AI 辅助开发过程控制的信息安全管理课程教学案例研究*

郭燕慧** 邹仕洪 徐国胜 陆天波

北京邮电大学网络空间安全学院 北京 100876

摘 要 针对 AI 辅助开发场景下《信息安全管理》课程缺少面向开发过程的安全控制案例及相应实践训练的问题，本文以用户认证与权限管理系统为载体，设计“风险—控制—证据”导向的实践教学案例，并围绕权限需求梳理、Prompt 安全约束表达、代码生成与多轮修改、安全审查和测试验证设置连续任务。对纳入分析的课程实施材料进行比较发现，与初始提交相比，经反馈修订后，Prompt 记录中明确表达与任务相关安全约束的比例由 10%提高至 85%，安全审查记录中给出具体问题及判断依据的比例由 10%提高至 70%，包含较完整测试与修复证据的验证材料比例由 0%提高至 60%。该案例面向 AI 辅助开发场景下的安全管理需求，以信息系统开发过程承载风险识别、控制设计、过程监督和审查验证等管理活动，为《信息安全管理》课程提供了一个具体、可操作的课堂实践载体。

关键词 信息安全管理，AI 辅助开发，过程控制，实践教学案例，认证授权

A Teaching Case Study on AI-Assisted Development Process Control in the Information Security Management Course*

Guo Yanhui** Zou Shihong Xu Guosheng Lu Tianbo

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
yhguo@bupt.edu.cn zoush@bupt.edu.cn guoshengxu@bupt.edu.cn lutb@bupt.edu.cn

Abstract—To address the lack of development-process-oriented security control cases and corresponding practical training in the Information Security Management course under AI-assisted development scenarios, this paper designs a practical teaching case guided by the "risk-control-evidence" approach, using a user authentication and access control system as the case context. A sequence of tasks is organized around access-control requirement analysis, security constraint specification in prompts, code generation and iterative modification, security review, and testing and verification. Comparison of the course implementation materials included in the analysis shows that, after revision based on feedback, the proportion of prompt records explicitly specifying task-related security constraints increased from 10% to 85%, the proportion of security review records identifying specific issues with supporting rationale increased from 10% to 70%, and the proportion of verification materials containing relatively complete testing and remediation evidence increased from 0% to 60%. The case addresses security management needs in AI-assisted development and uses the information system development process as a setting for risk identification, control design, process supervision, review, and verification, thereby providing the Information Security Management course with a concrete and operable setting for classroom practice.

Keywords—information security management, AI-assisted development, process control, practical teaching case, authentication and authorization

1 引 言

《信息安全管理》课程关注信息系统建设与运行中的风险识别、控制设计、过程监督和审计验证，强调从管理视角理解安全要求如何落实到信息系统生命周期的各个环节。相较于风险评估、制度规范和合规要求等知识的课堂讲授，实践教学还需要依托具体的系统建设场景，使学生能够识别风险产生的环节，理解控制措施如何落实，并形成可供检查和验证的过程证据。随着AI辅助开发工具逐步应用于软件开发[1]，Prompt表达、多轮交互和生成结果修改等活动开始进入开发过程。开发者可以通过自然语言Prompt描述功能需求和安全约束，并根据生成结果持续调整代码。输入校验、权限控制、敏感信息保护

*基金资助：本文得到北京市高等教育学会面上课题（MS2025178）、北京邮电大学本科教育教学改革项目（2026YB29）、北京邮电大学本科品牌课程“智能终端安全与实践”建设项目、北京邮电大学研究生课程建设项目（2025KCQ003）立项资助。

**通讯作者：郭燕慧 yhguo@bupt.edu.cn 。

等既有安全要求并未改变，但其落实过程中增加了新的关注内容。安全要求需要在Prompt中得到明确表达，并在多轮交互和代码修改中持续保持，生成结果也需要经过必要的审查和验证[2]。因此，AI辅助开发场景下的安全管理还需要将Prompt表达、多轮交互和生成结果处理等内容纳入过程检查，并保留相应的过程材料。

