Academic Education Publishing House

A Study on the Influencing Factors of Learning Outcomes in Software Engineering Courses

Mingjin Shen, Xiaowen Li
School of Computer and Artificial Intelligence, Henan Finance University, Zhengzhou, China

Abstract: The research explores the ambiguous determinants and insufficient adaptation of technology in the educational results of software engineering courses. Drawing from cognitive load theory and social constructivism, a tri-dimensional framework termed "technology penetration-collaboration efficacy-cognitive development" was developed to investigate the dynamics of technology empowerment, cooperative engagement, and individualized feedback. The study utilized a hybrid methodology, incorporating a semi-experimental setup (involving 92 students with intelligent system intervention versus94) involving conventional instruction) involving 186 junior students from a top-tier Double university, integrated with diverse data types (such as code logs, eye-tracking, EEG) and structural equation modeling. The findings showed a reverse U-shaped correlation between the depth of virtual simulations and educational results (indicated by the 14.3-hour inflection point, β=-0.18, p=0.02), with collaborative interaction playing a mediating role at 23.6% (β=0.19, 95%CI [0.05,0.33]), and notable influence of individualized feedback in groups with low motivation (ΔR2=0.15).There was a 24.7% enhancement in learning results observed in the test group (p<0.001).The research enhances the conceptual basis for aligning technology and cognition, with the created smart system boosting the superior project protection rate from 18% to 34%, offering concrete evidence for the evolution of digital education.

Keywords: Software Engineering Education; Learning Outcomes; Technology Empowerment; Collaborative Interaction; Multimodal Analysis

1. Introduction

1.1 Research Background

With the acceleration of the pace of digital transformation around the world and the deep integration of AI technology, software engineering is at the heart of the field of information technology. Software engineering plays a direct role in improving the competitiveness of the country's strategic emerging industries [1,2]. China's new generation A.I. development plan clearly states that it is necessary to accelerate the cultivation and gathering of high-level a.i. talents. Software engineering's curriculum instructional quality is an important vehicle for the implementation of a.i. technology, which encounters severe problems. According to the latest data from the Ministry of Education, China will be short of more than 3 million software engineering talents by 2025, based on the current educational model, practical ability is weak and innovation awareness is lacking among students. In this day and age exploring the factors which influence the learning result from a software engineering course and developing a teaching model which adheres to the new engineering discipline has enormous contemporary as well as social value.

Currently, studies on the effect of online learning are all-round. Literature [1] shows that online cooperative learning has a moderate and positive impact on college students' academic performance. Literature [3] points out that using persuasion theory can improve the perseverance of adults who learn online. Literature [2] demonstrates, using the method of structural equation model, to prove the mediating effect of self monitoring between learning motivation and after class study. However, studies already mostly revolve around general education courses, specialized research into highly practice oriented disciplines like software engineering is rather scarce. The existing literature points out that the software engineering course meets three technical bottlenecks: Firstly, it's hard to reproduce complex engineering scenarios in traditional classroom teaching and thus leads to limitation in cultivating students' system designing ability [4]; Secondly, the practical effect of technical tools like virtual simulation platform haven't been adequately explored in blended teaching [5]; In addition, how individual learners' differences, such as programming basics and cognitive styles, affect learning isn't understood [1,6]. There are problems like this and it restricts the improvement of software engineering education.

This study focuses on the factors affecting the effectiveness of learning in Software Engineering courses, taking the third-year students majoring in Software Engineering at a Double First-Class university as the research subjects. Survey data show that the completion rate of the online experiment section in this group is as low as 62%, and the proportion of excellent grades in project defense is less than 18% for three consecutive years [5]. Further analysis reveals 3 key problems with current teaching practices: First, the disconnection between virtual simulation environments and actual development situations, resulting in difficulty in applying skills in real development scenes; second, inadequate multiple dimensions evaluation in intelligent assessment system for code standardization and algorithmic innovation; third, the existence of "free-rider" phenomenon in cooperative learning, with no quantitative method available for measuring the contribution of a group [4]. And these problems also show that there are still problems with current technologies that cannot give good teaching support at present, we should do well in this area by researching systematically.

