

大模型驱动的软件工程实践课程改革与实践^{*}

康雁 张璇

杨云 朱锐

云南大学软件学院软件工程系, 昆明 650091

云南大学软件学院人工智能系, 昆明 650091

摘要 针对人工智能背景下软件工程专业应用型人才的需求, 以市场需求和毕业目标为导向, 本项目一方面通过基于大模型的智能化组件引入, 改进传统的软件开发、测试和运维工作, 改变软件人员的开发方式和工作习惯; 另一方面, 让软件工程人员掌握一定的人工智能技术, 完成智慧化软件项目的开发。为了更好地适应大模型技术代表新质生产力给软件工程领域问题解决方式带来的快速变化, 本项目面向软件工程工程认证的毕业要求, 结合大模型进行软件工程创新实践评价目标改革, 最终有效地提升学生的创新实践能力、人机协作能力、批判性验证能力和跨领域融合能力。最终学生团队代码编写速度平均提升 35%, 学生借助大模型生成的代码注释增加 70%, 项目熟悉时间平均从 1 周减少至 3 天, 软件项目中 100%融合人工智能技术。近两年, 软件工程系的就业率均为 90%以上。

关键字 教学改革, 新工科教育, 工程认证, 软件工程, 能力培养

Reform and Practice of Software Engineering Practice Course Driven by Artificial Intelligence^{*}

Yan Kang, Xuan Zhang

Yun Yang, Rui Zhu

Department of Software Engineering,
School of Software Yunnan University
Kunming 650091, China;

Department of Artificial Intelligence,
School of Software Yunnan University
Kunming 650091, China;

Abstract—In response to the demand for application-oriented talents in software engineering amid the context of artificial intelligence, and guided by market needs and graduation goals, this project, on one hand, improves traditional software development, testing, and operation and maintenance work by introducing intelligent components based on large models—thereby changing the development methods and work habits of software professionals. On the other hand, it enables software engineering personnel to master certain artificial intelligence technologies and complete the development of intelligent software projects. To better adapt to the rapid changes in problem-solving methods within the software engineering field, driven by the new productive forces represented by large model technology, this project aligns with the graduation requirements for software engineering program accreditation and integrates large models to reform the evaluation objectives for software engineering innovation practices. Ultimately, it effectively enhances students' capabilities in innovative practice, human-machine collaboration, critical verification, and cross-domain integration. As a result, the average code-writing speed of student teams has increased by 35%, code comments generated by students with the assistance of large models have risen by 70%, the average time to familiarize themselves with projects has been reduced from 1 week to 3 days, and 100% of software projects have integrated artificial intelligence technologies. Over the past two years, the employment rate of the Software Engineering Department has remained above 90%.

Keywords—Teaching reform, New Engineering Talent Training, Engineering Certification, Ability Training

1 引言

大模型技术代表新质生产力给软件工程领域问题解决方式带来快速的变化。大模型驱动的软件工程相关工作的智能化, 对软件工程应用型人才的培养提出了新的挑战。随着大模型人工智能的快速发展, 一方面, 在企业级应用软件系统中, 基于大模型的智能化组件不断引入, 正在逐步取代低端的软件开发、测试和运维工作。在此环境下, 需要改变软件人员的开发

方式和工作习惯; 另一方面, 人工智能的出现, 加速了智能化应用的发展, 软件工程人员需要掌握一定的人工智能技术, 才能完成智慧化软件项目的开发。为了适应当前社会需求, 围绕学生实践能力培养, 多所高校进行了各种有意义的实践教学改革。一些改革主要针对软件工程系统能力培养和解决复杂问题的能力, 依托学科竞赛、创新创业教育、校企合作、项目课题等对学生进行实践能力的培养, 如构建以学生为中心面向产出的实践教学和课程体系等[1-5]。但仍需有效地融合大模型和软件工程的全生命周期, 结合专业的培养目标提升软件工程学生的实践和创新能力。为此, 软件学院+人工智能学院在第六学期开设“软件

