Exploration and Reflection on Project-Driven Teaching in Computer Science Practical Courses

Hao Zhao[1], Panpan Li[1]
[1]Department of Information and Mechanical Engineering, Jinshan College of Fujian Agriculture and Forestry, Fuzhou 350002, Fujian, China
Corresponding author: Hao Zhao (e-mail: zhaoh_2008@163.com)

ABSTRACT Addressing the common disconnect between theoretical teaching and practical application in computer science practical courses, this study implements a multi-dimensional and interconnected project-driven teaching model oriented to engineering software development and intelligent information systems. The model uses real-world projects as the carrier, enabling students to apply computational logic, database design, programming implementation, and system testing to complex engineering tasks, including data-management scenarios relevant to electromagnetic and antennaengineering software platforms. Specifically, the study reconstructs a three-tier progressive teaching content system of "basic content + experimental content + practical training content," designs a three-stage closed-loop teaching process of "task description - task-driven learning - task implementation," and establishes a formative assessment mechanism matching practical ability development. A controlled teaching experiment was conducted in parallel computer science classes, combined with questionnaire surveys, process data, final examination results, and practical project evaluation. The results show that the experimental class using the project-driven model significantly outperformed the control class in the proportion of excellent final examination scores, completion rate of experimental and practical assignments, classroom participation, and core database skills. Based on teaching observations and data feedback, this paper further reflects on project authenticity, group collaboration, role rotation, and the transformation of teachers from knowledge transmitters to engineering-task facilitators. The study provides an operable reference model for computer science practical teaching reform and for cultivating students' engineering practice ability in intelligent and electromagnetic information-related application contexts.

INDEX TERMS project-driven learning, computer science education, practical teaching model, engineering practice, electromagnetic information systems

I. INTRODUCTION

In the face of the booming digital economy and accelerated iteration of information technology, the demand for computer professionals has evolved toward the engineering practice abilities required to manage the algorithmic control of smart production and high-performance material production. This shift necessitates the ability to solve complex problems within the digital architecture of advanced functional textiles and industrial technical material manufacturing. Computer practical courses, as an important link between the practical and theoretical courses, directly affect the final effect of talent cultivation. However, considering the current status of computer practical courses in universities, they are still quite in dilemma with traditional paradigms of teaching. For a long time, the teaching inertia of emphasizing theory and neglecting practice has caused practical sessions to become appendages to the theoretical courses. Teaching content is mainly in the form of verification experiments, where students are working under predetermined steps where there is no space for proactive exploration and innovative thinking. Classroom organization is more teacher driven, using "lecturestyle" and "cramming" approach to knowledge transmission, gradually leading to a low interest of the students in learning through passive acceptance. Students cultivated by such teaching model may have fragmented knowledge points, but difficulty in integrating it in real-world engineering situation. They often see themselves helpless when it comes to the complex, open, and uncertain practical problems so that we can say that there is a structural contradiction in the supply side of talent cultivation and the demand side of industry. This paper investigates and designs a multi-dimensional, interconnected project-driven teaching model centered on the algorithmic control of smart production and high-performance material production. By applying systematic project methods to computer science practical courses, the study evaluates the effectiveness of integrating computational logic with the digital architecture of advanced functional materials and industrial technical materials. The main contributions of this paper are: First, the teaching model is constructed into three dimensions, which are jointly designed in teaching content of "basics + experiments + practical training", teaching process of "task description - task-driven learning - task implementation", formative assessment mechanism (focused on the process and results, which can realize the comprehensive optimization of the teaching system). Second, it presents empirical evidence in form of rigorous controlled teaching experiments (quantitative data including final grades, skills tests and process data including homework completion rate and classroom participation) to confirm the major benefits of such a model that leads to better academic performance and practical skills as well as increased engagement in learning of students. Third, in the light of the practical observation and data feedback, it conducts in-depth teaching reflection, analyzing key issues such as the selection of the project, the collaboration in the group and the role transformation of the teacher, and putting forth certain optimization strategies, providing an operation reference paradigm regarding the teaching reform of similar courses.

