

人工智能时代软件工程教育中的协作与工具实践

刘晓彤*, 郑丹妮

北京信息科技大学计算机学院, 北京

收稿日期: 2025年2月23日; 录用日期: 2025年3月20日; 发布日期: 2025年3月27日

摘要

在人工智能快速发展的时代, 各行业对软件工程能力的需求进一步提升, 团队协作与现代化工具的使用成为关键。高校软件工程教育亟须与时俱进, 引入真实企业项目场景, 培养学生的工程意识和团队协作能力。本研究以北京信息科技大学计算机学院的《软件项目综合实践》课程为案例, 结合陶哲轩教授对人工智能(AI)与协作模式的理论观点, 探索如何通过版本控制工具和AI技术优化教学效果。课程引入Git、Jenkins、Gerrit等工具, 结合自动化评分和代码冲突管理, 激发学生的学习兴趣, 并提升其就业竞争力。研究表明, 通过真实企业场景和现代技术的融合, 课程改革显著增强了学生的学习动力和就业竞争力, 为软件工程教育提供了创新思路。

关键词

软件工程教育, 人工智能, 团队协作, 教学改革

Collaboration and Tools Practice in Software Engineering Education in the Era of Artificial Intelligence

Xiaotong Liu*, Danni Zheng

Computer School of Beijing Information Science and Technology University, Beijing

Received: Feb. 23rd, 2025; accepted: Mar. 20th, 2025; published: Mar. 27th, 2025

Abstract

In the era of rapid advancements in artificial intelligence, the demand for software engineering skills across industries has significantly increased, with team collaboration and the use of modern tools

*通讯作者。

文章引用: 刘晓彤, 郑丹妮. 人工智能时代软件工程教育中的协作与工具实践[J]. 教育进展, 2025, 15(3): 977-982.
DOI: 10.12677/ae.2025.153497

becoming critical. Higher education in software engineering urgently needs to keep pace with these developments by incorporating real-world enterprise project scenarios to cultivate students' engineering mindset and teamwork abilities. This study takes the "Comprehensive Software Project Practice" course offered by the School of Computer Science at Beijing Information Science and Technology University as a case. Combining with Professor Terence Tao's theoretical viewpoints on artificial intelligence (AI) and collaboration models, it explores how to optimize teaching effectiveness through version control tools and AI technologies. The course introduces tools such as Git, Jenkins, and Gerrit, combined with automated grading and code conflict management, to spark students' interest and enhance their employability. The findings reveal that integrating real-world enterprise scenarios with modern technologies significantly boosts students' learning motivation and job market competitiveness, offering innovative approaches to software engineering education.

Keywords

Software Engineering Education, Artificial Intelligence, Team Collaboration, Teaching Reform

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

在人工智能时代, 软件工程作为推动技术创新的重要学科, 经过多年的发展, 已经积累了丰富的经验和大量成熟的工具。这些经验和工具不仅使软件开发过程更加高效, 也为团队协作提供了坚实的基础[1]-[4]。菲尔兹奖得主, 加州大学洛杉矶分校数学系陶哲轩教授在 2024 年底的一次访谈中提出了一个发人深省的观点: 人类未来的研究可能会借鉴软件工程中的协作模式[5]。在人工智能时代, 数学等基础理论研究的发展方向或将从个人英雄主义的模式转向以团队协作为核心的模式。这种转变强调任务的模块化分工和高效工具的使用。这不仅没有削弱传统软件工程学科的重要性, 反而进一步凸显了其价值。因此, 我们应该更系统地向学生讲授如何进行有效协作, 如何使用现代工具, 以及如何理解和应用前人开发的协作算法。然而, 目前软件工程教育在培养学生协作能力和工程意识方面仍存在不足, 尤其是在真实企业项目实践的融入上有所欠缺。