*** 基金资助:** 云南大学 2023 年教育教学改革研究重点项目“新工科背景下‘人工智能+’复合工程型人才培养模式改革与创新”(2023Z06)。

“工程创新实践”课程，面向工程认证的要求，主要强化学生在软件工程专业领域的实践开发能力、创新研究能力以及团队协作精神。通过让学生了解大项目的前沿技术培养学生的自学能力，为毕业论文工作奠定基础。

云南大学软件学院具备坚实且多元的工作基础，设立了国家软件人才国际培训基地（昆明）、国家Linux技术培训与推广中心、中国信息安全产品测评认证中心云南测评中心、国家软件人培育改革创新试验区等国家级基地和中心，为人才培养与技术推广筑牢根基。“软件工程”专业是云南大学第一个通过工程认证的专业，每年招收学生100余人，课程体系紧跟技术演进趋势，强调能力导向与综合素质提升，与甲骨文、东软、南天、青软等多家头部企业共建了多个实践基地，实行“双导师制”与项目驱动式教学，致力培养具有软件工程领域复杂工程开发设计能力的高素质复合型人才。学生在ACM、互联网+、挑战杯、全国软件创新大赛等比赛中累计获国家和省级级奖项30余项，培养质量和社会声誉持续提升。

2 现有问题分析

大模型技术和智能化软件的出现带给软件工程领域从业者新的更高的要求，对软件工程人才的专业素养和实践能力提出了更高的要求。现有软件工程实践教学的特色性不显著并且前沿性不强；学生很难在短时间进行大项目的训练，与实际的工程实践要求有明显的差距；以往的课程实践学生大量的时间花在平台搭建和一些重复工作中，使得创新和实践能力的培养时间有限。结合大模型带来的机遇，本课程深化专业课程建设，在整个课程教学过程中有效地融合大模型技术，为学生的实践和创新能力培养提供有力的支持。

3 大模型驱动下软件工程创新实践人才培养

针对软件工程系统能力培养和解决复杂问题的能力，为适应新工科发展需求，本课程依托大模型对学生进行实践能力的全面培养，构建以学生为中心面向产出的实践教学体系；关注学生创新能力培养，构建基于大模型的新的实践评价体系。

3.1 培养目标改革

本课程为第六学期的实践课程。主要强化学生在本专业领域的项目开发能力、理论研究能力以及团队协作精神、了解软件工程安全的前沿技术、培养学生的自学能力，为毕业论文工作奠定基础。传统软件工程教育侧重于编程基础、设计模式和工程管理，而AI时代需要培养具备以下能力的新型人才：人机协作能力：熟练使用AI工具（如GitHub Copilot、ChatGPT）

辅助开发，理解其优势和局限。AI增强的工程思维：能够设计AI友好的软件架构（如支持动态生成的模块化系统）。批判性验证能力：对AI生成的代码、设计或文档进行有效性审查与优化。跨领域融合能力：结合AI技术解决特定领域问题（如医疗软件中的NLP处理）。

面向软件工程工程认证的毕业要求，本课程的目标见表1。

表1 课程目标与毕业要求指标点支撑关系表

毕业要求	指标点	课程教学目标	权重
2、问题分析	指标点2.3能认识到解决问题有多种方案可选择，能通过图书、文献资料、网络资源等通道，寻求软件工程问题的多种解决方案	课程目标1：针对项目的特性，通过各种资源的查阅、研究与分析，考虑完成创新项目的多重方案，设计实现项目的解决思路和方法	1.0
3、设计/开发解决方案	指标点3.3能够在软件算法、模块和系统的开发与设计环节中体现创新意识	课程目标2：能针对项目特性，在项目开发和设计中应用新技术或新知识	0.4
		课程目标3：设计新应用同时进行新技术的学习，并将其应用到项目开发与设计中	0.6
4、研究	指标点4.1能够基于科学原理和软件工程专业知识，通过文献资料研究，调研解决复杂软件工程问题的思路和方法	课程目标4：能够根据基于之前掌握的实现系统的一系列知识、工具和方法，通过查阅相关资料进行研究，调研解决项目的思路和方法	1.0
5、使用现代工具	指标点5.3能够针对具体的业务对象和需求，开发或选用满足特定需求的软件工具，对工程问题进行模拟和测试，并能够了解其局限性	课程目标5：能针对项目特性，对现有工具进行比较，选择适当的工具进行项目的开发，并对具体工程问题进行分析	1.0
9、个人和团队	指标点9.2在团队合作中能够承担负责人、合作者或团队成员的任务和职责，能组织、协调和指挥团队开展工作	课程目标6：能够承担负责人、合作者或团队成员的任务和职责	1.0