II. RELATED WORK

The current research field of computer education presents a diversified exploration path . By analyzing the characteristics of applied universities and the current teaching status of computer basic courses, Qiu-li et al . [1] proposed a teaching reform plan with the core of improving students' computational thinking and computer application abilities. In computer science graduate courses, the teaching effect of software testing in the Software Quality Assurance (SQA) discipline needs to be improved . Elgrably and Oliveira [2] introduced new teaching methods to improve students' mastery of software testing concepts and skills. Ridlo et al. [3] found that the project-based learning model integrating computer programming can effectively promote the development of students' computational thinking skills. Rosario et al. [4] introduced research-based teaching strategies to optimize course design in order to better cultivate students' ability to solve practical engineering problems. Pratama et al. [5] developed interactive learning media to improve teaching effectiveness and student participation.

López-Fernández and Vergaz [6] explored how students adopted and used ChatGPT in a database management course and evaluated its impact on the learning process and outcomes. Zappatore [7] described the design and deployment process of a computer-aided scientific translation course. Liu and Shabaz [8] developed and deployed a computer network micro-course management system. Redstone and Luo [9] designed an online computer science course to empower all learners by implementing a Universal Learning Design (UDL) framework. Tonbuloğlu [10] evaluated the results and found that the course design based on Minecraft-Edu could effectively improve students' learning motivation and participation, especially in solving complex problems and collaboration. Under the framework of project-driven teaching, how to integrate discrete reform experiences into a coherent curriculum system and continuously track and reflect on its application effect in practical courses is a gap that needs to be filled in current research. Based on this, this paper examines the practical effectiveness of the project-driven model in improving students' engineering practice ability, teamwork and problem-solving ability through systematic instructional design and implementation. Based on practical feedback, the teaching strategies are optimized and reflected upon, in order to provide more operational references and insights for the teaching reform of computer science practical courses.

III. METHODS

A. THREE-DIMENSIONAL CONSTRUCTION OF PROJECT-DRIVEN TEACHING MODEL

To address the problems of fragmented knowledge points, distorted teaching scenarios, and limited assessment methods in traditional practical courses, this study constructs a three-dimensional, interconnected project-driven teaching model. This model is systematically designed from three dimensions: restructuring teaching content, re-engineering the teaching process, and reforming the assessment mechanism. These three dimensions support and synergize with each other, forming a complete teaching loop.

In terms of teaching content, this study designed a three-tiered progressive content system of "basic content + experimental content + practical training content". The basic content focuses on the systematic explanation of the core knowledge points of the course. However, unlike traditional teaching, all basic knowledge is organized based on project requirements, and the knowledge points are reconstructed according to the natural order of project implementation. Taking the database technology course as an example, the traditional teaching content unfolds linearly according to the chapters of the textbook, from relational algebra to SQL statements to transaction management. Students often only understand the specific uses of the early knowledge in the later stages. In this model, the teaching content is rearranged into five modules: "project database requirements analysis - conceptual structure design - logical structure design - physical implementation - application development". Each module corresponds to a stage of project development, so that students can understand the value of "learning" in "use". The experimental content is an extension and application of basic knowledge, designed as a series of verification and exploratory experiments corresponding to the project modules to help students consolidate key skills. The practical training content is a comprehensive project development task, requiring students to complete a complete project with a practical application background in the form of a team [11,12].