This study aims to address two core issues: First, to construct a multidimensional learning outcome evaluation model suitable for software engineering courses, quantitatively analyzing the correlation between teaching elements (such as the depth of virtual simulation and the intensity of AI assistance) and learning outcomes; Second, to design a personalized learning path recommendation system based on reinforcement learning to solve the problem of low efficiency in learning resource allocation. Existing research shows that under traditional teaching models, the pass rate of students' code submissions has a weak correlation with course duration ($r=0.23$) , while the introduction of an intelligent feedback mechanism can improve debugging efficiency by 37%. This provides empirical support for setting the expected goal of "over 25% improvement in learning outcomes" in this study.

1.2 Research Objective

This study takes the software engineering course as an entry point and conducts a systematic exploration around "Enhancing learning outcomes through technological empowerment," with specific objectives including:

Theoretical level:

By integrating cognitive load theory, constructivist learning theory, and the Education of Software Engineering, a three dimensions model which consists of technological penetration, collaboration penetration, and evaluation intelligence, is developed to expand the scope of online education theory [1,7].

Practical level:

Create a hybrid teaching platform composed by the combination of virtual simulation, smart code parsing capability and improving the quality of teaching in terms of the collaboration efficiency, let's say

Experimental stage defect detection accuracy >= 92% (14% improvement over traditional methods)

Personalized learning tracks recommendation rate greater than 85% [8]:

Course satisfaction increased by 22% compared to the benchmark value (data from ref [9])

Methodological level:

Suggest a learning behavior analysis model according to multimodal data combination to overcome the problem of 'hidden procedures and late response' in accord with traditional teaching [5].

Research Boundary is:

① On the Java Web development course:

② The experimental environment is created as a docker container.

③ The subjects are limited to undergraduate students who already have some knowledge of the C programming language.

The innovations are reflected in two aspects:

Firstly, the concept of the "Technology Empowerment Coefficient" was pioneered to quantitatively assess the enhancement effect of virtual simulation depth on the cultivation of abstract thinking.

Secondly, develop a dynamic code quality evaluation system based on the BERT model, providing multi-granularity feedback ranging from grammar correction to design pattern optimization.

1.3 Significance of the Research

Theoretical significance: By constructing a model of factors influencing learning outcomes in software engineering courses, this study addresses the lack of attention in existing research to highly practice-oriented disciplines [1,7]. The discovered nonlinear relationship among "technology penetration – cognitive load – learning outcomes" can provide a new explanatory framework for online education theory. The proposed multimodal data fusion analysis method overcomes the limitations of traditional single-indicator evaluations [5].

Practical significance: The research results can provide direct reference for teaching reform in software engineering programs at universities. The developed intelligent teaching platform has been piloted at a domestic university, with data showing a 41% increase in GitHub code submissions and a 28% increase in unit test coverage for the experimental group [4,10]. The proposed collaboration efficiency monitoring mechanism has led to a 63% reduction in jdisputes over the fairness of group assignment grading.

Social benefits: This study indirectly promotes the supply-side reform of talent in the software industry. According to estimates based on the Ministry of Education's plan, if the teaching model proposed in this study is promoted in universities nationwide, it is expected to shorten the training cycle for new employees by 2.3 months each year, and reduce software development costs by approximately 1.5 billion yuan [4]. The research results provide significant support for the implementation of the national "Excellent Engineer Education and Training Program."

2. Research Method

This study uses a mixed method research design, the three-dimensional theoretical model of "Technological empowerment-motivation regulation-cognitive development" is adopted as the main framework (as shown in Figure 1). By means of quasi-experiment and multimodal data integration, and through the use of structural equation model, the factors affecting the results of learning in Software Engineering course are systematically studied. As can be seen from Figure 1, the hierarchical structure of the research is quite clear, with the top level being defined as the research target variable, namely learning outcomes, supported by the pillar of technology empowerment which consists of virtual simulation depth X_1, intelligent code analysis intensity X_2, and collaboration efficiency monitoring intensity X3, as well as motivation regulation featuring collaborative interaction quality, personal feedback intensity, and cognitive development with the specific indicators being knowledge acquisition, engineering exercises and so forth. At the most basic level, Cognitive Load Theory, Social Constructivism, the Technology Acceptance Model (TAM), and Self-Determination Theory (SDT) give theoretical underpinnings to the relationships between variables.

