

AIGC 赋能《软件需求分析与设计》教学改革研究

吴春江 甘建红 李志斌 陈廷辉 任李娟

成都信息工程大学软件工程学院, 四川省 610225

摘要 针对 AIGC 推动软件开发活动重心前移和需求表达方式转变的现状, 本文围绕 AIGC 赋能《软件需求分析与设计》教学展开相关研究, 提出了以需求编译为导向的教学内涵和教学目标, 从教学内容重构、教学流程优化和教学评价改进三个维度设计了 AIGC 赋能《软件需求分析与设计》教学改革的整体架构, 并以《校园宿舍维修系统》的需求分析与设计为案例, 对比分析了以文档建模为导向和以需求编译为导向两种教学模式的区别。并且, 本文将教学改革应用于软件工程专业的小班教学中, 课程实施结果显示, 学生在需求理解程度、结构化表达能力、AIGC 输出审查能力和需求追踪完整性四项指标上的平均得分均有较大提升。

关键词 AIGC, 需求编译, 教学内涵, 教学目标, 教学内容, 教学流程, 教学评价

Research on AIGC-Empowered Teaching Reform of Software Requirements Analysis and Design

Wu Chunjiang Gan Jianhong Li Zhibin Chen Tinghui Ren Lijuan

School of Software Engineering Chengdu University of Information Technology,
Sichuan 610225, China;

Abstract—In response to the current trend where AIGC is shifting the focus of software development activities earlier in the lifecycle and transforming how requirements are expressed, this paper explores AIGC-empowered teaching in the Software Requirements Analysis and Design course. We propose a teaching framework centered on 'requirements compilation,' outlining its core connotations and objectives. A comprehensive reform architecture is designed across three dimensions: content restructuring, process optimization, and evaluation improvement. Using the requirements analysis and design of a 'Campus Dormitory Maintenance System' as a case study, we compare two teaching models oriented toward traditional document modeling versus requirements compilation. Furthermore, the reform was implemented in small-class teaching for software engineering majors. The results demonstrate significant improvements in students' average scores across four key metrics: requirement comprehension, structured expression skills, AIGC output review capabilities, and requirement traceability completeness.

Keywords—AIGC, Software Requirements Analysis and Design, Requirement Compilation, Teaching Reform

1 引言

AIGC 工具的普及正在改变软件开发活动的组织形式。传统的软件开发通常按照“需求分析→概要设计→详细设计→软件编码→软件测试”的顺序推进, 而在 AIGC 模式下, 开发者仅需通过自然语言向大模型描述界面、流程、数据规则和功能场景, 即可快速生成页面蓝图、用户故事、接口示例、测试用例乃至代码片段。GitHub Copilot 的相关实验显示, 使用 AIGC 辅助编程工具的开发者, 其完成指定任务的速度显著提高。这种软件开发活动的变化使得软件开发瓶颈不再只是“能不能写出代码”, 而是转向“是否定义了正确的问题、是否准确理解用户、是否能够审查 AIGC 的输出并将结果纳入到工程约束中”。

在 AIGC 模式下, 软件需求分析与设计的重要性愈发凸显。本文依据“需求描述→原型生成→结果验证→迭代修订”的思路, 围绕“AIGC 赋能《软件需求分析与设计》教学”开展研究, 分析 AIGC 给软件工程教

育带来的新变化, 提出以“需求编译”为导向的教学内涵、教学目标、改革内容和实施路径, 并以《校园宿舍维修系统》的需求分析与设计为案例, 对比分析了以文档建模为导向和以需求编译为导向两种教学模式的区别。

2 相关背景

在教育数字化和人工智能应用的驱动下, 教育部等九部门提出了要推动教育理念、教学模式和教育治理整体性变革, 强调培养学生高阶思维、思考判断能力和实践能力^[1]。UNESCO 关于教育和研究中使用 AIGC 的指导文件也提出, 教育领域引入 AIGC 应坚持以人为中心, 重视伦理、隐私、安全和学习者能力发展^[2]。在 AIGC 的推动下, 软件需求工程的重要性被放大, 需求表达质量直接对后续软件开发结果产生重大影响, 软件开发活动的重心已被前移至软件需求分析与设计。同时, 如何利用 AIGC 快速地暴露需求偏差, 更早期地与用户沟通, 更系统性地审查结果, 也对软件需求表达