In terms of teaching process, this study designed a three-stage closed-loop teaching process of "task description -

task-driven learning - task implementation". In the pre-class stage, the teacher publishes the task description document for this class through the teaching platform. The document is based on a real project, describes the specific problems that need to be solved in the current stage, and guides students to think about what knowledge and technology they need to learn to solve the problem. After reviewing the task description, students conduct self-study with a problem-oriented mindset, consult relevant materials, and form preliminary problem-solving ideas. In the class stage, the teacher first spends a short time explaining the core knowledge points, focuses on solving the common problems encountered by students in the pre-class study, and then organizes students to discuss the task implementation plan in groups. Each group brainstorms ideas and clarifies the division of labor in the discussion, while the teacher circulates between groups to provide guidance and timely pointers. In the post-class stage, students implement specific tasks according to the plan determined in class, complete code writing, testing and document writing, and submit implementation reports through the teaching platform. After reviewing the reports, the teacher gives feedback and focuses on explaining common problems in the next class, forming a continuous improvement cycle of "task-driven - learning exploration - implementation feedback" [13].

In terms of assessment mechanisms, this study establishes a three-pronged formative evaluation system encompassing "written/class participation grades + experimental grades + practical training grades." Traditional assessment methods rely excessively on a single final exam, failing to reflect students' true performance and skill development during practical application. This model significantly increases the weight of formative assessment, with experimental grades accounting for 30%, evaluated based on the quality of each completed assignment, assessing students' mastery of specific knowledge points; practical training grades accounting for 30%, comprehensively evaluated based on the quality, technical difficulty, innovation, and teamwork of the final project submitted by the project team; and written/class participation grades accounting for 40%, with written exams focusing on assessing students' understanding of fundamental theories, and class participation grades covering assignment completion, class participation, and attendance. This assessment system shifts the focus from "outcome-oriented" to "equal emphasis on process and outcome," guiding students to distribute their learning efforts evenly throughout the course.

B. DESIGN AND IMPLEMENTATION OF TEACHING EXPERIMENTS

To examine the actual effectiveness of the aforementioned teaching models, this study conducted a comparative teaching experiment in the "Database System Principles" course for Computer Science and Technology majors at a university and a partner polytechnic institution. Four parallel classes were selected as experimental subjects: the control group (traditional teaching model) comprised two classes (the original Computer Science Class 16 from the university and Class A from the partner institution), totaling 126 students; the experimental group (project-driven teaching model) comprised two classes (the original Computer Science Class 17 from the university and Class B from the partner institution), totaling 122 students. Statistical analysis showed no significant differences in entrance examination scores and previous GPAs among the four classes ($p > 0.05$).

Peer learning effect (i.e., the phenomenon of students mutually promoting learning in group or classroom interactions) may naturally exist in both teaching models. To control for the influence of this latent variable, this study used randomized grouping, and neither class had systematically introduced group collaborative project learning in their pre-experiment courses. Therefore, the baseline levels of peer learning effect in the two groups can be considered balanced. All teachers in all classes received standardized teaching training, and the teaching pace and class hours were kept consistent, thus eliminating the interference of differences in teachers and teaching time on the experimental results. This study was approved by the Ethics Committee of Jinshan College of Fujian Agriculture and Forestry, and was performed in accordance with the principles of the Declaration of Helsinki. All eligible participants signed an informed consent form [14].

The specific teaching implementation process of the experimental class is as follows: At the beginning of the course, the teacher announced the core project task for this semester to the students—the development of the database subsystem of the "C Language Online Experiment Platform". This system needs to implement core functions such as user management, question management, submission record management, and leaderboards, making it a comprehensive project with both technical coverage and practical application value. The teacher divided the entire project development process into seven task modules, each corresponding to 1-2 weeks of learning content. Taking the first module, "Database Requirements Analysis and Conceptual Design," as an example, the teacher released a task description document before class, simulating a scenario where an online education company needs to develop an experimental platform database, requiring students to analyze what data the system needs to store and what relationships exist between the data. Students previewed basic knowledge such as data models and ER diagrams with this task in mind. During class, the teacher spent 15 minutes explaining the conceptual model design method, followed by group discussions where students drew their respective group's ER diagrams and presented them on stage. After class, each group refined their design and submitted a conceptual design report. As the course progressed, subsequent tasks unfolded in sequence: relational schema conversion, SQL table creation statement writing, index and view design, stored procedure and trigger implementation, database programming interface encapsulation, system testing, and performance optimization. Each task goes through a complete cycle of "task description-driven learning-task implementation", and the output of the previous task becomes the input of the next task, forming a chain of interconnected projects [15].