软件工程创新实践课程聚焦“技术深度×产业广度×创新思维”三维融合，培养能解决复杂工程问题、

具备商业敏感性的高阶人才。课程的核心定位不再是单纯的技术课或创业课，而是“真实问题驱动”的全栈能力训练场。课程的培养目标是培养具有敏捷开发、云原生、AI 工程化等现代软件工程能力的创新人才。课程形成从需求洞察到产业验证的闭环思维。软件工程创新实践课程按相关目标进行实施，教学中进行教师教学、学生汇报、小组讨论等多种方式，有效地利用现代教学工具，针对课程目标进行教学。

学生针对工程认证课程目标 1，学习和掌握软件工程的基本概念；理解软件的生命周期，掌握软件需求工程、软件开发、测试和进化的过程模型的概貌；根据软件项目的特色，能够确定软件边界，并使用 UML 进行软件需求建模。

学生针对工程认证课程目标 2，能针对软件项目的需求，掌握面向对象的设计和分析方法，了解敏捷开发等新型软件开发方法。能够对复杂的软件工程问题进行分析，设计复杂软件工程问题的解决方案。

学生针对工程认证课程目标 3，能够阅读前沿的软件框架使用文档及技术文档，了解能够解决复杂软件工程项目的前沿软件框架。

学生针对工程认证课程目标 4 能够使用流行的软件工具实施软件工程项目，并能进行系统的测试和原型模拟。

学生针对工程认证课程目标 5 能掌握软件工程项目有关成本控制、质量管理两个方面的管理方法。

学生针对工程认证课程目标 6 能掌握基本软件项目管理能力，能按照软件生命周期进行项目管理，并进行软件估算和决策。

课程中突出过程性考核，设计了多个考核节点，每个节点均配有相关考核的目标和评价标准，要求每个同学结合项目进展进行汇报。让学生能按流程进行整个项目的实施，课程通过实例讲解、问题讨论、汇报交流让学生综合利用所学的专业知识，全面地了解软件工程相关国际和国家标准，体会企业软件开发与学习过程中特点，培养软件开发的技能。

3.2 大模型驱动的软件工程全生命周期智能化改进

利用大模型进行软件工程教学实践教学内容升级，利用 AI 重构知识体系。在基础技能层的编程教学，减少语法记忆，强调“自然语言→代码”的转换能力（如用 ChatGPT 解释算法逻辑）。增加“提示工程（Prompt Engineering）”教学，指导学生如何有效与 AI 交互。在软件设计层次新增“AI 生成设计模式”案例（如让 AI 生成微服务架构，学生分析其合理性）。在工程实践层，AI 增强的开发流程，在需求分析阶段

利用 AI 辅助从用户故事生成用例图。在测试阶段利用 AI 自动生成测试用例并评估覆盖率。在运维阶段利用 AI 日志分析工具（如 Elasticsearch ML）定位线上故障。在前沿技术层利用 AI 原生软件开发，教授基于大模型的应用开发（如 LangChain 构建智能代理）。探索低代码/无代码平台（如 Appian、OutSystems）与 AI 的结合。本课程结合领域知识库和实时反馈机制的定制化大模型，进一步释放软件工程的创新潜力。本课程具体聚焦社会热点问题，通过技术深度、产业广度和创新思维的多维度融合，培养能解决复杂工程问题、具备商业敏感性的技术创新人才。本课程的核心定位不再是单纯的技术课或创业课，而是“真实问题驱动”的全栈能力训练场。本课程将大模型深度参与软件工程的关键阶段：需求分析与规划、设计与架构、代码开发与生成、测试与验证、部署与运维、维护与演化。通过大模型的智能化辅助，传统软件工程流程（如瀑布模型、敏捷开发、DevOps）得以优化，形成大模型增强的软件工程（LLM-Augmented Software Engineering），见图 1。