在安全类课程实践教学中，国内高校已围绕网络安全与信息安全管理课程开展案例教学，通过安全事件和管理情境组织教学，使风险识别、控制措施等较为抽象的内容能够在具体情境中加以分析[3-4]。随着生成式人工智能和智能编程工具的发展，国内高校已在信息安全技术、软件工程和计算机专业课程中开展教学改革，分别从AI赋能课程教学、生成式编程实践以及学生专业能力培养等方面进行了探索[5-8]。在AI辅助编程方面，相关研究还关注了AI工具在软件开发教育中的应用、学生的AI辅助编程行为以及Prompt任务设计等具体问题[9-11]。对于《信息安全管理》课程，Prompt表达、多轮交互和代码修改等环节直接关系到安全要求能否持续落实，以及生成结果是否经过必要的审查和验证。这些具体开发环节为风险识别、控制设计、过程监督和审查验证等管理活动提供了明确的实践对象；Prompt记录、代码变更、审查意见和验证结果则为课堂中的检查和追踪提供了过程依据。但如何在AI辅助开发过程中组织上述管理活动，形成连续的安全管理实践，在现有教学研究中尚未得到充分讨论。

本文以用户认证与权限管理系统为载体，设计面向 AI 辅助开发过程控制的《信息安全管理》课程实践教学案例。案例选取认证授权中的典型安全问题作为实践情境，以“风险—控制—证据”为组织线索，将权限需求梳理、Prompt 安全约束表达、代码生成与多轮修改、安全审查和测试验证组织为连续实践任务，使学生在具体开发过程中开展相应的安全管理活动。围绕该案例，本文分析 AI 辅助开发过程中的主要安全管理问题，说明案例框架和任务设计，并结合课程实施材料分析学生的过程表现。

2 教学案例设计背景与问题分析

2．1 AI 辅助开发中的安全控制过程

在规范化的信息系统开发中，安全要求通常伴随需求、设计、编码、审查和测试等环节逐步落实。需求文档、设计说明、代码提交记录和测试报告等材料既记录开发过程，也为核查安全要求、追踪问题整改和开展审计验证提供依据。相关软件安全开发框架同样强调，应将安全实践纳入软件开发生命周期，而不是留到系统完成后集中补救[12]。

AI 辅助开发引入后，安全要求的落实过程中增加了 Prompt 表达、多轮交互以及生成结果选择和修改等内容。开发者通过 Prompt 向模型描述功能需求和安全约束，并根据生成结果不断调整代码。在这一过程中，前期已经明确的权限边界、输入校验、敏感信息处理等要求，可能随着任务调整、上下文变化或代码重新生成而被弱化或遗漏。这使 Prompt 表达、多轮交互以及生成结果的选择和修改成为安全要求落实过程中不可忽视的环节。在这些环节中，需要持续检查既有安全要求是否得到保持，对生成结果进行必要的审查和验证，并通过 Prompt 记录、代码变更、审查意见和验证结果等材料保留相应的过程依据。

2．2 AI 辅助开发过程中的安全管理问题

前述分析表明，AI辅助开发中的安全要求需要贯穿Prompt表达、多轮交互、代码修改和结果审查等环节。基于上述过程特点和案例教学需要，本文归纳以下几类安全管理问题。

（1）开发输入约束不足。在 AI 辅助开发中，Prompt 是开发者向模型传递功能需求和安全约束的重要方式。如果输入主要描述功能目标，而对权限边界、输入校验、敏感信息处理等安全要求表达不足，生成结果可能无法完整体现既定安全要求。

（2）交互过程中的控制漂移。在 AI 辅助开发过程中，功能增加、接口调整和代码修改往往通过多轮交互完成。随着任务重点、上下文或代码结构发生变化，前期已经明确的安全要求可能在后续修改中被弱化或遗漏。本文将这种安全要求未能在连续交互和修改中得到保持，进而使实现结果偏离原有控制要求的现象称为“控制漂移”。

（3）生成结果未经审查采纳。AI 生成或修改的代码即使能够实现预期功能，也不能据此判断相关安全要求已经得到落实。生成结果仍需结合既定要求进行代码审查和必要的安全测试，以确认其中是否存在安全控制缺失或实现偏差。

（4）过程证据缺失 。AI辅助开发包含Prompt编写、多轮交互和人工修改等活动，其中部分安全判断和调整不会直接体现在最终代码中。如果缺少Prompt记录、代码版本差异、审查意见和测试结果，后续难以追踪安全要求的提出和调整，也难以判断发现的问题是否完成修正和验证。

2．3 用户认证与权限管理系统作为案例载体的依据

课程实践需要将AI辅助开发过程中的安全管理问题落实到具体的开发任务中。案例载体既要能够支持学生开展安全要求表达、持续检查和审查验证，也要适应课程周期和任务难度，并便于将相关开发