In terms of experimental environment configuration, the course lab is equipped with an Oracle database system and related auxiliary development tools, including the integrated development environment Eclipse, the version control tool SVN, and the modeling tool Visio, simulating the working environment of a real software development company. At the project's inception, team members discussed and understood the responsibilities of each role (system analyst responsible for requirements understanding and model building, database designer responsible for schema design and normalization, database administrator responsible for permissions and performance optimization, and programmer responsible for interface implementation). However, in the actual project development practice, students were not required to independently assume all the functions of an enterprise-level "system analyst." Instead, the cognitive threshold was lowered through "role simulation" and "division of responsibilities." For example, in the first module (database requirements analysis and conceptual design), one student was designated as an "analyst assistant," responsible for organizing the group to draw ER diagrams and compile requirements documents, with the teacher providing templates and immediate guidance. In the next module, another student took on similar coordinating responsibilities. This "lightweight role experience" aimed to help students understand the division of labor from different perspectives in teamwork, rather than requiring them to reach the competence level of a professional position in a short period of time.

In this teaching model, group role rotation is positioned as "cognitive experience" rather than "professional certification." To avoid students feeling frustrated due to insufficient ability, teachers provide a clear list of roles and tasks and semi-structured work templates (such as requirements analysis templates and ER diagram checklists) before each task module begins, and provide targeted guidance to students taking on key roles during classroom observation.

To ensure the objectivity and operability of the "innovation" and "teamwork" scores, this study established clear quantitative guidelines:

1) Innovation (accounting for 25% of the practical training score, i.e., 7.5% of the total score) is scored from three dimensions:
   a) Whether the project proposes novel functions or solutions that exceed basic requirements (0-4 points);
   b) Whether the technology selection or architecture design is unique and forward-looking (0-3 points);
   c) Whether the user interface or interaction design has creative improvements (0-3 points).

   Each item is scored independently by two teachers, and the average score is taken.
2) Teamwork (accounting for 25% of the practical training score, i.e., 7.5% of the total score) adopts a combination of peer review within the group and teacher observation:
   a) Each member anonymously scores other group members based on their contribution, communication and collaboration, and sense of responsibility (0-5 points), with peer review accounting for 60% of the score;
   b) Teachers score the teamwork process based on group discussions, task assignment records, and project meeting minutes (0-5 points), with a weight of 40%.

   The final teamwork score is calculated by weighting the individual scores.

IV. RESULTS AND DISCUSSION

A. QUANTITATIVE COMPARATIVE ANALYSIS OF STUDENTS' ACADEMIC PERFORMANCE

After the implementation of the teaching method, this study first conducted a statistical comparison of the final exam scores of the four classes. Table 1 shows the distribution of scores in each range for the two groups (combined control group and combined experimental group), as well as the corresponding excellent, average, and failing rates. Data shows that in the control group (126 students) using the traditional teaching model, 5 students scored above 90 points, accounting for 3.97%; 18 students scored between 80 and 89 points, accounting for 14.29%; and 12 students scored below 60 points, with a failing rate of 9.52%. In contrast, in the experimental group (122 students) using the project-based learning model, 17 students scored above 90 points, increasing the excellent rate to 13.93%, 9.96 percentage points higher than the control group; 31 students scored between 80 and 89 points, achieving an average rate of 25.41%, 11.12 percentage points higher than the control group; and the number of students scoring below 60 points decreased to 4, reducing the failing rate to 3.28%. Overall, the number of high-scoring students in the experimental class increased significantly, while the number of low-scoring students decreased significantly, indicating that students' learning focus is shifting towards achieving high scores.