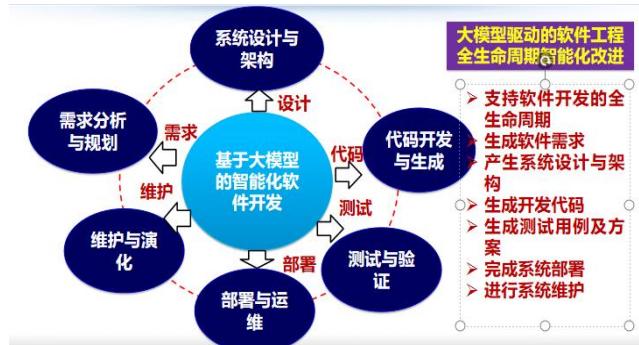


图 1 大模型驱动的软件工程全生命周期智能化改进

需求分析与规划（Requirements Engineering）：智能需求提取从自然语言文档（如用户故事、会议记录）中自动提取结构化需求。需求冲突检测利用大模型分析需求一致性，识别潜在矛盾。需求优先级排序基于历史数据和业务目标生成优化建议。比如用户输入：“我们需要一个支持多人协作的体检审核模型，需要增加版本控制功能。”大模型输出包括核心功能和非功能需求。核心功能包括实时协作、版本历史和权限管理。非功能需求包括低延迟（<200ms）和高并发支持（1 万+用户）。

系统设计与架构（Design & Architecture）：架构模式推荐根据需求自动推荐微服务、单体或事件驱动架构。API 设计生成从需求描述生成 OpenAPI 规范或 gRPC 接口定义。数据库建模将自然语言描述转换为 ER 图或 SQL Schema。比如用户输入：“设计一个医学图像处理模块，需支持高并发图像识别。”大模型输出：架构为事件驱动（Kafka）+ 无状态服务（Kubernetes）。数据库为分库分表

(OrderDB_shard_1, OrderDB_shard_2)。

代码开发与生成 (Coding & Generation) : 代码自动补全 (如 GitHub Copilot) 根据上下文生成函数、类或单元测试。跨语言转换将 Python 逻辑转换为等效的 Java/C++ 代码。代码优化识别冗余代码并建议重构 (如循环展开、缓存优化)。比如用户输入注释：“用 Unet 实现眼科医学图像的异常血管识别”，大项目输出具体的代码。

测试与验证 (Testing & Verification) : 测试用例生成根据代码逻辑自动生成边界值、异常场景测试。缺陷预测分析代码历史数据，定位潜在 Bug 高发模块。日志分析从运维日志中提取异常模式并关联代码缺陷。比如用户输入 java 代码片段：public int divide(int a, int b) { return a / b; }。大模型建议测试用例：divide(10, 2) → 正常; divide(10, 0) → 应抛出 ArithmeticException。

部署与运维 (Deployment & DevOps) : CI/CD 流水线优化自动生成 Jenkins/GitLab CI 脚本。异常根因分析通过日志和指标快速定位服务故障。资源调度建议基于负载预测调整 Kubernetes 副本数。比如用户输入：“部署一个高可用的 Redis 集群，要求自动故障转移。”大项目输出：工具 Redis Sentinel + Kubernetes StatefulSet，监控指标为 redis_connected_clients、redis_memory_usage。

维护与演化 (Maintenance & Evolution) : 技术债评估识别代码中的坏味道 (如重复代码、过长方法)。依赖升级建议分析第三方库漏洞并推荐安全版本。迁移辅助：指导从 Monolith 迁移到 Microservices。比如用户输入：“如何将 Spring Boot 2.x 升级到 3.x？”大模型输出：变更清单为 Jakarta EE 9+、Java 17+，并建议兼容性检查工具：spring-boot-migration-plugin。

