build practical teaching bases. For example, the university can provide venues and equipment, while the enterprise can invest funds and technical support to create high-level practical teaching bases that meet the needs of students' practical training.

The collaborative development of the faculty is also essential. For example, the university can arrange for teachers to undertake internships in enterprises, and the enterprise can send technical personnel to serve as part-time teachers in the university. This two-way exchange can enhance teachers' practical abilities and the teaching level of enterprises. Joint faculty training programs can also be explored. For example, regular training activities can be organized, inviting enterprise experts and university professors to teach, covering aspects such as computer frontier technologies, teaching methods, and project management to improve the overall quality of teachers.

Furthermore, the establishment of an interest coordination and incentive mechanism is vital. To encourage enterprises to participate in talent training, mechanisms for compensating enterprises for their involvement should be explored. For example, the government can introduce tax preferential policies to provide tax exemptions for enterprises participating in university-enterprise cooperation. Universities can also set up special funds to offer financial support to enterprises involved in talent training. At the same time, enterprises should be made aware of the potential benefits they can gain from participating in talent training. For example, enterprises can select outstanding talents in advance through participation in talent training, reducing recruitment costs. They can also leverage the research capabilities of universities to conduct technology research and development, and innovation.

Incentive measures for universities and teachers should also be considered. For example, universities can use performance appraisal and title promotion to motivate teachers to actively participate in university-enterprise cooperation. The outcomes of teachers' participation in university-enterprise cooperation projects can be included in performance appraisal indicators, and teachers who perform outstandingly in university-enterprise cooperation can be given priority in title promotion. External incentive mechanisms to attract universities to actively engage in industry-education integration can also be explored. For example, the government can establish a selection activity for industry-education integration demonstration universities, offering financial rewards and policy support to selected universities to encourage them to actively carry out industry-education integration work.

Finally, the establishment of an incentive and guidance mechanism for students is also important. For example, scholarships and honor certificates can be used to motivate students to actively participate in university-enterprise cooperation projects. A special scholarship for university-enterprise cooperation can be set up to reward students who perform outstandingly in projects. Honor certificates can also be awarded to students participating in projects to enhance their sense of achievement and honor. Guidance on career planning and employment prospects analysis can also be provided to encourage students to actively participate in university-enterprise cooperation. For example, universities can offer career planning courses to help students understand the development trends of the computer industry and guide them in making rational career plans.

## 5. Conclusion

This research concludes that the integration of industry and education offers a transformative pathway for enhancing the management mechanisms of applied talent training in computer science at universities. The alignment of university curricula with industry needs, the strengthening of practical teaching components, and the optimization of university-enterprise cooperation are identified as critical factors for success. Effective management mechanisms, including clear organizational structures, collaborative teaching models, and robust interest coordination and incentive mechanisms, are essential for fostering a sustainable and high-quality talent training environment. The study emphasizes that a well-coordinated industry-education model not only improves the employability and practical skills of graduates but also drives innovation and development within the computer science field. Future research should focus on further refining these mechanisms and exploring new models of collaboration to adapt to the rapidly evolving technological landscape.

## References

[1] Zhang, J., Zhou, Y., & Luo, X. (2022). Reconstructing the competency framework for applied computer-science talent in the era of cloud-native computing. Journal of Higher Engineering Education, 44(3), 15-28.
[2] Su, L., Chen, H., & Wang, M. (2023). Upstream contribution volume as a predictor of graduate employability: Evidence from CNCF projects. Computers & Education, 198, 104-118.
[3] Li,W., & Wu, Q. (2021). Micro-modularity for curriculum agility: A GitOps-inspired approach to computer-science education. Chinese Journal of Distance Education, 39(6), 27-35.
[4] Dong, Y., Liu, S., & He, J. (2025). Integrating LLM APIs into programming courses: Effects on creative problem-solving. ACM Transactions on Computing Education, 25(2), 1-20.
[5] Xu, R., Pan, Z., & Lin, T. (2025). Embedding SLA literacy and chaos engineering into national programme standards. Software Engineering Review, 42(1), 55-68.
[6] Huawei Cloud Education Lab. (2023). GitOps-driven syllabus pipeline: A case study on university-industry co-governance. Internal White Paper.
[7] Liu, C., & Xu, J. (2023). Telemetry-triggered remediation workshops: An observability approach to talent flywheels. Proceedings of the 7th International Conference on Computer Science Education Innovation, 112-119.
[8] National Computer Basic Education Summit. (2025). Generative-AI talent flywheel: Synthesis report. Beijing: Higher Education Press.