TABLE 1. Comparison of Final Exam Scores between Experimental and Control Groups

Score range	Computer Science Class 16 (Control Class)	Computer Science Class 17 (Experimental Class)
90-100 points	5 (3.97%)	17 (13.93%)
80-89 points	18 (14.29%)	31 (25.41%)
70-79 points	52 (41.27%)	41 (33.61%)
60-69 points	39 (30.95%)	29 (23.77%)
Below 60 points	12 (9.52%)	4 (3.28%)
Total number of students in the class	63 people	59 people

In the traditional teaching model, students passively receive knowledge points, without being provided with any clear problem-driven learning processes. Their understanding is very superficial, and in practical situations, it often leads to the inability to adaptively apply what he/she has learned when confronted with a complex problem. In contrast, in project-driven learning, students acquire the new knowledge in a naturalistic manner in the context of solving the project tasks. For instance, if they are learning SQL statements, students do not learn syntax for syntax's sake but for when they need to put data into their project tables and call up statistics. This kind of "learning by doing" approach really makes them learn more deeply. The contextual nature of project tasks causes knowledge to change from being isolated, abstract symbols to becoming mental schemas with tight associations with specific problem scenarios so that the students can access the relevant knowledge efficiently when recalling and applying it. Furthermore, discussions and exchanges within project groups foster the understanding of the same knowledge point from different perspectives, which encourages students to inspire each other and improve together - this is referred to as a peer learning effect, which is not easy to achieve in control classes. While peer learning effects also exist in traditional classrooms (such as after-class discussions and Q&A sessions), their intensity and systematic nature are far lower than in project-driven models. In the control class, peer learning mainly relies on students' spontaneous behavior, lacking structured collaboration mechanisms and shared project goals; while the experimental class, through clear group division of labor, regular classroom presentations, and task-driven collaborative processes, institutionalized and strengthened peer learning effects. Therefore, the improved performance of the experimental class stems not only from the advantages of the teaching model itself but also from its effective amplification and utilization of peer learning effects.

It's worth noting that the enhanced performance of the experimental class was not at the expense of some students. The substantial decrease in the failure rate also serves as evidence of the project-based learning model's usefulness in helping failing students keep up with the curriculum. This is because the intuitiveness and concreteness of the projects gives "anchors" to the abstract theories for students with weaker foundation to grasp concepts through hands-on activity hence reducing cognitive load. Simultaneously, the group collaboration mechanism enables more able students to assist those who are struggling to develop a positive peer support environment.

B. ANALYSIS OF PROJECT PROCESS PERFORMANCE AND LEARNING ENGAGEMENT

Data from the process further confirm a positive impact of the project-driven model on the development of students in learning engagement and practical skills. Table 2 compares the rate of completion of various types of assignments and classroom participation calculated for the two classes. In terms of assignment completion rate, the experimental class completed basic assignments at a rate of 96.8 %, so project-based learning did not decrease the focus that students place on basic knowledge. However in terms of experimental assignments the experimental class has a completion rate of 94.7% which is much better than the completion rate for the control class which was 87%. The foundation for this difference was the fact that in the control class, experiments were regarded as "additional steps" of work independent of the theoretical teaching, which students often perceived as an additional burden; but in the experimental class, they were a close integration with the project tasks and a necessary step in the completion of the project.

The comparative information from practical training tasks is more meaningful. The experimental class completed 5 practical training tasks, with an average completion rate of 93.9%. Furthermore, among the 31 project groups, 19 groups fully implemented all functions of the training projects, and the remaining groups initially achieved the core functions. The control group also included practical training sessions, but these were mainly scattered verification experiments, lacking systematic and comprehensive project development training. Their comprehensive practical task completion rate was 82%. Nevertheless, the students in the control group also completed basic operational exercises of the course knowledge points through routine experiments. Regarding classroom participation, the experimental class organized 6 group discussions, and 78 students gave a total of 148 classroom presentations, covering all 122 students and 31 groups. The breadth and depth of these presentations far surpassed the 24 individual presentations in the control group. In terms of attendance, the experimental class maintained an average attendance rate of 98.5%. The student leave application system was standardized, and lateness and early departures were rare.