软件工程是一门强调实践的学科, 但传统教学往往聚焦于理论传授, 忽视了学生在团队协作和工程化思维上的培养。为了弥补这一缺陷, 北京信息科技大学计算机学院在《软件项目综合实践》课程中, 引入了企业级工具和流程, 旨在模拟真实的项目开发环境。通过 Git、Jenkins 和 Gerrit 等工具的实践应用, 学生得以在真实场景中学习如何进行版本控制、代码评审以及持续集成。这种课程设计试图缩小高校教育与行业需求之间的差距, 使学生在就业时具备更强的竞争力。软件工程的学科属性决定了实践的重要性。传统课堂教学模式难以满足培养高素质工程师的需求, 因为真实项目中的协作和技术应用往往涉及复杂的流程与多角色分工。学生需要通过直接的项目实践, 了解如何在团队中分工合作, 如何高效使用现代化工具完成任务。本研究探讨 AI 技术和协作工具如何进一步优化教学效果。研究的核心问题包括: 如何通过工具和 AI 技术模拟真实企业场景; 如何设计公平、透明的评分体系激发学生学习兴趣; 如何通过课程改革提升学生的工程实践能力和就业竞争力。

2. 理论基础

现代软件工程的开发和运维过程已经高度依赖于各种协作和自动化工具[6]-[8], 如图 1 所示。这些工

具贯穿开发生命周期的各个环节, 从代码的创建与管理到测试、部署和监控, 全面提升了团队的协作效率和项目的交付质量。

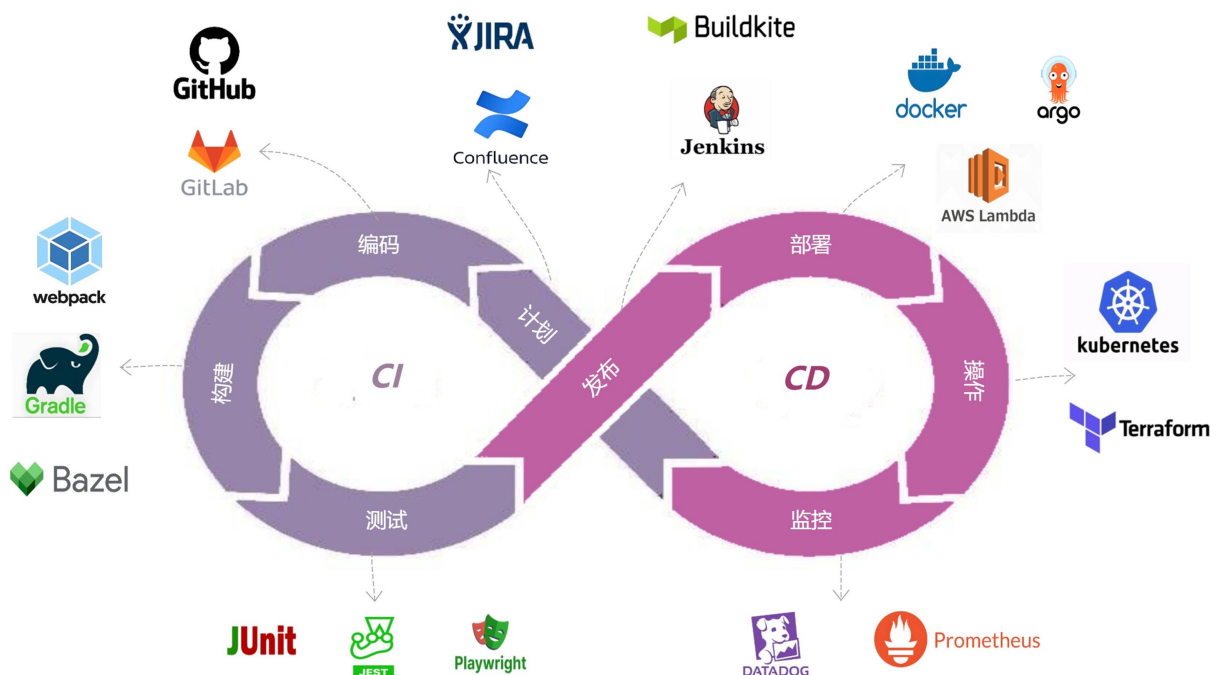


Figure 1. Flow chart of common tools in software engineering and the CI/CD loop process