方式提出了更高的要求。

同时, AIGC 课程实践、生成式 AI 教学过程评价和程序设计课程改革已有学者进行了探讨研究^[3-6]。AIGC 引入课堂后, 教学评价不能只关注任务完成效率, 还应考查学生参与分析与决策的过程, 以及发现和修正生成结果问题的能力。

2.1 AIGC 推动软件开发重心前移

软件需求分析与设计在软件工程全生命周期中具有基础性和连接性, ISO/IEC/IEEE 29148:2018 把需求工程界定为贯穿系统和软件全生命周期的需求形成、表达、验证、管理和变更控制过程^[7]。经典需求工程也强调, 需求分析与设计工作并不是单纯记录用户表达, 而是需要在利益相关者目标、业务流程、约束条件和实现方案之间建立可追踪关系^[8-9]。

在 AIGC 模式下, AIGC 能够在短时间内生成大量候选方案, 包括页面结构、交互文案、数据表字段、接口示例和测试脚本。这种效率的提升对于软件工程教学具有双重意义, 一方面学生可以不再把大量时间消耗在重复的软件编码上, 而可以把更多的精力投入到问题分析、方案比较和结果验证上; 另一方面, 由于 AIGC 的输出具有不稳定性、上下文依赖和可能的事实错误, 这就要求具备较强的需求审查能力^[10-12]。

2.2 AIGC 促进需求表达方式转变

在传统的《软件需求分析与设计》教学中, 需要老师给出的目标和需求足够清晰, 学生只需要根据文档模板完成需求规格说明、用例图、类图和数据库的设计。然而, 在真实的软件工程项目中, 用户的需求往往是口语化、片段化和带有感情色彩的。这些话语不是可以直接编码的需求, 却包含了角色、目标、痛点、业务规则和质量要求。

因此, 在 AIGC 模式下, 软件需求分析与设计的关键应是把自然语言转换为可讨论、可验证、可追踪的工程表述。并且, 这种转换还需要进一步延伸为“将用户的需求转化为 AI 理解的需求”。也就是说, 学生不仅要听懂用户言外之意, 更要能将模糊的用户表达转化为 AIGC 可处理的结构化提示, 包括明确用户角色、业务场景、操作目标、功能边界、约束条件、输出格式和评价标准。例如“维修进度没人告诉我”不能简单转写为“增加通知功能”, 而应进一步澄清为哪些角色需要接收通知、在哪些节点触发什么通知、用户如何确认维修状态、异常情况如何处理。

3 教学内涵与教学目标

在 AIGC 模式下, 只有将用户需求转化为 AI 理解的需求, AIGC 输出的用户故事、页面原型、流程说明

和验收标准才可能真正贴近业务需求。因此, AIGC 赋能《软件需求分析与设计》教学改革不能只把 AIGC 视为生成文档或代码的工具, 而应将其嵌入到需求澄清、需求表达和需求验证的全过程。学生需要在“用户语言-工程需求-AI 可理解需求-可验证原型”之间反复转换, 并在转换过程中发现歧义、补充约束、修正偏差, 这个过程正是“需求编译”的核心所在, 同时也对教学目标提出了新的要求。

3.1 以需求编译为导向的教学内涵

需求编译是一种面向 AIGC 协同开发的教学隐喻和实践流程, 即以自然语言描述的需求为输入, 以需求澄清、场景建模、结构化提示、AIGC 生成、用户反馈和工程化整理为主要环节, 最终形成用户故事、场景流程、原型界面、交互说明、验收标准和需求追踪等结果。

需求编译的核心任务包括两个方向的转换和三类语言的翻译。第一是将用户言语翻译为 AI 可处理的提示词, 即将用户需求转化为 AI 可理解的需求, 把模糊、零散、带有语境依赖的用户表达整理为角色、目标、触发条件、业务规则、数据对象、约束条件、异常场景和优先级等结构化信息。第二是将 AIGC 的输出翻译为工程语言, 即将 AIGC 生成的专业描述、页面建议和流程方案转化为用户看得懂、能判断、可反馈的需求规格、软件用例、领域模型、数据模型、接口约束和验收条件。