活动转化为可检查、可追踪的过程材料。基于这些教学要求，本文选择用户认证与权限管理系统作为实践案例载体。

用户认证与权限管理系统可围绕注册、登录、身份认证、角色管理和基本访问控制等功能组织开发任务，不必扩展到复杂身份治理、单点登录或细粒度权限策略。这样的任务范围既能够形成较完整的开发过程，又可避免将过多实践时间用于复杂业务逻辑和环境配置，使教学重点集中于安全要求在开发过程中的落实。

该系统涉及输入校验、密码存储、Token处理、角色权限判断和异常信息反馈等安全内容，相关安全要求较为具体。学生可围绕这些要求，在功能开发过程中识别问题、明确开发约束，并通过代码审查和测试验证其落实情况。

同时，该系统中角色、接口和权限之间的关系相对清晰，前期提出的安全要求较容易与后续代码变更、审查结果和验证结果建立对应关系。接口清单、权限矩阵、Prompt记录、代码版本以及审查和测试材料可以记录相关要求在开发过程中的变化，为后续分析安全要求的落实、调整和验证过程提供依据。

3 教学案例框架与任务设计

3．1 案例设计的课程目标定位

案例实践围绕用户认证与权限管理系统的AI辅助开发过程展开。学生在完成注册登录、角色管理和访问控制等功能的过程中，需要识别其中的安全风险，明确相应的安全要求，并结合 Prompt 记录、代码变更、安全审查记录和测试结果检查这些要求的落实情况。案例教学目标分为知识、能力和素养三个层面，如表 1 所示。

表 1 案例教学目标

目标类型	目标内容
知识目标	理解 AI 辅助开发中安全要求贯穿需求表达、多轮交互、代码生成与修改、审查验证等环节的特点，掌握认证授权、输入校验、敏感信息保护和异常信息处理等基本安全要求。
能力目标	能够识别用户认证与权限管理系统中的典型安全问题，将安全要求转化为开发约束，并结合 Prompt 记录、代码审查和测试结果检查安全要求的落实情况及问题修正结果。
素养目标	形成在 AI 辅助开发过程中持续维护安全要求、主动审查生成结果并保留关键过程证据的意识。

3．2 “风险—控制—证据”的任务组织方式

教学目标确定后，案例进一步围绕AI辅助开发过程组织实践内容。对于需求梳理、Prompt表达、代码生成与修改、多轮修改、安全审查和测试验证等环节，先明确其中可能出现的安全问题，再提出相应的过程控制要求，并通过过程材料记录和检查控制要求的落实与验证情况，由此形成“风险—控制—证据”的对应关系，具体如表2所示。

表 2 AI 辅助开发各过程环节的风险—控制—证据对应关系

过程环节	主要风险	过程控制要求	过程材料
需求梳理	角色和权限边界不清，敏感数据识别不足	明确用户角色、接口权限、敏感数据和异常处理要求	需求说明、接口清单、权限矩阵、敏感数据说明
Prompt表达	Prompt偏重功能实现，安全约束表达不足	在Prompt中明确功能要求及相应安全约束	Prompt记录、修改说明
代码生成与修改	生成或修改后的代码可能存在安全控制缺失	检查生成和修改代码中关键安全要求的实现情况，并对发现的问题进行修正	代码版本、问题标注、人工修改说明
多轮修改	前期安全要求在后续交互中被弱化或遗漏	持续检查认证授权、输入校验等安全要求是否得到保留	关键交互记录、约束说明、代码版本差异
安全审查	功能能够运行，但部分安全要求未得到落实	围绕认证授权、密码存储、Token处理、异常反馈等进行审查	安全审查报告、审查清单
测试验证	关键安全要求及问题修正结果缺少充分验证依据	设计测试用例并记录修正前后的结果	测试报告、修复说明、关键日志或截图

3．3 用户认证与权限管理系统的阶段任务设计

具体教学中，案例按用户认证与权限管理系统的开发顺序安排五个阶段任务，并在各任务中纳入表2相应的安全问题、控制要求和过程材料。五个任务分别为权限需求梳理与系统边界确定、注册与登录功能开发、角色管理与访问控制、多轮修改中的安全约束维护以及认证授权安全审查与验证。其中，Prompt 表达和代码生成贯穿功能开发任务，安全审查与测试验证集中在系统初步完成后开展。

（1）权限需求梳理与系统边界确定

首先梳理系统基本需求，明确用户角色、接口范围、访问权限和敏感数据类型，对密码、Token、用户身份信息等敏感数据进行标注，并说明相应的数据处理和异常反馈要求。在此基础上形成需求说明、接口清单、权限矩阵和敏感数据说明，为后续开发提供约束依据。