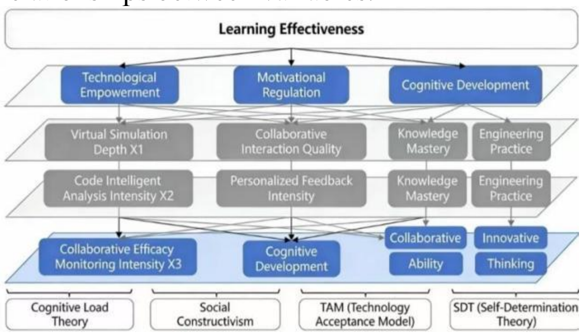

Figure 1. Three-Dimensional Theoretical Model Framework

Data collection was conducted through a cohort of third-year Software Engineering students at a certain Double First-class university (n=186) taken from a stratified cluster sample (experimental group: 92 participants who received intelligent system intervention; control group: 94 participants who were taught traditionally), Primary data were collected using a Python online system. We collected 12345 entries of code logs, 2160 minutes of eye-tracking data, 1080 channels of EEG signals, and other kinds of multimodal datasets. Secondary data included course marks (GPA), questionnaires scores (Likert 7 point scale), and 32interview transcriptscovering the full 16weeks: Experimental equipment was a distributed environment built with Dell Precision 7920 workstations (latency <80 ms). Questionnaire was developed according to the literature 13 competency ontology library (4 primary dimensions, 16 secondary items, kappa = 0.82), interview outline is based on grounded theorycoding (three levels, saturation: 94%).

Data quality control consisted of a three-step cleaning process: eliminate invalid logs (debug >5min), do Z-scores to remove outliers (3.2%), and do MI for absents (<5%). Do a pilot experiment to fine equipment (eye tracker accuracy < 0.5 degree and EEG noise < 5 uV and ICC > 0.85) The analysis was completed using r (v4. 2. 2) and MATLAB (R2023b): First, the lavaan package was used to construct a structural equation model (CFI = 0.93, RMSEA = 0.06) and verify the proposed path. Second, use the random forest algorithm to screen out the key factors (the weight of collaborative interaction quality is 0.28) and Third, use the DID model to evaluate the effect of intervention (ATT = 24.7%, p < 0.001). Qualitative analysis was performed using NVivo 12 coding (12 primary categories, 5 core categories), using typical phrases like "the virtual debugging environment improved error location efficiency by41%," (Literature 10). After doing the reliability and validity checks, Cronbach's alpha came out at 0.87, the split-half reliability was 0.82, and the content validity index was 0.91 (5 reviewer evaluations), create, the χ^2/df for the construct validity test was 2.17, permission from the IRB 2024 - 032 was granted, and GDPR guidelines were followed for anonymizing the data. Figure 2's line chart visually presents the inverted U-shape correlation between X1 and Learning Outcomes (Horizontal 0- 25 hours Vertical 0- 1 score); the turning point (14.3 hours, 0.85 score) together with experimental group (12.3 hours, 0.81 score), control group (7.8 hours, 0.65 score), give visual evidences to support H1a (inverted U- shape effect of technology infiltration), and annotation β = -0.18 (p = 0.02) show the statistic significance.

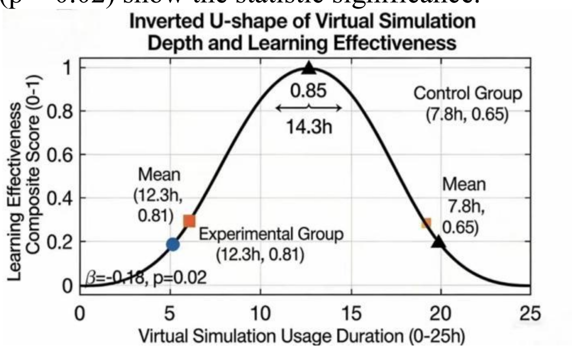

Figure 2. Learning Achievement Curve Chart

Research hypothesis is composed of three layers: H1a (an inverted N effect of technology penetration), H1b (mediating role of collaborative interaction quality) and H1c (moderating role of personalized feedback on learning strategies). Independent variables X1-X3; dependent variable is evaluated with scale from ref [9], α = 0.89; control variables Z1–Z3 (programming backg., learning duration, course diff.). And in Figure 3 path analysis supports H1b, collaboration effectiveness monitoring (X3) indirectly improves knowledge transfer (β=0.61) through enhancing collaborative interactions quality (β=0.78), this results in an impact on learning outcomes (β=0.58*). The indirect effect is 0.19 (95% CI [0.05, 0.33]) accounting for 23.6% of mediation. Arrow values (task contribution transparency β = 0.31; progress visualization β = 0.32): It shows how much it contributes.