3.2 以需求编译为导向的教学目标

在 AIGC 模式下, 以需求编译为导向的教学目标从单一能力扩展为复合能力。第一是问题定义能力, 要求能够识别真正要解决的问题, 而不是机械地接受题目表述。第二是用户沟通能力, 要求能够通过访谈、观察和情境复述澄清需求。第三是提示设计能力, 要求能够将用户需求整理成 AIGC 可处理的结构化输入, 控制输出格式并追问模糊点。第四是原型表达能力, 要求能够用页面草图、流程说明和用户故事, 使用户提前感知系统形态。第五是结果审查能力, 要求能够从需求一致性、可实现性、数据安全、异常处理和用户价值角度评估 AIGC 输出。第六是工程责任能力, 要求能够认识到需求偏差、隐私泄露、算法偏见和代码缺陷都可能影响真实用户。

4 改革内容与实施路径

为应对 AIGC 模式下软件开发重心前移与需求表达方式转变的现实挑战, 基于以需求编译为导向的教学内涵和教学目标, 软件需求分析与设计教学改革应围绕教学内容重构、教学流程优化和教学评价改进三个维度开展系统性设计, 如图 1 所示。

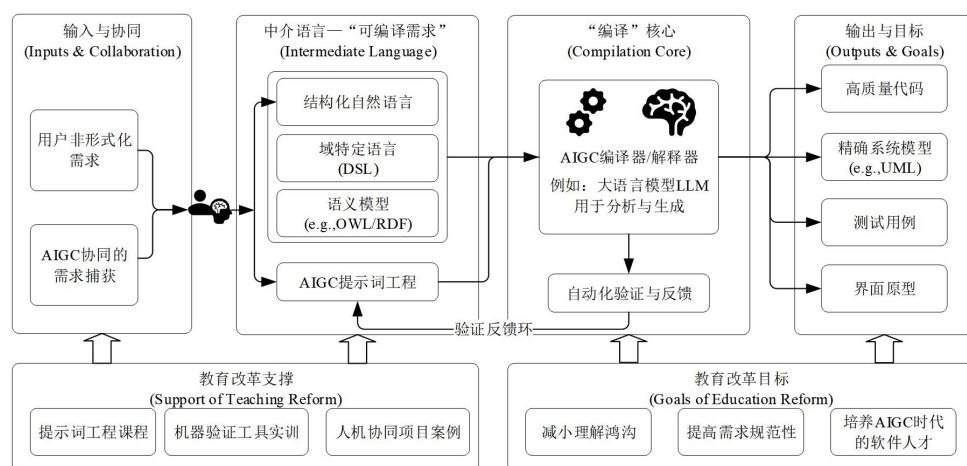


图 1 AIGC 赋能《软件需求分析与设计》教学改革架构图

在教学内容上，课程应突破传统需求文档编写与UML建模的局限，进阶为对痛点问题洞察、跨角色沟通、结构化提示设计、AIGC输出审查及规范化工程表达等复合能力的培养。

在教学流程上，课程应以非形式化需求和AIGC协同需求为起点，依托域特定语言（Domain-Specific Language, DSL）和提示词工程等“中介语言”的表征转换，将模糊、零散的用户话语精准转化为大模型可解析的指令，并在AIGC生成候选方案的基础上，通过机器校验、人工审查与用户反馈交织的“验证反馈环”，最终沉淀出用户故事、测试用例、界面原型等可讨论、可验证、可追踪的工程核心产物。

在评价方式上，课程评价应实现由“结果黑盒”向“过程证据白盒化”的跃迁，从主要关注文档建模与系统实现转向对需求调研、提示词交互、原型多轮修订及技术伦理反思等过程性证据的全息度量。

4.1 以需求编译为导向的教学内容重构

在AIGC模式下，《软件需求分析与设计》的教学目标由“会写文档、会画模型、会做系统”转向“能理解需求、能编译需求、能验证需求、能实现需求”，其教学内容应该在知识、能力和素养三个层次进行相应的重构。

在知识体系后构建基础上，教学内容不仅要涵盖需求获取、需求分析等基础概念与方法，还需要系统性地融入AIGC基本原理、提示词工程范式、领域知识嵌入机制以及AIGC输出可信度评估框架。在能力训练设计上，教学内容重点应聚焦于将模糊用户意图转化为结构化提示词的实操方法，增设基于领域语境对AIGC生成结果进行偏差识别与逻辑校验的实践环节，引导学习者掌握在人机协同中主导需求演进的技巧。在素养培育融合上，教学内容应将技术向善的价值判断与工程伦理作为核心思政内容，结合需求模糊性与技术不确定性交织的现实场景，探讨如何保持对用户