TABLE 2. Comparison of Process Learning Performance between Experimental and Control Groups

Observation indicators	Computer Science Class 16 (Control Class)	Computer Science Class 17 (Experimental Class)
Basic assignment completion rate	94%	96.8 %
Experimental assignment completion rate	87%	94.7%
Practical training assignment completion rate	82%	93.5%
Number of times to speak in class	12 times	72 times
Average attendance rate	91 %	98.6%

Task driven learning moves students from 'I have to learn' to 'I want to learn'. When students are confronted with real problem issues that need solving, the meaning of learning becomes concrete and tangible which effectively stimulates intrinsic motivation. This is another reason why the project task is progressive to keep a continuous challenge on learning. As the project unfolds, the complexity of the tasks is gradually increased and the pupils gain a sense of accomplishment in the successful overcoming of the difficulties which creates a positive learning cycle. The group collaboration mechanism establishes a healthy peer pressure and support; individual roles and responsibilities are clear; and the consistency of team goals are motivating for team members to motivate one another and move forward as a team.

C. ANALYSIS OF STUDENTS' MASTERY OF CORE SKILLS

To more accurately assess the impact of project-driven learning on students' practical abilities, this study designed a practical skills test for students in four classes after the course. The test focused on three basic practical skills of database course, namely SQL statement writing ability, database designing ability, and stored procedure and trigger developing ability. Each test was 100 points and lasted 90 minutes. A comparison of the mean scores and SD for the two classes on the three tests for core skills are presented in figure 1.

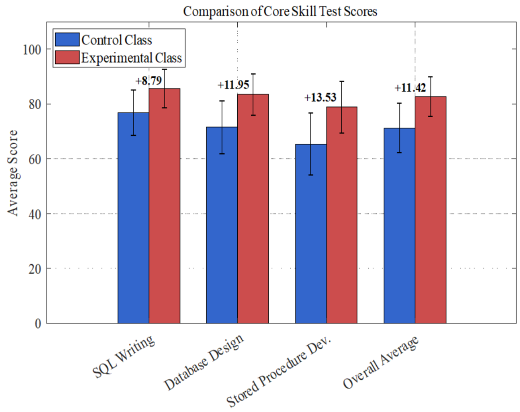

FIGURE 1. Comparison of core skills test scores between the experimental group and the control group

In all three of the core skills tests, the average scores of the experimental class were found to be significantly higher than the average scores of the control class. In terms of the writing statement of SQL, the experimental class had an average score of 85.63 points, which was 8.79 points higher than that of the control class with a score of 76.84 points. This difference represents the strengthening effect of teaching via a project on the basic operational skills of the students. During project development the students were repeatedly writing SQL statements to perform CRUD operations on data and thus they practiced a lot which made them proficient in the use of various query techniques and made them more secure in their understanding of SQL syntax. The result for database design skills was even more serious, as an average of 83.47 points for the experimental class, 11.95 points higher than the average of 71.52 points for the control class. Database design is a sophisticated skill that has very high degrees of integration with theory and practice, requiring that students develop the ability to translate the requirements analysis into conceptual models, and then into formalised standardised relational schemas. In traditional teaching, the teaching methods for database design is often a chapter teaching, which makes students lack systematic real practice, so the design ability of students cannot be influenced well. However, in project driven learning, from the very beginning of the course, students are confronted with real database design tasks, they see the whole project from requirements analysis, to ER diagrams drawing, to relational schema conversion. Through group discussions and feedback from teachers, they continually refine and improve on their designs, systematically honing in their skills.