3.3 AI 代码助手改进教学实践

大模型正在成为软件工程的“智能副驾驶”，通过全生命周期的渗透，显著提升开发效率、降低人力成本，并推动软件工程向自动化、智能化、可解释化方向发展。本课程通过引入 AI 辅助代码开发助手成为学生的“结对编程伙伴”，通过实时理解学生的需求和意图，提供精准代码建议和生成服务。在开发过程中，帮助学生快速实现想法，减少重复劳动，让学生更专注于核心逻辑和创新。提供高质量代码片段和建议，显著提升编码效率，减少开发时间。通过 AI 代码助手指导学生生成的代码遵循良好编程规范，有助于提高学生实践项目的代码质量，通过减少潜在错误和漏洞，提升实践系统中软件整体质量和可靠性。针对学校的实践项目特色，从项目初期原型开发阶段到后

期代码优化和维护阶段，本课程都使用 AI 辅助开发助手适用于不同的学生开发项目，对课程中个人开发者小型项目提供有效支持。

本课程引入 AI 代码助手如 GitHub Copilot，Cursor 等帮助同学进行开发，基于自然语言生成代码，理解注释或代码片段，快速生成相应代码，提高开发效率。与传统工具基于代码库模式匹配不同，GitHub Copilot 借助大模型深度学习，进行逻辑推理和复杂代码生成。它不仅提供代码补全，还能生成完整函数、类甚至模块，可根据自然语言描述生成代码，超越传统工具能力范围。通过深度集成 AI 代码助手和多种流行 IDE，如 VS Code、JetBrains 等，学生科享受强大代码生成和建议功能，实时获取指导、提示、故障排除、补救措施以及针对特定编码挑战的解决方案。通过课程实践发现，使用 AI 编程助手后学生编写和审查的代码质量全面提高。课程教学中通过使用 AI 代码助手帮助学生快速生成代码，减少重复劳动，让开发者专注核心功能开发和创新，提高开发效率和项目成功率。

3.4 面向工程认证的软件工程创新实践教学实施情况

如图 2，课程通过以项目为实施形式，学生小组分别针对构思、设计、实现进行创新。体现学生的创新实践能力。大作业体现四大实践导向：

- (1) 以市场需求驱动的功能筛选机制；
- (2) 基于用户反馈的持续优化路径；
- (3) 最低资源投入的验证模式；
- (4) 可扩展的迭代开发框架。软件工程创新期末考核为大作业+汇报的方式。

项目选题原则是具有一定的真实性和可行性，来自企业未解决的痛点 (如基于大语言模型的医疗辅诊系统)，前沿性覆盖技术热点 (大语言模型、国产软件、自动化编程工具等)。按软件工程人才培养标准，学生基于最简可行产品 (Minimum Viable Product, MVP)，结合软件工程相关领域的前沿技术和社会热点问题进行软件工程工程中相关流程的实践。小组题目涉及大语言模型、知识图谱、深度学习模型、微服务、软件安全等，完成基于大语言模型的医疗辅诊系统、疾病多分类语义分割的探究与优化、蛋白质相互作用等项目。通过实践，学生能学习和掌握软件工程的创新实践工具和标准，以组为单位，按照软件过程和软件生命周期各阶段的定义和内容，进行项目实践。学生结合社会热点，学习和使用主流的软件工程技术和工具，在项目中融合微服务和软件安全等软件创新的概念，针对项目中的重要问题进行求解。课程引导学