真实诉求的敬畏之心，并系统讲解如何主动识别与规避AIGC可能引发的需求漂移、数据偏见与责任归属模糊等潜在风险。

同时，以需求编译为导向的教学内容重构，并不是对需求工程概述、需求获取方法、UML建模、需求规格说明、系统设计等课程内容的删除，而是将AIGC协同任务嵌入到每个教学模块中。例如在需求获取模块中加入“访谈记录到结构化需求卡片”的训练；在用例建模模块中加入“用AIGC生成用户故事并人工筛选”的训练；在原型设计模块中加入“根据用户故事生成页面草图说明并用原型工具表达”的训练；在需求验证模块中加入“把验收标准转化为用户可判断的测试场景”的训练；在需求管理模块中加入“AIGC输出版本、提示词版本和需求版本的追踪”训练。

4.2 以需求编译为导向的教学流程优化

以需求编译为导向的教学流程优化采用“六步迭代”而非单向的线性流程，即“场景导入→需求理解→需求编译→AIGC生成→反馈编译→工程整理”。

第一步是场景导入，老师提供具有不确定性的真实问题背景，而不是一次性给出完整需求。第二步是需求理解，学生通过角色分析、访谈提纲、痛点梳理和业务流程复述形成原始需求集合。第三步是需求编译，学生把原始材料整理为AIGC可理解的结构化提示，包括角色、目标、上下文、输入输出、约束、异常、优先级和希望生成的成果格式。第四步是AIGC生成，学生获得用户故事、功能清单、流程建议、页面说明和验收标准后，通过人工审查的方式标注“可采纳、需验证、需删除、需补充”的内容。第五步是反馈编译，学生将原型和场景说明展示给真实或模拟用户，收集“是否符合预期、哪里不清楚、是否遗漏、是否过度设计”等反馈，并重新编译需求。第六步是工程整理，学生把已验证的需求转化为需求规格说明、用例模型、领域模型、数据库设计和实现计划。

以需求编译为导向的教学流程采用“由模糊到清晰”的螺旋递进式设计。课程教学不应局限于给出最终题目,而应分阶段提供材料包,包括项目背景文档、原始访谈片段、用户调研摘要、需求讨论会议纪要、业务规则补充说明、冲突需求清单、用例文档、详细需求文档和变更通知等。学生每获得一份材料,就需要更新需求假设、AIGC 提示、原型版本以及需求追踪表。这样的材料组织方式可以模拟真实需求的调研过程,让学生体验软件需求不是一次性“写出来”的,而是在证据不断增加、利益相关者不断反馈、约束不断显现的过程中逐步稳定的。

4.3 以需求编译为导向的教学评价改进

以需求编译为导向的《软件需求分析与设计》课程评估聚焦于学生的编译能力表现,可采用“老师引导、团队协作、AIGC 辅助、用户参与”的混合模式。

老师的角色不再只是讲授概念,而是场景设计者、访谈组织者、提示审查者和质量把关者。学生团队可以设置需求负责人、用户沟通负责人、原型负责人、AIGC 记录负责人、建模负责人和测试负责人,要求每个成员说明自己如何参与需求判断。AIGC 在课堂中应被限定为“候选方案生成器”和“表达辅助器”,其输出不能替代人工需求确认。用户可来自真实的需求方,也可以是同学或老师模拟角色,他们不需要了解专业模型,但必须能基于原型和场景说明清晰表达“这是否是我要的”。

在这种混合模式下,教学评价从“最终代码结果”转向“过程证据与结果质量并重”,课程成绩也可由六部分组成。其中,原始需求调研与角色分析占比 15%,需求编译提示与 AIGC 交互记录占比 15%,用户故事、场景流程和验收标准占比 20%,原型展示与用户反馈修订占比 20%,系统设计与实现成果占比 20%,反思报告与伦理规范占比 10%。并且,在这种混合模式下,与 AIGC 交互并非旨在鼓励学生滥用工具,而是为了评价提出清晰问题、审查输出结果及阐明取舍依据的能力。最终的实现仍然重要,但它不再是唯一标准。需求理解是否准确、原型是否被用户理解、修订是否有证据、验收标准是否可执行,都会成为课程评价的关键。

