

人工智能对软件工程实践教学的挑战及优化策略

何 兵¹, 何腾蛟², 朱 烨¹, 王 刚¹

¹成都信息工程大学, 计算机学院, 四川 成都

²暨南大学, 信息科学与技术学院, 广东 广州

收稿日期: 2025年5月6日; 录用日期: 2025年7月7日; 发布日期: 2025年7月16日

摘 要

软件工程作为理论与实践并重的学科, 其教学模式需紧跟技术发展与行业需求。文章分析了人工智能技术对软件工程实践教学带来的多维度挑战, 包括学生独立思维能力与编程能力弱化、学术诚信风险增加与评价体系失准、教师角色转型及教学策略适配难题, 以及学习动力可能受到抑制等问题, 并针对性地提出了教学方法革新、协作学习模式优化、实践驱动的教学内容设计以及评价体系完善等一系列优化策略。通过系统性教学改革, 实现技术赋能与教育目标的深度协同, 培养适应产业变革的高素质软件工程专业人才。

关键词

思维能力, 学习动力, 评价体系, 教学策略, 协作模式, 智能工具

Challenges and Optimization Strategies of Artificial Intelligence on Practical Teaching of Software Engineering

Bing He¹, Tengjiao He², Ye Zhu¹, Gang Wang¹

¹School of Computer Science, Chengdu University of Information Technology, Chengdu Sichuan

²College of Information Science and Technology, Jinan University, Guangzhou Guangdong

Received: May 6th, 2025; accepted: Jul. 7th, 2025; published: Jul. 16th, 2025

Abstract

Software engineering, as a discipline that emphasizes both theory and practice, requires its teaching

文章引用: 何兵, 何腾蛟, 朱烨, 王刚. 人工智能对软件工程实践教学的挑战及优化策略[J]. 创新教育研究, 2025, 13(7): 258-265. DOI: 10.12677/ces.2025.137523

model to keep pace with technological advancements and industry demands. This paper analyzes the multi-dimensional challenges posed by artificial intelligence (AI) technologies to the practical teaching of software engineering. These challenges include the weakening of students' independent thinking and programming abilities, increased risks of academic integrity violations and inaccuracies in the evaluation system, difficulties in the transformation of teachers' roles and adaptation of teaching strategies, as well as potential suppression of students' learning motivation. In response to these challenges, a series of optimization strategies are proposed, including the innovation of teaching methods, the optimization of collaborative learning models, the design of practice-driven teaching content, and the improvement of the evaluation system. The aim is to achieve a profound synergy between technological empowerment and educational objectives through systematic teaching reforms, thereby cultivating high-quality software engineering talents who are well-equipped to adapt to industrial transformations.

Keywords

Thinking Ability, Learning Motivation, Evaluation System, Teaching Strategies, Collaborative Models, Intelligent Tools

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

当前，数字化正以前所未有的态势改变着生活、工作以及社会的各个层面。软件技术作为数字化变革中的关键驱动力，不断地推动着智能制造、智慧城市、金融科技等领域的创新，引领着各个行业迈向新的发展阶段。统计数据显示，2024 年全球软件市场规模约为 6133.4 亿美元，预计到 2033 年将增长至 13005.5 亿美元，年复合增长率达 8.71%。然而，在软件行业迅猛发展的背后，人才供需的矛盾也日益凸显。市场对于软件技术理论深厚又能解决实际需求的复合型软件人才，已然成为企业亟需的稀缺资源。

在此背景下，软件工程实践教学作为衔接理论知识与产业应用的关键纽带，其重要性愈发凸显。它作为培养软件工程专业人才的核心环节，一直以来都致力于让学生通过实际项目操作，深入掌握软件开发流程、方法与技能，培养学生全生命周期软件开发与设计能力以及工程伦理意识。随着软件工程教育向实战化转型的加速，传统重理论轻实践的弊端已难以满足企业需求[1][2]，改革实践教学体系已然成为全球共识。从已有研究来看，传统软件工程实践教学已形成项目驱动、校企合作、案例教学、敏捷开发等多元化教学模式，学生经过需求分析、系统设计、编码实现、测试运维等各个环节锻炼，已能逐步构建起工程化思维体系、团队协作精神、问题解决能力和创新意识。