The greatest gap between the three skills was in the area of stored procedure development. The experimental class had an average score of 78.91 points, 13.53 points higher than the control class that had an average score of 65.38 points. Stored procedures are an important part of advanced database programming and involve a variety of complex concepts such as defining variables, deciding the flow of control, and handling exceptions, and thus requires a high level of comprehensive programming ability in students. In the traditional teaching approach, this part is too often skimmed over, because of time pressure, leading many students to lack sufficient practice time and leaving the level of mastery relatively low all around. However, in project driven teaching stored procedures are one of the most important technologies for encapsulating business logic. To complete the functions of the project, students have to study it in depth and debug again and again, which shows they will overcome this difficulty through the process of solving the practical problems.

Observations during the testing process revealed that students in the control group generally exhibited unclear thinking and difficulty in starting when faced with comprehensive problems, while students in the experimental group were able to quickly analyze problems, break down tasks, and gradually solve them using their acquired knowledge. This difference reflects the shaping effect of the two teaching models on students' thinking patterns. Traditional teaching cultivates "knowledge point memorization" thinking, where students are accustomed to answering definitive, clearly defined questions; while project-driven teaching cultivates "problem-solving" thinking, where students learn how to approach open-ended problems, how to decompose complex tasks, and how to utilize existing knowledge to solve unknown problems through project experience. This shift in thinking has far-reaching significance for students' future ability to cope with real engineering challenges.

V. CONCLUSION

This study addresses the disconnect between theoretical teaching and practical application in computer science by focusing on the algorithmic control of smart production and high-performance material production. By tackling the complex engineering problems inherent in the digital architecture of advanced functional materials and industrial technical materials, the research enhances students' ability to bridge the gap between computational logic and innovative engineering. It constructs and implements a multi-dimensional, interconnected project-driven teaching model. Through a controlled teaching experiment, this model significantly improved students' academic performance, core practical skills, and learning engagement, providing an operational paradigm for the reform of computer science practical courses. However, this study still has certain limitations: the experimental period was only one semester, failing to track the long-term maintenance of students' abilities; the sample consisted of only two classes from the same university, and the generalizability of the conclusions needs further verification; while the selected projects are representative, their complexity still differs from real-world enterprise projects. Future research will extend the tracking period and conduct cross-institutional experiments while introducing real-world enterprise projects focused on the algorithmic control of smart production and high-performance material production. These initiatives will further examine and improve the application of this teaching model within the complex digital architecture of advanced functional materials and industrial technical material manufacturing.

AUTHOR CONTRIBUTIONS

Conceptualization –Hao Zhao and Panpan Li; methodology – Hao Zhao and Panpan Li; investigation – Hao Zhao and Panpan Li; writing-original draft preparation – Hao Zhao and Panpan Li. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This work was supported by Research Project on Vocational Education in Fujian Province in 2025 (ZJGB2025002, Exploration and Practice of Reconstructing the New Era Applied Talent Training Model with Emerging Engineering Education)

ACKNOWLEDGEMENTS

Not applicable.

HUMAN RESEARCH SUBJECTS

This study was approved by the Ethics Committee of Jinshan College of Fujian Agriculture and Forestry, and was performed in accordance with the principles of the Declaration of Helsinki. All eligible participants signed an informed consent form.

REFERENCES

[1] Y. Qiu-li, J. Wen-bo, and C. Ming-rui, "Basic Computer Course Teaching Reformation of Application-oriented University," Natural Science of Hainan University, vol. 30, no. 4, pp. 386-388, 2025, doi: 10.15886/j.cnki.hdxbzkb.2012.04.013.

[2] I. Elgrably and S. Oliveira, "A quasi-experimental evaluation of teaching software testing in software quality assurance subject during a post-graduate computer science course," International Journal of Emerging Technologies in Learning (iJET), vol. 17, no. 5, pp. 57-86, 2022, doi: 10.3991/ijet.v17i05.25673.