生针对用户痛点分析与竞品格局研判，在设计过程中完成核心功能价值排序与优先级划分，使用最小可用原型的快速构建系统。最终完成的系统实现目标用户

群体的有效性测试，能根据反馈进行系统的迭代方案改进。



图 2 软件工程创新实践教学流程

整个课程教学体现出专业的培养特色和学生能力的有效提升，采用多维评估指标，对学生以团队完成的系统的技术先进性、创新价值、工程规范、文档写作和团队协作等进行指导。针对学生技术栈差异大的问题，教学中进行分组时技术互补，更好地培养学生的团队合作能力，实现优势互补。在教学中模拟企业需求变更频繁的情况，让学生更好地完成技术选型和风险控制，从而更好地完成实际情况模拟和实践。有效地选择企业的痛点问题，提升学生服务社会的能力，培养学生的爱国情怀。课程教学中强调自主和终身学习的必要性，培养学生拥有终身学习和批判性意识和能力，能及时跟踪和学习软件工程领域的理论、技术及国际前沿动态，适应新技术变革。使得学生能在软件产业及社会发展的宏观背景下，结合批判性思维方法，深刻理解技术变革对软件工程领域和社会发展的广泛影响。课程主要培养学生的社会责任感，结合中国国情和时代需求进行项目选择、设计和实现。课程尤其注意培养学生的工程报国、为民造福的意识，树立社会主义核心价值观。使学生理解软件工程专业的职业性质与工程伦理要求，能在软件工程专业实践中自觉遵守职业道德、行业规范和相关法律，履行作为工程技术人员应承担的责任。

3.5 大模型驱动的课程评价改革

大模型驱动的课程评价改革覆盖效率、质量、创新、合规等多维度，形成“可量化、可追溯、可优化”的闭环体系。评价体系包括分层评估，从微观层面评价单次任务（如生成一个 API）的准确率、耗时。从宏观层面评价整个项目周期的综合效益（如成本节约、交付周期缩短）。本课程的总成绩=平时（20%）+课后

实验 20%）+期中（20%）+期末大作业汇报（40%），根据技术演进迭代评价标准。课程大作业要求学生以开发小组形式协作完成，并建立阶段评审机制和交易模式，使学生亲身体验软件开发过程、管理和商业模式的重要性。强调面向培养学生在计算环境下解决问题的能力，全面改造传统课程的评价体制。从软件需求出发，逐步建设相关的软硬件环境，将开发流程、文档模板、评审标准融合在一起，改变学生以往突出个人技巧的杂乱过程。随着大模型在软件工程中的广泛应用，传统的软件评价体系需要相应调整，以更科学地衡量 AI 增强开发的效率、质量和创新性。评价目标的改革主要包括以下六个关键部分：

（1）开发效率

量化大模型对开发速度的提升效果。使用代码生成率评价大模型生成的代码占总代码量的比例；使用需求到代码的转化时间评价从需求描述到可运行代码的平均时间缩短比例；使用重复劳动减少率评价如自动生成文档、测试用例等任务的节省时间。

（2）代码质量

确保大模型辅助下学生生成的代码符合工程标准。进行静态分析缺陷率分析，利用 SonarQube 等工具检测生成的代码缺陷密度（如每千行代码的 Bug 数）。通过测试覆盖率评价 AI 生成代码的单元测试覆盖率。通过可维护性评分分析，基于代码复杂度（圈复杂度）、重复率等指标进行评价。分析安全漏洞数，通过 OWASP Top 10 等工具检查生成代码的安全性，比较大模型生成的代码初始缺陷率和人工优化后的缺陷率。

(3) 需求符合度

验证大模型对需求的理解和实现准确性。评价需求追溯覆盖率,分析生成的代码/设计能追溯到的原始需求条目比例。通过需求变更响应速度评价需求变更后的代码调整效率。

(4) 创新性

评估技术方案创新性,结合毕业要求评价学生自主学习和应用新技术和工具求解实际问题的能力,利用新颖设计提案数评价非传统架构或算法方案占比。将学生的竞赛项目/软著/专利/论文等产出加入评价指标。

(5) 协作与知识管理

衡量团队协作和知识沉淀。利用知识库利用率评价从历史代码/文档中提取知识的频率。评价跨团队协作效率,研究软件系统接口设计一致性。评价新人上手速度,通过AI生成的代码解释文档缩短培训时间