然而，随着人工智能技术的突破性发展，软件工程实践教学正面临着前所未有的挑战。特别是，近年来以深度学习、自然语言处理等前沿技术为依托的智能化编程工具的涌现，如 GitHub Copilot、Baidu Comate、iFlyCode 等[3]，在代码生成和优化能力上展现出了强大的优势。与传统工具相比，智能化编程工具不仅具备语法高亮、代码补全等基础功能，还可通过对上下文的理解并模拟程序员的思维，进行代码编写甚至代码优化。而对于开发者，只需通过自然语言描述所需功能，就能快速生成相关代码。这将大幅缩短开发周期，降低编程门槛。这一变化不仅影响着软件系统开发的实践，而且将对传统软件工程实践教学模式带来冲击。因此，在人工智能发展趋势下，如何适应时代发展需求，优化并改进软件工程实践教学，成为了当前亟待解决的重要问题。

2. 软件工程实践教学回顾

软件工程的实践性要求较高，其目标不仅是开发和维护等基本理论讲解，更重要的是培养学生的实践、创新和综合应用素质，形成“认知 - 应用 - 创新”学习与成长路径。目前来看，经过多年的探索与实践，新的教学理念、方法和技术手段已逐步引入到实践教学，众多高校已构建了更加贴近软件产业实际和创新的实践教学环境。