图 1. 软件工程中常用工具及持续集成/持续交付循环流程图

以下是软件工程中常见工具的简要介绍(对应工具的官网链接详见表 1)。在代码管理和协作阶段, 工具如 GitHub、GitLab 提供了版本控制与代码仓库服务, 帮助团队管理代码历史, 协作开发, 处理代码合并与冲突。同时, JIRA 与 Confluence 等项目管理工具支持团队规划开发任务、记录文档和协同管理。在构建与测试阶段, Gradle 和 Bazel 等构建工具通过定义构建流程, 优化了代码的编译和打包过程。Webpack 作为前端的模块打包工具, 简化了复杂项目的构建。测试工具如 Jest 和 JUnit 能够自动执行单元测试和集成测试, 确保代码的质量和稳定性。在发布阶段, Jenkins、Buildkite 等工具通过自动化管道实现代码的持续集成和交付, 显著减少了手动操作的复杂性和错误率。在部署与运维阶段, Docker 和 Kubernetes 等容器化和编排工具使得应用的环境管理更加灵活和高效。Argo 提供了 GitOps 工作流, 用于自动化部署。Terraform 则通过基础设施即代码(IaC)的方式管理和配置云资源。在监控与反馈阶段, Prometheus 和 Datadog 等监控工具提供了实时的数据分析和警报服务, 帮助团队快速发现并解决潜在问题, 确保系统的可靠性和稳定性。这些工具相辅相成, 形成了一个完整的 CI/CD (Continuous Integration/Continuous Delivery) 闭环, 支持现代软件开发团队在高效协作的同时, 交付高质量的软件产品。

在教育理论方面, 本研究借鉴了基于项目的学习(Project-Based Learning, PBL)理论[9][10]。PBL 强调通过实际项目推动学生对理论知识的掌握, 并通过实践培养学生的创新能力和团队合作精神。结合 AI 技术, PBL 模式可以进一步扩展, 通过实时反馈和个性化建议, 为学生提供多样化的学习体验。例如, AI 可以在学生的项目开发过程中自动评估代码质量, 提供优化建议, 从而帮助学生更高效地完成学习任务。

在 PBL 的框架下, 学习者通过亲身参与项目完成知识的建构过程。这种方式既强调学生的主动性, 也注重培养其批判性思维和协作能力。通过结合陶哲轩的理论, 研究进一步拓展了协作的内涵, 即在现代技术工具的辅助下, 学习者可以更高效地处理复杂问题, 达成目标。此外, AI 工具的应用不仅优化了

PBL 的教学效果，也为团队协作的公平性和效率提供了技术支持。这样的融合为软件工程教育改革提供了坚实的理论基础。协作模式和分布式认知理论为软件工程教育提供了重要的支撑框架。分布式认知认为知识和技能可以通过工具、团队成员以及外部环境共同承载。在这一理论框架下，软件工程团队协作的核心在于如何通过工具有效分配认知负担，从而实现团队的整体效能最大化。引入 AI 作为协作工具，不仅能够为团队成员分担部分任务，还可以为学生提供动态反馈和智能支持，帮助他们在团队中发挥更大的潜力。

Table 1. Table of official website URLs of common tools
表 1. 常见工具官网网址对应表

阶段	工具名称	官网网址
计划(PLAN)	JIRA	https://www.atlassian.com/software/jira
	Confluence	https://www.atlassian.com/software/confluence
代码(Code)	GitHub	https://github.com
	GitLab	https://about.gitlab.com
构建(Build)	Gradle	https://gradle.org
	Bazel	https://bazel.build
	Webpack	https://webpack.js.org
测试(Test)	Jest	https://jestjs.io
	Playwright	https://playwright.dev
	JUnit	https://junit.org
发布(Release)	Jenkins	https://www.jenkins.io
	Buildkite	https://buildkite.com
部署(Deploy)	Docker	https://www.docker.com
	Argo	https://argoproj.github.io
	AWS Lambda	https://aws.amazon.com/lambda
操作(Operate)	Kubernetes	https://kubernetes.io
	Terraform	https://www.terraform.io
监控(Monitor)	Prometheus	https://prometheus.io
	Datadog	https://www.datadoghq.com