5 案例分析与模式对比

本文选取《校园宿舍维修系统》作为《软件需求分析与设计》课程教学的案例,对比分析以文档建模为导向和以需求编译为导向的教学区别。校园宿舍维修系统具有用户角色明确、业务流程完整、反馈周期短和原型表达直观等特点,涉及学生、宿舍管理员、后勤维修人员等多类角色,各个角色的需求又相互关联。这个案例看似简单,却包含需求获取、角色冲突、流程优化、非功能需求和数据管理等多个教学点。

5.1 案例分析

在以文档建模为导向的《软件需求分析与设计》教学过程中,教师通常首先讲授需求分析概念、需求规格说明模板、UML 用例图、类图、顺序图、数据库设计和界面设计方法,并给出“开发一个校园宿舍维修系统”的项目任务。然后,学生根据题目拆分功能模块,编写需求文档,画出用例图和数据表,完成软件编码,最后进行系统功能演示。

这一模式的优点是知识结构清晰、进度容易控制、成果便于统一评价。但其局限也较明显,学生容易把需求分析当成“填表”任务,认为只要图画得规范、文档写得完整就完成了需求工作。同时,用户参与不足,许多“进度透明”、“派单准确”、“维修及时”等真实诉求没有被转化为可验证指标。需求偏差往往在系统完成后才暴露,修改成本较高。

在以需求编译为导向的《软件需求分析与设计》教学过程中,教学首先是从口语化、模糊化的原始材料开始的,例如“宿舍灯坏了三天没人来修”、“我不知道报修单到底有没有被看到”。

其次,学生需要提取角色、痛点、任务和约束,然后把这些材料整理为 AIGC 可理解的需求提示,包括项目背景是什么、用户角色有哪些、核心流程包括哪些步骤等,例如“请以结构化格式输出学生端报修的用户故事,并明确列出‘上传损坏图片过大’或‘报修类别错误’等异常场景的验收标准”。同时,学生还必须针对 AIGC 生成的功能描述进行追问,例如学生需要对“进度查询”具体追问“已提交、已派单、维修中、待确认”等核心状态的流转节点,以及触发这些状态变化的前置条件与通知机制。

然后,学生可以直接引导 AIGC 生成响应式的前端页面源码,并在此基础上进行人工代码审核与细节微调,以此作为高保真的交互原型展示给用户。用户在体验后,还可能反馈“希望看到预计上门时间”、“希望能撤销重复报修”等其他功能需求,这些反馈需要再反向修订并重新进行需求编译。

经历这种“编译→生成→审查→反馈→再编译”的多次循环迭代,可以逐步消除理解偏差,最终沉淀出高度贴合真实业务场景的系统原型。这一模式的特点就是将用户的反馈实时转化为可执行的验证项,推动需求从静态文档向动态演进转变。

5.2 模式对比

以文档建模为导向和以需求编译为导向的《软件需求分析与设计》教学模式,并非简单的“有技术”和“无技术”之别,而是教学起点、表达对象、反馈时点和评价逻辑的差异。前者是从知识点和规范文档出发,后者是从用户场景和自然语言需求描述出发;

前者强调专业符号表达,后者强调专业表达与用户可理解表达并重;前者在实现后展示结果,后者在原型阶段就开始组织反馈;前者更多的是评价最终系统实现,后者则对需求证据、AIGC 协同记录、原型修订和系统实现进行过程评价。两者的模式对比如表 1 所示。

表 1 教学模式对比表

比较维度	以文档建模为导向	以需求编译为导向
教学起点	从知识点、模板和教师给定题目出发	从真实场景、用户话语和模糊问题出发
需求表达	以需求规格说明和UML模型为主	用户故事、原型、验收和工程模型并重
AIGC 角色	课外工具,缺少过程记录和审查	候选方案与表达必须记录提示和取舍
反馈时点	完成后集中展示	原型阶段即开展用户反馈和需求修订
评价重点	主要评价文档完整性和系统运行效果	评价需求证据、编译过程、原型质量、用户反馈和实现效果
能力指向	偏重建模规范和功能实现	强调问题定义、用户沟通、AIGC 协同、结果审查和工程责任

6 实施成效

如图 2 所示,以“需求编译”为导向的《软件需求分析与设计》教学改革应用于成都信息工程大学软件工程专业 40 人的小班教学中,有效调动了学生学习和项目实践的积极性和主动性。从课程实施情况来看,学生在需求理解程度、结构化表达能力、AIGC 输出审查和需求追踪完整性四项指标上的平均得分分别由 68.4 分、66.9 分、61.7 分和 64.3 分提高到 84.2 分、83.6 分、80.9 分和 84.7 分;项目组的需求澄清、首轮原型生成和缺陷返工时分别下降 33.3%、40.2%和 40.7%;课程问卷调查显示,90.0%的学生认为结构化需求卡片有助于准确理解用户诉求,87.5%的学生认为 AIGC 辅助原型提高了沟通与开发效率。