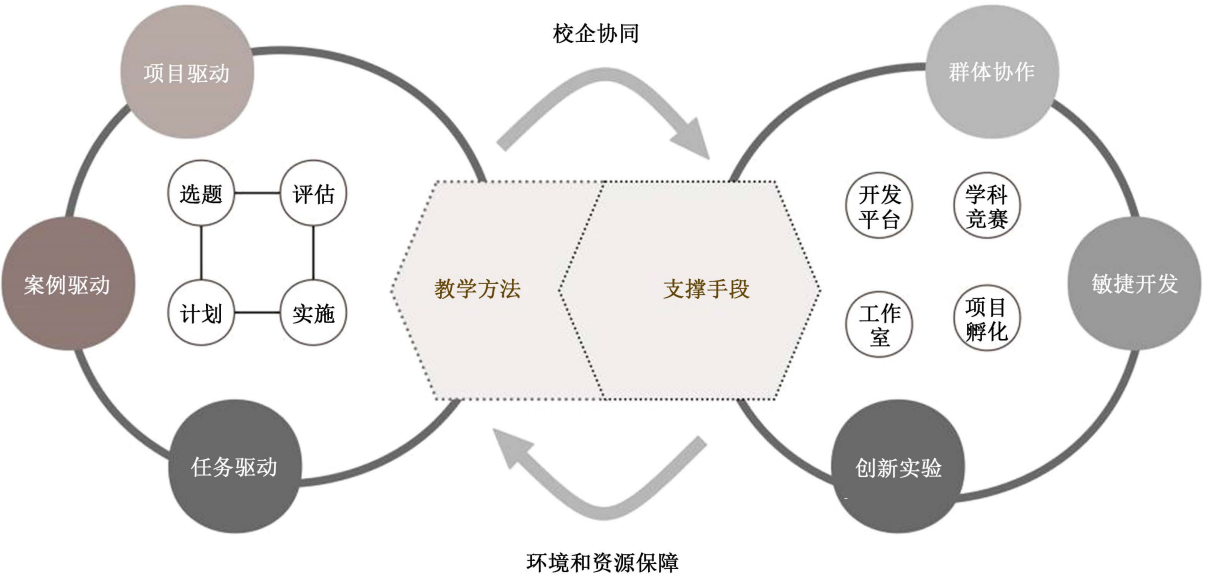


Figure 1. Practice-oriented teaching mode
图 1. 实践教学模式

1) 软件工程实践教学的改革与创新

软件工程实践教学模式已从早期以基础技能训练为主导、形式相对单一的教学模式，逐步迈向多元化、系统化与实践化深度融合的轨道。在此进程中，也产生了一系列具有创新性的实践教学改革举措，如基于群体化、企业开发平台、敏捷开发、案例教学等改革与创新方法(见图 1)。在实践教学过程中，教育者积极探索并构建了分层、模块化、渐进式的实践教学体系。依据学生的认知规律和能力发展水平，划分为基础认识性实践、综合应用性实践和研究创新性实践三个递进的实践教学阶段[4]，强调通过群体协作促进知识共享与能力互补[5]。逐步建立起对软件工程基本概念、原理和方法，提升实践操作水平和问题解决能力，并培养学生的创新思维和探索精神。而 IT 企业软件开发平台的引入[6]，为实践教学提供了真实的团队开发环境和先进的开发工具，使学生能够在接近企业实际的工作场景中进行团队协作开发。除此之外，为适应软件行业快速迭代的特点，敏捷开发方法更强调学生在实践过程中学会如何高效沟通、如何快速响应变化、如何共同解决问题[2]。以涵盖不同地域、行业和时效性案例库的形式，通过多样的实践素材和真实的问题情境，提升学生的自主学习、创新能力和实践技能业已成为普遍的做法[7]。

2) 软件工程实践教学的具体实施方法

在强化学生创新创业能力培育的时代背景下，传统软件工程教学模式与企业实际需求之间的脱节问题日益凸显。在此形势下，以企业真实项目贯穿实践教学全流程已成为当前软件工程实践教学领域的主流发展趋势。其中，项目驱动教学法 PBL 作为一种将真实或模拟的工程项目深度融入教学过程的有效方法广受欢迎[8]。它以具体且明确的任务需求为导向，不仅注重学生专业知识和技能的传授，更着重培养学生的团队协作精神、项目管理能力以及创新思维，逐步将企业项目所需的关键技术及环节融入教学[9]。

同时,为有效解决项目式实践中项目同质化以及跨学科复杂工程问题,辅以案例库作为支撑[10],通过“选-计划-实施-评估”的完整闭环[8],推动学生不断解决复杂工程问题的能力。

其次,学科竞赛已成为反哺软件工程实践教学的一种行之有效的方法。借助学科竞赛平台孵化项目,能够将竞赛项目实施过程中积累的各类资源与宝贵经验进行系统梳理与提炼,形成具有高度可借鉴性的实践成果,并进一步将这些竞赛案例转化为优质的教学资源,实现竞赛成果向教学效益的有效转化[11][12],提升学生竞赛表现与增强其实践能力的双重目标。然而,以学科竞赛反哺实践教学的有效实施,离不开健全完善的工作室机制的有力支撑。一方面,需要聚焦特定技术方向并建立初级、中级、高级项目库,以适应不同年级和能力水平的学生。另一方面,需要固定的工作室,并形成教师选拔高年级带动的“传、帮、带”机制,以此形成良好的学习氛围与传承体系,从而产生积极的教学效果。但此模式需持续投入硬件资源[13],且依赖教师的企业经验。

另外,在产教融合持续深化的背景下,校企协同模式正经历着从传统简单实习模式向全流程共建模式的深刻转型。以 DevCloud 为例,赵伟[14]在实施敏捷开发后,迭代效率提升了 50%。陈建平[15]则构建了“八环节”模型通过校企双导师制重构课程体系,其中企业案例占比达 70%。然而,在这一转型过程中,也暴露出了一些亟待解决的问题。姜立凯[16]指出师资工程经验不足仍是痛点,需建立教师企业挂职长效机制。与此同时,在新工科的时代背景下,社会对人才综合能力的要求日益提高,这也对校企协同模式提出了新的挑战和更高的要求。

3. 人工智能赋能实践教学的优势

AI 在软件工程实践中的首要优势在于能够为学生提供高度个性化的学习路径和智能辅导。传统教学模式往往采用“一刀切”的方式,难以满足学生在基础、兴趣和学习节奏上的差异。对于基础薄弱的学生,AI 可以自动生成基础语法练习题并提供分步讲解;对于能力较强的学生,则推荐项目实战任务。此外,在具备实践数据积累的条件下,AI 还能根据学生的兴趣方向推荐相关学习资源,实现“因材施教”。通过 AI 辅助的个性化学习支持,激发学生的主动性和成就感,帮助他们在建立持续进步的正向循环。