（2）注册与登录功能开发

围绕注册和登录功能开展 AI 辅助开发，将输入校验、密码安全存储、登录失败提示和 Token 处理等安全要求转化为具体的 Prompt 约束，引导生成结果同时满足功能和安全要求。代码生成后，对照前期约束检查关键安全要求是否落实，并对不符合要求的实现进行调整。该任务重点训练学生在 AI 辅助开发初始阶段明确表达安全约束。

（3）角色管理与访问控制

在前期角色和权限划分的基础上，进一步实现角色管理和受保护接口的访问控制。开发过程中以权限矩阵为依据检查角色与接口之间的授权关系，重点关注后端权限校验遗漏、普通用户越权访问管理员接口以及新增接口未纳入权限控制等问题，并根据检查结果修正相关实现。该任务重点关注既定权限要求向具体安全控制的落实。

（4）多轮修改中的安全约束维护

随着功能增加、接口调整和代码重构，前期已经明确的安全要求可能在后续修改中被弱化或遗漏。该阶段要求结合关键交互记录和代码版本差异，检查新增接口是否纳入认证控制、权限校验是否继续生效，以及密码安全存储等既有要求是否得到保持，据此识别并修正可能出现的控制漂移。

（5）认证授权安全审查与验证

系统初步完成后，围绕输入校验、密码存储、Token 处理、角色权限和异常信息等内容开展安全审查，并记录发现的问题及判断依据。针对关键安全要求和审查中发现的问题设计测试用例，通过非法输入、越权访问、无效或过期 Token 访问等方式进行验证；问题修正后再次测试，并记录修正前后的结果。

4 教学案例实施与分析

4．1 案例实施情况

本案例在面向信息安全专业三年级本科生开设的《信息安全管理》课程中实施，共有7个小组参与，每组5～6人，共40名学生。学生以小组为单位，围绕“用户认证与权限管理系统”完成AI辅助开发实践。课程提供基础项目框架、接口说明、安全审查清单和过程材料模板，学生可使用Cursor、Claude Code等AI辅助开发工具完成代码生成、修改和调试。在实施过程中，各小组依次开展权限需求梳理、注册与登录功能开发、角色管理与访问控制、多轮修改中的安全约束维护以及安全审查与验证等阶段任务。

为加强对实践过程的记录和反馈，课程使用AI-SCD Platform和GitHub仓库进行任务管理与材料留存。其中，AI-SCD Platform主要用于汇总项目入口、发布任务要求和模板材料，并提供相关规范查询；GitHub仓库主要用于保存代码版本、Prompt记录、安全扫描报告、Pull Request评论、测试验证材料和修复前后对比说明。教师结合阶段任务对学生提交的过程材料进行检查和反馈，引导学生在完成功能开发的同时关注权限边界、安全约束、生成结果审查和修复验证。

各组在实践过程中形成的过程材料完整程度有所差异。5 个小组较完整地保留了 Prompt 交互、代码修改、安全审查和测试验证等主要环节的记录，因此纳入后续分析。相关材料包括 5 个 GitHub 仓库、10 个 Pull Request、20 项 Prompt 任务记录、10 项安全审查记录和 10 项测试验证记录。Prompt 任务记录、安全审查记录和测试验证记录均包含初始版与修订版，另有 11 份修复前后对比材料作为验证过程的补充。

4．2 学生过程表现分析

基于纳入分析的过程材料，本文从需求拆解、Prompt安全约束表达、安全要求维护、安全审查、测试验证五个方面观察实践表现。其中，对具有前后版本的任务材料比较初始提交与反馈修改后的修订版，并结合关键交互记录、代码版本和Pull Request评论分析多轮修改中的安全要求保持情况。主要变化及其材料依据如表3所示。

表 3 学生过程表现变化及材料依据

观察维度	初始版表现	修订版表现	材料依据
需求拆解	偏重功能模块划分，安全边界描述不足	补充权限矩阵、接口清单和敏感数据说明	需求文档、接口清单、权限矩阵、敏感数据说明
Prompt 安全约束表达	多为功能性请求，安全约束表达较少	加入密码存储、错误提示、角色校验等安全约束	Prompt 记录
安全要求维护	多轮修改中安全要求可能被遗漏	通过约束文档、关键交互记录和 PR 反馈重新确认安全要求	约束文档、Git 记录、PR 评论
安全审查	主要关注代码能否运行	增加对输入校验、认证授权、Token 处理和错误反馈的检查	安全审查报告、审查清单
测试验证	常以“已修复”“测试通过”概括说明	补充测试输入、预期结果、实际结果和修复前后对比	测试验证报告、修复对比材料