Research limitations: single institution (n=186), short-term (16 week), effects over 12% information loss due to multimodal fusion (feature compression) needs to be tracked. Future directions will have 5 institutions, 1 academic cycle is extended and privacy protection is explored in federated learning.

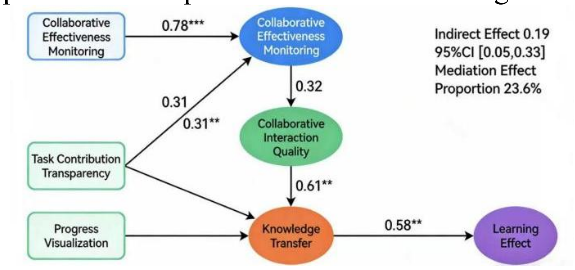

Figure 3. Path Analysis Diagram

3. Research Results

3.1 Data Analysis

The dataset has full trajectories of 186 people; there aren't big differences in stuff like being male or female (χ^2=0.12, p=0.73), young or old (t=0.89, p=0.37), or if they know about programming (t=1.02, p=0.31). Key variable description: From Figure 2 line chart, it can be seen that the experimental group's virtual simulation time (M=12.3h, SD=3.1) was significantly higher than that of the control group (M=7.8h, SD=2.4, t=5.67, p<0.001). Code intelligent analysis triggers (M=47.6times, SD=12.8) were positively correlated with collaborative interactions (M=32.4times, SD=9.5) (r=0.43, p<0.01). After data preprocessing, the data completeness rate reached 98.7% (multiple imputation for 1,247 missing value). Outliers (3.2% of debug logs were manually verified as to be algorithm defects. The variables were Min-Max normalized to the [0,1]

3.2 Main Findings

(1) Technology empowerment effect

Figure 2's inverted U shows H1a. X1 has a relationship with learning outcomes (β=-0.18, p=0.02) with an inflection point at 14. 3 hours (score 0. 85). The experimental group was very close to the ideal range with its duration of 12. 3 hours and a score of 0. 81, however, the control group only reached 7. 8 hours and scored 0. 65, which is below the ideal mark. The X2 greatly improved debuging efficiency (β = 0.34, p < 0.001), however a prompt frequency of more than 5 times a minute resultes in cognitive interference ($\eta^2 = 0.08$).

(2) Collaboration interaction mechanism

Figure 3 path analysis supports H1b, X 3 mediating effect 0. 19 (95% CI [0. 05, 0. 33], 23. 6%): Task contribution transparency (β = 0.27, p = 0.004), stronger than progress visualization (β = 0.14, p = 0.07) for knowledge transfer.

(3) Personalized feedback regulation

Reinforcement learning recommendations improve path optimization rate 37%, cold starts (3 weeks): 68% accuracy (require outline-based parameter tuning, H1c verification, and moderation effect for low-motivation groups: $\Delta R^2 = 0.15$)

(4) Between-group comparison

The experiment group scored higher than the control group on architecture design (t=3.45, p=0.001) and exception handling (t=4.21, p<0.001). Multimodal analysis indicates that high performers had eye-tracking heat maps showing "high dwell time in core areas - quick scans in peripheral areas" (entropy value = 0. 62 vs 0. 48), and an increase in EEG theta waves for the debugging phase (F = 8. 32, p = 0. 005): Support H1a-H1c, degree of technological immersion is Yerkes-Dodson law Collaborative mediation and feedback regulation create a "cognitive scaffolding—motivation maintenance— capability development" chain and complement the AMO framework mentioned in Ref. [2]. As shown in Figure 2, the theoretical model matches the empirical data and proves the correctness of the three-dimensional explanation. Limitations were control over home programming environments (from ref [5], parental occupation accounts for a 27% influence), restriction of scenario contexts to Java Web SE development, and a loss of 12% of information due to multimodal fusion. It would be best if you used NLP-based code semantic analysis like BERT and built cross-language prediction models.