AI 工具的另一显著优势是能够提供即时、精准的反馈和自动化评估。在编程实践中,基于自然语言处理技术的 AI 编程助手能够理解学习者用自然语言描述的编程问题,并生成相应的代码示例或解决方案。这不仅降低了初学者的入门门槛,还能帮助学生快速跨越“不知道如何开始”的障碍期。在代码编写过程中,AI 工具通过静态代码分析、动态测试用例执行和模式识别技术,可以实时检测学生的代码缺陷,并给出具体修改建议,甚至能够优化代码结构。这种即时反馈机制显著缩短了“尝试-纠错”的周期,让学生能够快速调整思路。同时,借助 AI 也可以开发自动化评估系统,实现从代码规范性、功能完整性和性能等多个维度生成评分报告,通过高效、客观的评估方式,帮助教师提前干预潜在的学习困难。

AI 工具还能通过数据驱动的方式优化整体教学策略并整合优质资源。传统教学中,教师难以及时掌握全班学生的知识掌握情况,而 AI 可以通过学习分析技术可视化班级的薄弱知识点分布,帮助教师调整授课重点。此外,AI 能整合多源教学资源,根据课程目标智能生成教学案例或实验任务,确保教学内容与行业需求同步。这种资源整合能力可以丰富教学手段,推动课程内容的持续更新,使软件工程实践更具前沿性和实用性。

4. 人工智能对实践教学的冲击

当前,人工智能技术持续演进,其所提供的自动化、智能辅助系统等开发工具,为软件工程实践教学提供了诸多便利。但其也带来了软件开发实际流程与需求的变革,引发了多维度挑战,具体表现为以下几个方面:

1) 学生独立思维能力弱化与学习动力抑制

智能工具能够快速生成代码框架与错误修复方案[17]，但其高便捷性导致学生对工具的依赖性增强。部分学生在实践任务中倾向于直接获取工具的现成解决方案，而非主动分析问题逻辑或探索算法设计[18]。调查发现，中低绩效学生更频繁使用 ChatGPT 追问答案，而高绩效学生仅用于前期辅助[19]。以网络应用开发为例，学生可能借助大语言模型直接生成网站代码，却无法阐述 HTML、CSS 等技术原理与实现细节。当项目调试过程中出现异常时，缺乏独立排查问题的能力[20]。从长远来看，长期依赖这类工具会严重削弱学生的调试能力和算法设计思维。编程不仅仅是写出代码，更重要的是能够发现并解决代码中的问题，以及设计出高效合理的算法。若学生始终依赖 AI 来完成这些工作，自身在这方面的能力必然得不到有效锻炼和提升，在面对复杂多变的编程任务时，将难以独立应对。尤其在团队协作实践中，工具无法替代教师在小组矛盾调解或创新思维激发中的作用，并且可能削弱学生的协作意愿与创造力。

2) 学术诚信风险与评价体系失准

工具的代码生成功能增加了学术不端行为的监管难度。在学术诚信风险层面，主要存在过度依赖与思维惰性、数据隐私与伦理问题、误导性内容以及评价环节潜藏的风险。学生可能直接复制如 ChatGPT、GitHub Copilot 等 AI 工具生成的代码，部分学生还会因工具的便捷性放弃自主思考，将 AI 生成的代码当作作业提交，从而引发学术剽窃风险[19]。数据隐私与伦理问题上，AI 工具可能记录学生提交的代码或数据，存在敏感信息泄露风险。同时，使用公开数据集生成的代码可能涉及知识产权争议。此外，AI 生成的代码可能存在逻辑错误或效率低下问题，若学生未经验证直接使用，就可能被误导[3]。而对教师来讲，学生可能通过智能工具快速完成编程作业或考试任务，导致教师难以区分代码原创性、难以准确衡量学生对核心概念的真实掌握程度[21]。

3) 教师角色转型与教学策略适配挑战

智能工具的介入迫使教师从知识传授者转向学习引导者与过程监督者。一方面，AI 具备自动生成教学内容、提供个性化学习资源的能力[21]，这使得教师角色从传统的知识传授者向学习引导者转变，更注重组织学习过程、设计高阶任务，如项目式学习[20]。以编程课程为例，教师无需再逐行讲解语法，而是借助 AI 辅助学生自主纠错，自身则聚焦于算法优化和系统设计等复杂能力的培养。另一方面，由于 AI 缺乏情感交互能力。尽管 AI 工具如虚拟助教能通过模拟人类语言提供积极反馈、挫折疏导等情感支持和激励，但本质上这种互动是预设算法响应，缺乏真实人际交流的深度与灵活性，比如无法像教师那样通过表情、肢体动作等非语言线索感知学生情绪变化。教师需弥补这一短板，通过一对一辅导、团队协作任务等方式增强学生的社会性学习体验[22]。同时，教师还需引导学生批判性评估 AI 生成的内容，避免学生过度依赖技术。此外，在新工科背景下，教师还需承担起跨学科整合与课程设计者的角色，将 AI 技术与专业领域相结合，融合计算机智能技术与工程实践[23]。