3. 教改项目设计与实施

本课程改革的核心目标是让学生通过实践掌握行业标准工具的应用，并在团队协作中理解企业项目的开发流程。为了实现这一目标，课程设计涵盖了以下几个方面。

首先，在工具的选择上，课程引入了 Git、Jenkins 和 Gerrit 三种企业常用工具。Git 被用于版本控制，学生可以在实际操作中体验代码冲突的处理方法，并理解分支管理的核心概念。Jenkins 作为持续集成工具，让学生感受到自动化测试和构建在项目开发中的重要性。Gerrit 则通过代码审查功能，帮助学生提高代码质量意识，并学习如何与团队成员有效沟通。

课程的实施过程中,教学团队设计了一系列逐步递进的实践任务。例如,学生首先学习 Git 的基本操作,包括初始化仓库、创建分支和合并代码。随后,他们会参与模拟的团队项目,其中故意引入代码冲突,以便学生掌握解决冲突的方法。通过这种方式,学生逐步适应团队协作中的实际场景。

在团队任务中,学生被分配到不同的角色,如项目经理、开发人员、质量保证人员等。这种分工让学生体验到真实项目开发中的角色定位和责任分担,同时培养了他们的沟通能力和领导力。任务的设计注重逐步加深,例如初期的简单任务以工具学习为主,而后期的综合项目则要求学生整合学到的技能,解决更复杂的问题。为了提升学生的参与度和学习效率,课程设计了基于 Git 提交记录的评分体系。这一体系通过量化学生的提交频率和代码质量,结合团队整体表现,对个人贡献进行评估。AI 技术在其中发挥了重要作用,通过分析学生的代码风格和工作量,提供即时反馈,并辅助教师完成评分任务。

评分体系的设计强调公平性和透明度。例如,学生的提交记录会被公开展示,并与团队成员共享。评分 excel 表格可见后文“实践效果与反馈”部分列出的腾讯在线文档。学生工作量考核以 Git 上文档和代码的行数为参考,使用 git-quick-stats 工具[11]完成。这种方式不仅促进了团队内部的交流,也增强了学生对自身任务的责任感。此外,课程中还引入了实时反馈机制,学生可以在完成任务后立即获得评分和建议,从而迅速调整自己的学习策略。

为了帮助学生在毕业后更快融入行业,课程还加入了职业发展模块,帮助学生优化个人的 GitHub 主页,将其作为展示自身能力的“代码名片”。这种策略直接提高了学生的就业竞争力,使其在求职中更具吸引力。

4. 实践效果与反馈

在 2022 年至 2024 年的课程实践中,全部学生(约 50~60 人/学期)共完成了超过 2000 次/学期代码提交,其中提交次数最多的团队达到了 790 次。通过工具的实际应用,学生深刻体会到了协作开发的复杂性和必要性。尤其是在代码冲突的处理过程中,学生逐渐掌握了利用 Git 进行分支管理和问题解决的技巧。这些实践经验显著提高了学生对版本控制和团队协作的理解。

课程还通过学生反馈收集了大量数据。许多学生表示,在解决代码冲突的过程中,他们逐步理解了团队协作的精髓,并感受到了完成复杂任务的成就感。一名学生提到:“以前我觉得版本控制工具很麻烦,但经过这门课的实践,我发现它是团队开发中不可或缺的工具。”这样的反馈反映了课程设计的成功之处。

评分体系的透明性和公平性也得到了学生的高度评价。通过清晰的评分标准,学生能够明确自己的努力方向,从而激发了他们更高的学习热情。同时,学生的 GitHub 主页被视为“代码名片”,不仅记录了他们的学习成果,还为未来的求职提供了有力的支持。多名学生表示,这一实践课程显著提升了他们在面试中的竞争力。从量化数据来看,学生的综合成绩和团队协作能力显著提高。课程的通过率达到了 100%,与《软件工程》等使用传统教学模式同类课程相比提高了数个百分点。此外,学生在课程结束后对工具的应用满意度达到了 95%,这表明课程改革取得了良好的效果。可查阅的学生评分表及共 9 个团队项目文档和代码 Git 地址参见以下网址,Git 工具记录了每位同学的每次提交,见证了整个项目的从无到有。