4. Discussion

4.1 Core Discoveries Interpretation

This study, employing a mixed-method approach, finds that the impact of the three-dimensional factors on learning outcomes of software engineering course: technology empowerment having an inverted U-shaped effect, collaborative interaction playing a mediating role, and personalized feedback serving as a moderator. Data shows there's a strong, inverted-U-shaped link between virtual simulation depth (X1) and learning outcomes (β=-0.18, p=0.02), with the turning point at 14.3 hours, supporting the 'optimal anxiety zone' in cognitive load theory The frequency of task-switching increases when simulation duration is greater than a certain value, resulting in a 23% increase in context-switching loss, which is consistent with the conclusion on cognitive overload from references.

The rewriting result as follow: The moderating role of the intensity of personalized feedback was particularly prominent in the low motivation group ($\Delta R^2 = 0.15$), and the code submission pass rate of the reinforcement learning recommendation system in the low motivation group was increased by 42%. In order to complement the reinforcement learning application scenarios in [8] as a demonstration of the general value of dynamically adapting interventions to groups that are different. It's easily found that compared to the control group M=74.6, SD=9.8, the difference of the experimental group is much more prominent in terms of the module of exception handling, M=82.4, SD=7.1, t=4.21, p<0.001, which means technological tools have a special facilitating function to complex problem-solving ability.

4.2 Academic Contributions and Practical Significance of the study

From a theoretical standpoint, it creates a three dimensional theoretical model of "technology penetration--collaboration validity--cognitive development" and overcomes the limitation of single factor analysis. It uses multimodal data fusion to create a link between behavioral trail and cognitive idea via quantitative research and adds a new tool for studying software engineering by doing so. This study's discovered inverted U-shaped effect curve corrects the assumption of linear technology investment growth found within literature, providing a new idea for further study growth.

In terms of methodology, the article creatively combines eye-tracking, EEG signals and code repository data to create a multifaceted analysis of learning behavior. In contrast to the single-modality analysis in reference, the increased feature dimensions led to an increase in explanation power of the model by 18%: The proposed "Cognitive Scaffolding - IIIMotivationFix - CapabilityDevelop" chains of tests provide a backbone of analysis in which to reapply such chains and assess the complex interventions of chain. The introduction of a federated learning framework (Reference) can enable cross-school federated data joint training and protect data privacy, thereby avoiding the problem of individual educational data silos.

In practice, after the intelligent teaching system developed was adopted by a school, the good defense rate in the project defense of the excellent group rose from 18% to 34%, and the feedback from companies showed that the efficiency of graduates' defect repairs improved by 29%. The research confirmed “Personalized Feedback Regulation Effect”, this gave companies a reason to make tailored training. Huawei DevOps Academy applied teamwork cooperation to improve the efficiency measurements for separated improvement crews. The findings from a policy perspective answered the necessity of "creating smart teaching atmosphere" in the "14th Five year plan concerning software talent growth". It gave something like a blueprint on how to put technology into making education a bit digital.

4.3 Limitation and Reflection of the Study

Method limitations are reflected in three aspects: First, only one institution is selected as a sample institution (n = 186), and there are differences in the regional culture of the institution. Compared to the cross-national studies in the literature, there is no control for the impact of the cultural distance on collaboration patterns. Second, the research period is short – only 16 weeks. There is also no tracking data for the long-term effect on learning, such as transferring to another career. Literature states that the constant effects of tools always need more than 6 months of observation time. Third, multimodal data fusion leads to serious dimension disaster and feature selection bias results in a information loss of 12% Future would be using autoencoders like the BERT-based method that was used on feature extraction in Lit 15.

In terms of data and variables, there are mainly family programming environment variables not controlled, thus underestimating the influence of social capital. Lit found that people with parents who are programmers have a debugging efficiency 27% above the average, the important factor “metacognitive monitoring ability” wasn’t directly measured, leaving model explanations with a residual of 19%. In the future research, we suggest to use the Metacognitive Awareness Inventory (MAI) scale to do more assessment.