为进一步考察其中与案例核心教学目标直接相关、且能够依据现有材料进行明确判断的内容，本文选取三项指标进行统计：Prompt中是否明确表达与当前任务相关的安全约束；安全审查中是否给出具体安全问题及相应判断依据；验证材料是否同时记录测试输入、预期结果、实际结果以及修复后的验证结果。统计以同一任务的初始版和修订版为对应单位，分别记录满足上述要求的材料数量，结果如表4所示。

表 4 关键过程材料初始版与修订版表现统计

观察内容	初始版	修订版
Prompt 中明确表达与任务相关的安全约束	2/20（10%）	17/20（85%）
安全审查中给出具体问题及判断依据	1/10（10%）	7/10（70%）
验证材料形成较完整的测试与修复证据	0/10（0%）	6/10（60%）

为直观呈现三项指标在初始版与修订版之间的差异，根据表4数据绘制图1。

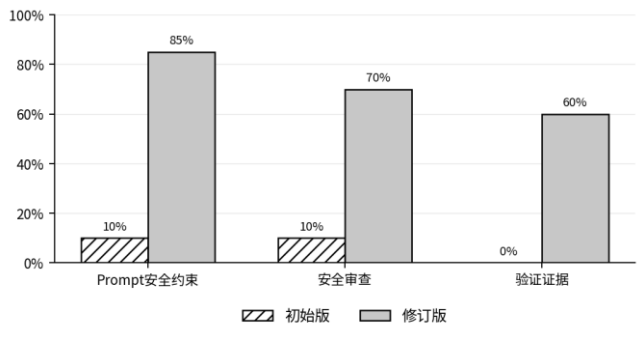


图 1 关键过程材料初始版与修订版表现对比

从需求文档、权限矩阵和 Prompt 记录看，初始材料较多围绕功能模块和开发任务进行拆解，部分材料一次性要求 AI 生成注册、登录、角色管理和权限校验等功能，对用户角色、接口权限和敏感数据边界的说明相对不足。反馈修改后的材料中，角色边界、接口权限、敏感数据和异常处理要求得到进一步明确，Prompt 中也补充了密码不得明文存储、登录失败信息不应暴露过多账户状态、管理员接口需要进行角色校验等安全约束。这些变化与表 4 中 Prompt 安全约束表达比例的提高相一致，说明安全要求在修订材料中的表达更加明确。

从项目约束文档、关键交互记录和代码版本差异看，多轮修改中仍可观察到前期安全要求被弱化或遗漏的情况。例如，已经提出的参数化查询、密码安全存储或角色校验要求，在新增接口、代码重构或切换交互上下文后可能未能继续保持。经过阶段检查和 Pull Request 反馈后，部分小组在项目约束文档中重新确认关键安全要求，并结合交互记录和代码版本差异检查相关控制是否得到保留。这些材料反映了 AI 辅助开发中安全要求在多轮交互和代码修改过程中可能发生变化，交互记录和版本差异也为识别这类变化提供了依据。

安全审查报告、测试验证报告和修复前后对比材料显示，修订后对任务完成情况的记录更加具体。初始材料中较多出现“已修复”“测试通过”等结论性表述，对测试条件、输入、预期结果、实际结果和修复依据记录不足；修订材料中逐渐增加非法输入、越权访问、无效或过期 Token 访问以及接口响应、关键日志等验证内容。以某小组的注册登录模块为例，初始提交仅以接口能够正常调用并返回登录结果作为完成依据，安全审查记录也仅说明“登录功能测试通过”；经过反馈修改后，该组补充了登录失败、非法 Token 访问受保护接口以及普通用户访问管理员接口等测试记录，并提交修复前后的接口响应对比材料。上述小组的前后材料变化与表 4 所呈现的总

体趋势一致，修订后对安全问题判断依据和修复验证过程的记录更加具体。

4．3 实施反思与改进方向

从课程实施材料看，安全审查和验证环节暴露出的不足较为明显。一些 Prompt 记录、安全审查报告和测试验证材料仍停留在“已修复”“测试通过”等结论性描述，对测试输入、预期结果、实际结果以及问题修正依据记录不足。输入校验、密码存储和错误信息返回等较为直观的问题通常能够被发现，而角色权限边界、Token 失效处理和接口越权访问等问题的审查记录相对不足。针对这类问题，现有材料模板和审查提示还需要进一步细化。