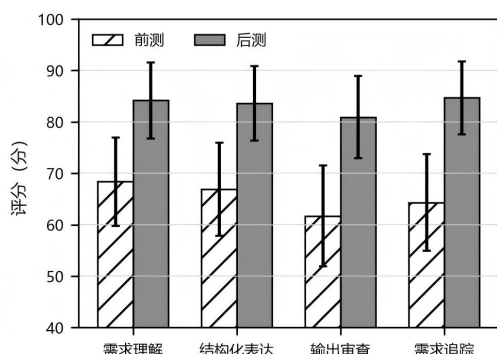


图 2 教学改革前后学生需求能力指标对比

7 结束语

AIGC 正在推动软件开发活动从强调编码实现转向重视需求理解、快速原型、用户验证和结果审查。

《软件需求分析与设计》课程位于软件工程人才培养链条的前端,最适合承接这种变化。本文提出的以需求编译为导向的教学改革,强调的是以用户自然语言需求为起点,通过结构化提示和 AIGC 协同,将用户的需求转为 AI 理解的需求,再把 AIGC 输出转化为用户可理解、可反馈、可验证和可实现的需求表达。

实施成效显著,学生在需求理解、结构化表达、AIGC 输出审查与需求追踪方面均有提升,项目组的需求澄清、原型生成和缺陷返工时相应下降,说明需求编译导向的教学模式能够兼顾需求质量与开发效率。

这一教学模式的变革,绝非走向“去文档化”或削弱传统的模型训练,而是倒逼文档规范与系统模型重新回归到“解决真实用户问题”与“基于证据验证”的逻辑原点。当 AIGC 能够低成本地构建基础文本与图表时,工程表达的考查重心便从单纯的“语法与符号正确”转向了深度的“业务意义对齐”。只有当详尽的需求规格说明、严谨的用例模型、底层的数据库设计乃至最终的系统实现代码能够构建起严密的全链路双向追踪机制,并精准映射回原始的真实用户痛点与多轮敏捷迭代中的反馈证据时,需求分析才不再是一项机械的规范填表任务。

参考文献

- [1] 教育部等九部门. 教育部等九部门关于加快推进教育数字化的意见[EB/OL]. (2025-04-11)[2026-07-28].
- [2] MIAO F, HOLMES W. Guidance for Generative AI in Education and Research[R]. Paris: UNESCO, 2023.
- [3] 杨晓敏, 王宝丽. 基于AIGC的计算机网络原理课程教学实践[J]. 计算机技术与教育学报, 2026, 14(3):62-67.
- [4] 马瑞强, 王迪, 蒋永生, 等. 生成式AI融入课堂的过程证据框架构建与实践改革[J]. 计算机技术与教育学报, 2026, 14(3):156-162.
- [5] 杨雨倩, 陈嘉敏, 周富肯, 等. 生成式AI赋能面向对象程序设计课程教学改革与实践[J]. 计算机技术与教育学报, 2026, 14(3):163-167.
- [6] 刘佳荟, 张思佳, 胡泽元, 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, 13(1):76-81.
- [7] ISO/IEC/IEEE. ISO/IEC/IEEE 29148:2018 Systems and software engineering—Life cycle processes—Requirements engineering[S]. Geneva: ISO, 2018.
- [8] WIEGERS K, BEATTY J. Software Requirements[M]. 3rd ed. Redmond: Microsoft Press, 2013.
- [9] POHL K. Requirements Engineering: Fundamentals, Principles, and Techniques[M]. Berlin: Springer, 2010.
- [10] 郭旦怀, 吴若玲, 卢罡, 等. AIGC在大学计算机教育教学中的有效利用[J]. 计算机教育, 2024(7):35-40.
- [11] 刘小丽, 古天龙. ChatGPT对计算机教育的影响及对策[J]. 计算机教育, 2023(11):38-44.
- [12] 沙行勉, 王寒, 徐珑珊, 等. ChatGPT对计算机基础教育的挑战分析与应对策略[J]. 计算机教育, 2023(11):51-54.