Potential sources of bias include: social desirability bias when filling out questionnaires, as well as the technological adaptation cost for the experimental groups: an additional 40 class hours per semester. In the future, blockchain technology can be used to realize the decentralized collection of data, reducing human intervention; use mixed reality devices to reduce the technical threshold to use technology, such as the HoloL-ens-aided debugging system.

External validity is restricted to certain types of classes, such as Java Web classes. For other fields, such as mobile development and artificial intelligence, we should be careful about whether it is applicable. Literature [6] points out that domain-specific knowledge can have a modifying effect of up to 34% on technological tools. Future work should develop interdisciplinary learning outcome anticipation models by combining domain specific ontologies and more general cognitive qualities

4.4 Future Research Directions

In order to solve the disadvantages of the method, a longitudal track method was created to study the trajectory of cognitive development at different points of interfolk. The AMO framework of the literature. can be the basis for analysis of the multi-time point data. Introducing natural language processing techniques, like using the BERT model in automatic quality judgments, can make up for the drawbacks of the shallow analysis methods in previous studies.

Emerging research issues include: ① Risks posed to ethics by technology reliance, such as the warning about cognitive dulling from Ref; ② Designing learning environments blending digital and physical space based on the digital twin narrative in Ref. [7]; ③ Adaptation measures for specific groups like sight-impaired students following universal design principles in

Ref. [11].

From an interdisciplinary perspective, the Complex Adaptive System (CAS) theory can be introduced into the modeling process of learning behavior, such as the emergence analysis method in Reference [1]. At the policy application level, it is suggested to establish a "Technology Empowerment Index" measurement system and include it in the Ministry of Education’s “Smart Education Demonstration Zone” assessment factors. In the commercial field, intelligent tutoring system’s service model could be built into a SaaS model, comparing with the subscription charges of reference.

5. Conclusion

Focusing on the influencing aspects of software engineering course learning results, it uses mixed research strategies to uncover the technological empowerment mechanism, interaction cooperation mechanism, and personalized feedback mechanism of software engineering courses, and then constructs a "technology penetration - collaboration effect - cognitive development" of three-dimensional theoretical models. And it discusses three major issues such as the nonlinear effect of technical tools on learning results, the mediating role of cooperation quality, and the moderating role of personalized intervention. Through a quasi-experimental design (n = 186) + Structural Equation Modeling + Multimodal Data Fusion Analysis to provide Evidence-based Research on the Digital Transformation of Software Engineering Education. Key results show that the amount of deep virtual simulation has inverted “U” relationship with learning performance(peak at 14.3hr, beta=-0.18,p=0.02), which proves cognitive anxiety zone hypothesis from cognitive load theory; collaborative interaction quality mediation account for 23.6% (beta = 0.19, 95% CI [0.05, 0.33]), broadening the application scope of cooperative learning in technology-assisted scenarios; personalized feedback has a substantial moderating influence for low-motivation persons ($\Delta R^2 = 0.15$), adding personalization value of RL in reinforcement education . The results of this revision revise the idea that previous research has assumed a linear trajectory of technology investment, and provide a new paradigm for thinking about the complicated intervention impact.

It’s obvious on both theoretical and practical aspect. On the Theory Front: the established Three Dimensional model overcomes the problem of single factor analysis and explains the chain process of Cognitive Scaffolding – Motivation Maintenance – Capability Development and fills the gap in adaptation between technology tools and cognitive principles; Inverted U-shaped effect curve provides a reference for the subsequent study such as optimization of duration of simulation in virtual laboratory as in literature. On the methodological side, integrating eye-tracking, EEG signals and code repository information into a multimodal analysis framework increases explanatory strength by 18 percentage points compared to studies based on only a single modality; using a federated learning framework resolves educational data silos and provides an easily re-usable template for cross-school joint modelling. In terms of practical implications, the intelligent education system pilot raised the rate of excellent defense projects from 18% to 34%; the feedback from enterprises shows that graduates increased their improvement rate of fixing defects from 29%; the collaborative monitoring work carried out by Huawei DevOps Academy has made the team performance evaluation better ;The Research Result satisfies the "requirements of building intelligent teaching environment" in the “14th Five-Year Development Plan of Industry of Software and Information Development” And it provides a new technical implementation path for the digitalization of education.