以上挑战表明，智能工具的应用需结合系统性教学改革，既需优化工具本身的功能边界界定，亦需同步重构教学评价、师资能力与资源生态体系，以达成技术赋能与教育目标的深度协同。

5. 软件工程实践教学优化策略

5.1. 教学方法革新

在 AI 带来的影响之下，教学方法的也将呈现出多元化特征。在学术诚信保障层面，教师肩负着关键职责。一方面，引导学生对 AI 生成的内容展开批判性评估。在编程教学中，指导学生通过验证代码的正确性、深入剖析代码中的逻辑漏洞等方式，培养学生独立思考和判断的能力。另一方面，为减少抄袭现象，应明确禁止学生直接提交 AI 生成的作业，同时精心设计开放性问题或创新性项目，激发学生的创新思维和自主探索精神。

教师角色也亟需重新定义和定位。传统实践教学模式下，教师侧重知识传授。新形势下，教师将转变为学习的引导者，更多地承担起学习组织者和促进者的角色。因此，在 AI 工具辅助的前提下，实践老师更应为学生提供个性化的辅导，并聚焦于批判性、创新性等高阶思维的培养。此外，AI 以助教的形式来处理作业批改、知识答疑等重复性任务，而教师则可以将更多的精力投入到创新教学设计和情感支持上。通过给予学生充分的人文关怀，在线下课堂中加强师生之间的互动交流，弥补 AI 在情感沟通方面的不足。

教学方法的创新是推动教学改革的核心动力。可采用项目驱动的模式，通过布置真实场景任务，激发学生的实践能力和创新思维。利用翻转课堂与混合式教学融合、逆向设计实验的方式，课前要求学生 AI 推送的知识点通过教材、视频等多样化的学习资源自主学习基础知识，课堂上则专注于实践指导、组织讨论以及解答学生的疑难问题，加深学生对知识的理解和掌握。此外，还可依据 AI 分析的学习数据，为每个学生制定个性化的学习路径，并根据学生的学习情况动态调整学习难度，满足不同学生的学习需求，如为学习薄弱的学生推荐基础练习，为学有余力的学生提供挑战任务。

5.2. 协作学习模式

AI 辅助下的协作学习模式可优化策略分为以下几个方面。在线上线下的协作策略上，一方面采取双师协同教学模式，线上由 AI 教师发挥优势，通过大数据分析学生作业错误类型，为学生提供个性化编程路径推荐、实时代码纠错及智能反馈，并自动推送针对性练习；线下真人教师则聚焦情感引导和项目实践，例如组织小组调试竞赛，结合 AI 分析出的共性问题进行重点讲解。另一方面构建无缝衔接平台，整合 GitHub Copilot 等工具支持线上协作编程，同时线下课堂借助其他辅助技术模拟团队开发场景，营造真实开发氛围。在师生互动革新方面，引入 AI 虚拟助教支持，将学生提问自动分类为语法错误或逻辑问题；还能利用情感识别技术监测学生学习焦虑情况，触发教师介入机制。同时，基于 AI 对学生能力画像实现动态分层分组，自动组建异质化小组，如将编程能手与设计特长者搭配，并为薄弱学生分配 AI 生成的脚手架代码，防止学生搭便车。推行人机结对编程，初级阶段学生与 AI 结对，AI 实时建议代码优化，高级阶段转为真人轮换结对，AI 仅在代码审计时标记冗余代码，担任评审员角色。