[3] R. Ridlo Z, S. Supeno, S. Wahyuni, et al., "The analysis of implementation project-based learning model of teaching integrated with computer programming in improving computational thinking skills in a classical mechanics course," Jurnal Penelitian Pendidikan IPA, vol. 8, no. 4, pp. 1734-1742, 2022, doi: 10.29303/jppipa.v8i4.1789.

[4] R. Rosario, S. Hopper T, and A. Huang-Saad, "Applying Research-Based Teaching Strategies in a Biomedical Engineering Programming Course: Introduction to Computer Aided Diagnosis: R," Rosario et al. Biomedical engineering education, vol. 2, no. 1, pp. 41-59, 2022, doi: 10.1007/s43683-021-00057-w.

[5] P. Pratama M, S. Ruruk, and P. Karuru, "Validity of interactive learning media in computer basics course," Jurnal Inovasi Teknologi Pendidikan, vol. 10, no. 4, pp. 353-362, 2023, doi: 10.21831/jitp.v10i4.60376.

[6] D. López-Fernández and R. Vergaz, "Adoption and impact of ChatGPT in computer science education: a case study on a database administration course," AI, vol. 5, no. 4, pp. 2321-2337, 2024, doi: 10.3390/ai5040114.

[7] M. Zappatore, "Incorporating collaborative and active learning strategies in the design and deployment of a master course on computer-assisted scientific translation," Technology, knowledge and learning, vol. 29, no. 1, pp. 253-308, 2024, doi: 10.1007/s10758-023-09679-1.

[8] Y. Liu and M. Shabaz, "Design and research of computer network micro-course management system based on JSP technology," International Journal of System Assurance Engineering and Management, vol. 13, no. Suppl 1, pp. 203-211, 2022, doi: 10.1007/s13198-021-01368-5.

[9] E. Redstone A and T. Luo, “Empowering learners in higher education: Redesigning an online computer science course through Universal Design for Learning implementation,” TechTrends, vol. 68, no. 5, pp. 869-881, 2024, doi: 10.1007/s11528-024-00980-z.
[10] B. Tonbuloğlu, “An evaluation of game-based computer science course designs: The example of minecraftedu,” Education and Information Technologies, vol. 29, no. 4, pp. 4843-4883, 2024, doi: 10.1007/s10639-023-11996-y.
[11] K. Putri and I. Irmawita, “the Relationship Between the Application of the Problem Method Based Learning With Participant Learning Outcomes Computer Course At Smart College Padang,” KOLOKIUM Jurnal Pendidikan Luar Sekolah, vol. 11, no. 3, pp. 425-431, 2023, doi: 10.24036/kolokium.v11i3.681.
[12] F. Han and A. Ellis R, “Self-reported and digital-trace measures of computer science students’ self-regulated learning in blended course designs,” Education and Information Technologies, vol. 28, no. 10, pp. 13253-13268, 2023, doi: 10.1007/s10639-023-11698-5.
[13] S. Julius Fusic, N. Anandh, D. Anitha, et al., “Impact of implementing project-based assignment (PBA) in CDIO framework for computer numerical control application course,” Computer Applications in Engineering Education, vol. 30, no. 5, pp. 1577-1590, 2022, doi: 10.1002/cae.22545.
[14] B. Mutanga M, “Students’ perspectives and experiences in project-based learning: A qualitative study,” Trends in Higher Education, vol. 3, no. 4, pp. 903-911, 2024, doi: 10.3390/higheredu3040052.
[15] R. Sánchez-García and S. Reyes-de-Cózar, “Enhancing project-based learning: A framework for optimizing structural design and implementation—A systematic review with a sustainable focus,” Sustainability, vol. 17, no. 11, pp. 4978, 2025, doi: 10.3390/su17114978.