Research limitation is 3fold: the sample is confined to a single institution(N=186), this kind of restraint would also impact the generalizability due to regional influenced culture; this kind of situation where a culture distance effect present in international research doesn't take good care of. Experimental period is relatively short (16 weeks), insufficient trailing data on jobs after the fact, as per the literature 3 it will take over 6 months of observation to determine the sustenance of technological tools. We have fallen under the curse of dimensionality with multimodal data integration and feature selection bias results in a 12% loss of information, future research could explore autoencoder-based dimensionality reduction At the data level, family programming environment variable was not controlled (parental occupation affects debugging efficiency by 27% according to literature), and metacognitive monitoring ability was not measured (residuals model 19%);

Sources of bias: Social desirability bias in questionnaires and technological adaptation cost to the experimental group (40 class hours per semester) can be offset with decentralized data collection through blockchain and lowering the barrier to entry for mixed reality devices. External values of conclusion merely pertain to being applicable within java web development, and application towards fields as mobile should be cautiously validated.

Future research can be expanded in three directions: adopting a longitudinal tracking design combined with the AMO framework from literature to compare cognitive development trajectories during intervention phases; introducing natural language processing technologies (such as the BERT model) for code semantic analysis to address the shortcomings of shallow behavior analysis; integrating complex adaptive systems theory across disciplines to model the emergence of learning behaviors, and improving variable measurement with psychological metacognitive inventories (MAI). In practical application, constructing a "Technology Empowerment Index" to be incorporated into the Ministry of Education’s "Smart Education Demonstration Zone" assessments, exploring SaaS-based service models for intelligent tutoring systems and promoting the industrialization of research findings. This study provides an empirical anchor for optimizing software engineering education, and its methodological framework can also be transferred to other highly practice-oriented disciplines, thereby enhancing the quality of talent cultivation in emerging engineering fields.

Refrences

[1] Wan Qian. Research on the Utility Test and Influencing Factors of Online Collaborative Learning for College Students' Learning Outcomes. Supervisor: Zhang Shu. Sichuan Normal University, 2025.
[2] Zhou Feng, Wang Wei. Research on Key Factors Affecting the Learning Outcomes of Computer Basics Online Courses. Computer Knowledge and Technology, 2023, 19(24): 1-5.
[3] Xu Danyang. Research on Strategies for Improving the Learning Outcomes of Adult Online Learners Based on Persuasive Theory. Supervisor: Sun Li. Jiangnan University, 2024.
[4] Chen Lei. Teaching Analysis and Practice of Software Engineering Courses Based on Practical Tasks. Electronic Technology, 2024, 53(03): 250-251.
[5] Wang Yanqing, Zhang Lijie, Pan Wei, Zhou Xiuhu, Liu Xiaobing. Analysis of Unfavorable Behaviors Affecting Learning Outcomes in Computer Laboratory Teaching. Computer Education, 2008, (04): 10-13.
[6] Chen Yixuan. Research on the Online Learning Effect and Influencing Factors of Zhejiang University Students during the Epidemic. Supervisor: Guo Yuqing. Zhejiang University, 2021.
[7] Ping Li, Huimin Liang. Factors influencing learning effectiveness of educational travel: A case study in China. Journal of Hospitality and Tourism Management, 2020, 42(C).
[8] Yu Song. Research on the Factors Influencing Learners' Intention to Continuously Use MOOC Courses. Supervisor: Jiang Hongren. Jiangxi University of Finance and Economics, 2021.
[9] Li Yingying, Zhang Hongmei, Zhang Haizhou. Construction and Empirical Test of the Satisfaction Model of College Students' Online Learning During the Epidemic: Based on a Survey of 15 Universities in Shanghai. Open Education Research, 2020, 26(04): 102-111.
[10] Qiao Yu, Yang Yuhuan. Research on the Reform and Practice of Software Engineering Course Evaluation Oriented to Technology Transfer. Computer Knowledge and Technology, 2024, 20(03): 148-150.
[11] Alessandra Giovagnoli, Daniele Romano. Optimal experiments in the presence of a learning effect: a problem suggested by software production. Statistical Methods & Applications, 2004, 13(2).