5.3. 实践驱动的教学内容

AI 辅助下软件开发实践驱动的教学内容主要呈现于多个方面。在个性化学习路径设计上，一方面借助自适应学习系统，利用 AI 深入分析学生的学习行为、进度以及能力水平，从而动态地调整学习内容和难度；另一方面构建丰富的教学案例和项目库，学生能依据自身兴趣选择项目，而 AI 会辅助生成契合其需求的个性化学习路径。在智能编程辅助工具的应用方面，集成如 GitHub Copilot、讯飞星火等 AI 编程助手，实现代码生成与纠错功能，并给出优化建议。同时，运用 AI 技术进行自动化评估，自动批改代码、分析代码质量，涵盖相似度查重、性能评估等方面，并提供可视化反馈。在项目驱动的实践教学，引入行业实际案例作为真实场景项目。此外，鼓励学生开展跨学科整合，将 AI 与其他领域相结合，开发具有创新性的应用。

5.4. 评价体系优化

实践评价体系涵盖以下几个方面。在多元化评价指标上，利用 AI 技术从代码可读性、可维护性、健壮性、安全性等维度对代码质量进行自动化评分。利用在线学习时长、错误修复次数、提交频率等学习行为数据，对学习进度开展过程性评价。通过项目成果、团队协作表现及 AI 生成的个性化反馈，评估学生的解决问题能力和创新思维，衡量其创新能力与协作能力。智能化评价工具方面，集成 GitHub Copilot、通义灵码等静态/动态分析工具进行自动化代码分析，检测代码缺陷并提供实时优化建议。动态反馈与个

性化调整方面, AI 依据学生编程错误实时推荐复习资料或调整题目难度, 构建实时反馈机制。针对学生算法逻辑缺陷等薄弱点, 生成针对性练习, 形成自适应学习路径以优化学习。人机协同评价方面, AI 初步评分后教师复核代码创新性等关键指标, 确保评价全面性。并借助 AI 分析学生互评数据, 识别共性问题和个体差异, 实现学生自评与互评。实践与项目导向评价方面, 通过 AI 分析项目完成度、功能实现复杂度及文档规范性, 评估学生解决方案的实用性。

5.5. 自动化工具支持

自动化工具在实践进程中正逐渐成为不可或缺的关键要素。在智能代码生成与补全环节, GitHub Copilot、Baidu Comate、通义灵码等工具发挥着重要作用。这些工具能够显著减少开发人员的重复性输入操作, 并具备支持复杂逻辑生成的能力, 从而有效提升代码编写效率。在自动化测试与调试领域, 相关工具可自动检测代码中存在的语法错误、逻辑漏洞等缺陷, 为代码质量提供基础保障。同时, Cursor、Tabnine 等 AI 工具集成于集成开发环境(IDE)中, 能够即时提示代码错误并提供针对性的修复建议, 进一步优化代码调试流程。在代码质量优化与审查方面, 通过分析标识出具有高维护风险的代码模块, 为后续的代码优化提供明确方向。集成 GitLab 等平台进行自动化分析, 可确保代码风格在跨团队协作过程中保持一致性。在个性化学习与开发辅助的教育领域, 利用 AI 生成分难度的编程题目, 为学生提供定制化的实训内容。同时, 通过分析学生的代码提交历史, 为学生提供个性化的学习路径建议。

6. 总结

在人工智能技术迅猛发展背景下, 软件工程实践教学面临多维度的深刻挑战。智能工具广泛使用, 学生独立思维能力与编程能力将出现弱化趋势。不仅如此, 智能工具的即时反馈虽能带来一定激励, 但过度依赖却会削弱学生的内在学习动机。长此以往, 将难以应对复杂多变的编程任务。同时, 智能工具生成的代码使得学术不端行为监管难度加大, 评价体系将面临失准困境, 教师也将难以准确衡量学生对核心概念的真实掌握程度。此外, 教师角色转型及教学策略适配难题凸显, 教师需从知识传授者转变为学习引导者与过程监督者, 并适应智能工具带来的教学变化。针对这些挑战, 本文针对性地提出了一系列优化策略, 涵盖教学方法革新、协作学习模式优化、实践驱动的教学内容设计以及评价体系完善等方面。通过系统性教学改革, 以实现技术赋能与教育目标的深度协同, 从而培养出适应产业变革的高素质软件工程人才。