本案例以认证授权为主要实践场景，涉及内容集中在应用层安全，对组织制度、合规治理、供应链风险和业务连续性等信息安全管理内容覆盖较少，同时要求学生具备一定的 Web 开发、接口测试和代码阅读基础。7 个小组中有 5 个小组的过程材料能够较完整地支持后续分析，因此本文的观察主要限于这些可连续追踪的材料。

后续教学中，可进一步明确 Prompt 记录、安全审查报告和测试验证报告中的必要记录项，并针对权限边界、Token 处理和越权访问等较难发现的问题补充审查提示和测试示例。Git 提交记录、Pull Request 评论及修复前后对比材料则可继续用于串联问题发现、修改和验证过程。对于认证授权案例难以覆盖的其他信息安全管理内容，可通过补充不同类型的实践任务逐步拓展。

5 结束语

AI 辅助开发并未改变信息系统开发中的基本安全要求，但 Prompt 表达、多轮交互、生成结果选择和代码修改等活动，使安全要求的落实增加了新的过程关注内容。在这一场景下，需要进一步关注安全要求能否在 Prompt 和后续修改中得到明确表达和持续保持，生成结果是否经过必要的审查和验证，以及相关过程是否得到记录和追踪。对于《信息安全管理》课程而言，这些变化也要求实践教学进一步关注安全要求在 AI 辅助开发过程中的表达、保持、审查和验证。

本文以用户认证与权限管理系统为载体，将权限需求梳理、Prompt安全约束表达、代码生成与修改、多轮修改中的安全要求维护、安全审查和测试验证组织为连续实践任务。对课程实施材料的分析表明，反馈修改后的任务材料在安全约束表达、安全审查记录和验证记录方面较初始提交更加完整，同时，部分材料仍反映出多轮修改中安全要求保持不足、安全审查不够深入和验证记录不完整等问题。从课程实践看，风险识别、控制设计、过程监督和审查验证等管理活动可以结合AI辅助开发的具体环节展开，并通过相应的过程记录进行检查和追踪。这一实践目前主要围绕认证授权场景展开，后续还需在更多课程轮次和不同的信息安全管理实践内容中继续观察并完善。

参 考 文 献

[1] Dakhel A M, Majdinasab V, Nikanjam A, Khomh F, Desmarais M C, Jiang Z M J. GitHub Copilot AI pair programmer: Asset or Liability?[J]. Journal of Systems and Software, 2023, 203: 111734

[2] Pearce H, Ahmad B, Tan B, et al. Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions[C]//2022 IEEE Symposium on Security and Privacy (SP). IEEE, 2022: 754-768

[3] 陈红松, 张冬艳, 王小妹. 网络安全与管理课程的案例教学[J]. 计算机教育, 2018(11): 53-56

[4] 郭燕慧, 陆天波, 段蓬勃. 课程思政理念下“信息安全管理”案例教学研究[J]. 计算机技术与教育学报, 2021, 9(2): 97-102

[5] 单棣斌，王楠，李冰，等. 人工智能赋能信息安全技术课程教学改革研究[J]. 计算机技术与教育学报，2025, 13(5):99-104

[6] 李亚坤，颜荣恩，杨波，等. 生成式人工智能背景下高校软件工程课程的教学改革与探索[J]. 计算机技术与教育学报，2024,12(5):8-12

[7] 徐晓飞, 贺祥. 生成式人工智能赋能软件工程教育与学生能力提升[J]. 计算机教育, 2025(8): 1-8

[8] 何思源，卢玲. 生成式编程 AI 辅助计算机专业课程教学的探索与实践[J]. 计算机技术与教育学报，2026,14(2):123-129

[9] Bull C, Kharrufa A. Generative Artificial Intelligence Assistants in Software Development Education: A Vision for Integrating Generative Artificial Intelligence into Educational Practice, Not Instinctively Defending Against It[J]. IEEE Software, 2024, 41(2): 52-59

[10] Sun D, Boudouaia A, Zhu C, et al. Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study[J]. International Journal of Educational Technology in Higher Education, 2024, 21: 14

[11] Denny P, Leinonen J, Prather J, et al. Prompt Problems: A New Programming Exercise for the Generative AI Era[C]//Proceedings of the 55th ACM Technical Symposium on Computer Science Education V.1. New York: ACM, 2024: 296-302

[12] Souppaya M, Scarfone K, Dodson D. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities[R]. Gaithersburg, MD: National Institute of Standards and Technology, 2022