表 2 引入大模型后课程达成度表

课程教学目标	2024 达成度	
课程目标 1: 针对项目的特性, 通过各种资源的查阅、研究与分析, 考虑完成创新项目的多重方案, 设计实现项目的解决思路和方法	0.80	0.81
课程目标 2: 能针对项目特性, 在项目开发和设计中应用新技术或新知识	0.81	0.83
课程目标 3: 设计新应用同时进行新技术的学习, 并将其应用到项目开发与设计中	0.81	0.84
课程目标 4: 能够根据基于之前掌握的实现系统的一系列知识、工具和方法, 通过查阅相关资料进行研究, 调研解决项目的思路和方法	0.81	0.82
课程目标 5: 能针对项目特性, 对现有工具进行比较, 选择适当的工具进行项目的开发, 并对具体工程问题进行分析	0.82	0.85
课程目标 6: 能够承担责任人、合作者或团队成员的任务和职责	0.82	0.83

(6) 伦理与合规

确保生成的代码符合伦理和法律要求。针对版权合规率评价生成代码与开源许可证(如 GPL、MIT)的冲突检测。结合数据隐私保护检查AI处理需求时是否泄露敏感信息,进行合规检查。通过偏见检测代码逻辑中是否存在歧视性规则。让学生了解网络安全等级保护测评等相关知识,更好地满足实践需求。

通过课程评价可以引导学生能以工程化的理论、方法和技术为指导开发软件系统,完成软件工程工具与环境实践、软件工程标准化实践、软件开发绘图工具实践、软件分析与建模实践、软件自动化测试实践、软件项目管理实践、软件配置管理实践等,促进小组内学生能有效地进行沟通和交流,体现团队合作和交流能力,最后按要求提交完成的项目和规范完整文档。本课程结束时,团队开发的项目代码量明显增加,学生团队从以前只能做小项目转型为中型项目。基于本课程,申报获得云南省一流课程“创新设计”、云南大学教育教学改革研究重点项目“新工科背景下‘人工智能+’复合工程型人才培养模式改革与创新”

(2023Z06)。通过调研,本课程学生团队代码编写速度平均提升35%,学生借助大模型生成的代码注释增加70%,项目熟悉时间平均从1周减少至3天,软件项目中100%融合人工智能技术。近两年,软件工程系的就业率达到91.75%和96.30%。从工程认证的达成度可以看出引入大模型后各项指标均有提升,尤其是新技术和工具的学习使用。

4 结束语

人才培养目标导向是工程教育与新工科建设的基本原则。“软件工程创新实践”作为软件工程的重要实践课程,达到相关软件工专业培养体系的要求。课程以培养具有可持续竞争力的创新型人才为目标进行了课程改革,为学生搭建平台、创造机会和提供实践教育。在产出导向(OBE)理念的指导下,进行应用型软件工程人才实践能力培养体系的实践。从应用型软件工程实践能力目标出发,通过结合毕业要求和专业技能,将培养任务落实到实践课程中。考虑到人工智能技术的迅速发展对应用型软件工程人才实践能力培养的影响,在“软件工程创新实践”课程实践内容中融合人工智能技术进行应用软件开发。为了保证培养质量,构建了整体的实践能力评价。

参 考 文 献

- [1] 邵波, 史金飞, 郑峰, 等. 新工科背景下应用型本科人才培养模式创新[J]. 高等工程教育研究, 2023(2): 25-31.
- [2] 毛新军, 孙艳春, 褚华, 等.“101计划”软件工程课程建设思想及成果[J]. 计算机教育, 2023(11): 29-33.
- [3] 王雅娣, 谢玉琳, 郭小丁, 等. 数字化转型背景下行业大数据应用课程教学改革[J]. 计算机技术与教育学报, 2025 2(13): 35-39.
- [4] 郁正良, 禹朴勇, 朱向彩, 等. 新工科背景下软件工程专业个性化人才培养模式探索与实践[J]. 高教学刊, 2020(25): 158-164.
- [5] 吴怀广, 王晓, 金松河, 等. 融合软件工程思想的Java程序设计实验教学改革与探索[J]. 计算机技术与教育学报, 2025 2(13): 79-83.