基金项目

成都信息工程大学人才引进计划(KYTZ202261); 全国高等院校计算机基础教育研究会计算机基础教育教学研究项目(2024-AFCEC-241); 成都信息工程大学本科教学工程(JYJG2024164, JYJG2024024); 成都信息工程大学自动化学院教改项目(ZDHXYJG-2024004)。

参考文献

- [1] 刘原. 协作学徒制在计算机专业实践教学中的实施[J]. 计算机教育, 2015(14): 103-106.
- [2] 廉咪咪, 张聪品, 范黎林. 敏捷开发在软件工程实践教学中的应用探索[J]. 计算机教育, 2021(6): 155-158.
- [3] 尹良泽, 徐建军, 李姗姗, 等. 人工智能时代下的计算机程序设计课程教学探索[J]. 计算机教育, 2025(2): 123-127.
- [4] 闫俊伢. 基于能力培养的软件工程专业实践教学研究与探讨[J]. 内蒙古师范大学学报(教育科学版), 2014, 27(11): 143-145.
- [5] 毛新军, 尹良泽, 尹刚, 等. 基于群体化方法的软件工程课程实践教学[J]. 计算机教育, 2018(7): 14-17.

-
- [6] 贾经冬, 林广艳, 谭火彬. 基于企业开发平台的软件工程实践教学改革[J]. 计算机教育, 2020(5): 123-126.
 - [7] 韩锐, 刘驰. 基于大数据案例的软件工程实践教学设计[J]. 计算机教育, 2023(2): 173-176+181.
 - [8] 胡炜, 罗来俊. 基于项目驱动的软件工程实践教学研究[J]. 赤峰学院学报(自然科学版), 2015, 31(7): 229-230.
 - [9] 邓泽林. 以企业项目驱动的软件工程专业实践教学研究[J]. 计算机教育, 2015(17): 91-93+110.
 - [10] 刘海燕, 王雅轩, 陈恒. 基于项目案例驱动的《软件工程》实践教学研究[J]. 科技创新导报, 2015, 12(14): 142-143.
 - [11] 蒋湘涛, 辛动军, 黄辉, 等. 学科竞赛促进软件工程专业实践教学模式探索[J]. 软件工程, 2018, 21(4): 58-60.
 - [12] 周泽寻, 林晓珊, 邱树伟. 融合 AI 与软件工程的 OBE-CDIO 创新创业实践教学改革[J]. 计算机教育, 2024(4): 139-143.
 - [13] 戴平. 基于“企业项目情境”的软件工程实践教学模式研究[J]. 赤峰学院学报(自然科学版), 2017, 33(11): 18-20.
 - [14] 赵伟, 卫琳. 以产教融合为导向的软件工程课实践教学改革[J]. 计算机教育, 2023(4): 177-180.
 - [15] 陈建平, 王建彬, 陈昌兴, 等. 一种基于产教融合的软件工程专业实践教学模式[J]. 肇庆学院学报, 2019, 40(5): 49-54.
 - [16] 姜立凯. 软件工程专业实践教学体系中企业合作机制的探讨[J]. 信息通信, 2016, 29(11): 288-290.
 - [17] 曲海成, 孙宁, 刘腊梅. 新工科背景下软件工程专业实践能力提升改革与实践[J]. 高教学刊, 2024, 10(15): 133-137.
 - [18] 张红卓, 周小宝, 许玉焕, 等. 生成式人工智能赋能计算机程序设计类课程教学创新[J]. 计算机教育, 2024(7): 44-48.
 - [19] 张伟均. 基于生成式人工智能的编程学习反馈策略设计与应用研究[D]: [硕士学位论文]. 广州: 广州大学, 2024.
 - [20] 刘凌, 吴永芬, 陈卫卫, 等. 大语言模型对高校程序设计类课程的挑战与机遇[J]. 计算机教育, 2025(2): 118-122.
 - [21] 赵璐, 蔡惠, 王烟濛, 等. 生成式人工智能赋能程序设计课程教学研究[J]. 大学教育, 2025(8): 33-39.
 - [22] 李莹. AI 双师编程线上线下双翼融合教学策略探索[J]. 科教导刊, 2024(17): 132-134.
 - [23] 马晓原. 基于项目式学习的人工智能课程教学设计与实践研究[J]. 中国信息化, 2025(3): 64-66.