2022 年:【腾讯文档】软件项目综合实践打分表 2022

<https://docs.qq.com/sheet/DSHZpeVR2Ulh6YlJQ?tab=BB08J2>。

2024 年:【腾讯文档】软件项目综合实践打分表 2024

<https://docs.qq.com/sheet/DSFlrWHZkd1p6VmZw?tab=01mtly>。

5. 理论与实践的结合

本课程的实践证明, 陶哲轩提出的“通过工具和模块化任务分工提升协作效率”的观点在软件工程教育中具有重要价值。尽管人工智能技术日益发展, 大型语言模型和智能代理的能力也在不断增强, 但这并未削弱传统软件工程思想的价值。相反, 正因为各行各业的复杂性和对协作的依赖日益加深, 传统软件工程中模块化、工具支持和团队协作的核心思想显得尤为重要。例如, 学生在完成项目时, 尽管可以借助 AI 工具加速某些特定环节的效率, 但实际的工程问题仍然需要通过团队的协作与系统性思维来解决。这种基于传统软件工程理念的团队实践, 不仅适用于教育, 还能推广到其他学科和行业。通过模块化的任务分配和高效协作工具, 传统软件工程的思维方式在跨学科项目中同样能够发挥重要作用。这一模式对教育未来的发展也提出了新的启示。无论技术如何进步, 教育的核心仍在于帮助学生掌握解决复杂问题的能力。通过引入现代协作工具与团队实践方法, 教师可以培养学生的系统性思维 and 创新能力, 使其在不断变化的环境中依然能够胜任挑战。

6. 结论与展望

本研究通过对《软件项目综合实践》课程的改革, 验证了传统软件工程思想在人工智能时代的持久价值与深远意义。尽管人工智能技术为教育带来了诸多便利, 如自动评分、智能反馈等, 但这些技术并不能替代软件工程的核心原则。通过模块化设计、工具支持和团队协作的实践教学模式, 学生不仅能够掌握软件工程的基础技能, 还能学会在复杂项目中高效协作和解决问题。

未来, 研究可以进一步探讨在不同领域中推广传统软件工程理念的可能性, 例如如何结合 AI 技术优化跨学科项目中的协作效率。此外, 课程改革还需要适应不同文化背景和教育体系, 通过因地制宜的方式满足学生的实际需求。通过持续的研究和实践, 软件工程教育将不断创新, 为各行各业培养出更为全面和适应力强的优秀人才。

基金项目

北京信息科技大学校级教学改革一般项目(项目编号: 2023JGYB24, 项目名称: 构建更真实企业项目场景的团队综合项目实践课程改革)。

参考文献

- [1] 杨芙清. 软件工程技术发展思索[J]. 软件学报, 2005(1): 1-7.
- [2] 顾明. 软件工程中几种常用软件生命周期模型的简介[J]. 计算机时代, 2003(1): 20-21.
- [3] 田道云, 陈翔, 仲伟俊, 等. 浅析软件项目的规范化管理[J]. 计算机与现代化, 2002(3): 7-10+14.
- [4] 翁南钊, 蔡德钧. 基于构件的软件工程及其测试、维护和实践[J]. 计算机工程与应用, 2000(2): 33-36.
- [5] 腾讯网. 陶哲轩对话 OpenAI: 从数学出发, AI 如何改变每个人的工作方式? [DB/OL]. 2024-12-11. <https://news.qq.com/rain/a/20241211A00UZ400>, 2024-12-31.
- [6] 王晓龙. 基于敏捷开发的持续交付系统的研究[J]. 信息技术, 2023(5): 131-136.
- [7] 刘畅. 配置管理在软件能力提升中的作用[C]//天津市电子学会. 第三十六届中国(天津) 2022' IT、网络、信息技术、电子、仪器仪表创新学术会议论文集. 天津: 天津光电通信技术有限公司, 2022: 5.
- [8] 李正阳. 系统研发的持续集成与持续交付技术的研究与实现[J]. 中小企业管理与科技(中旬刊), 2021(3): 177-180.
- [9] 单美贤, 董艳, 洪荣昭. 基于项目式学习的同伴在线协作学习活动设计与应用研究[J]. 黑龙江高教研究, 2021, 39(1): 94-100.
- [10] 刘丽霞. 浅谈 PBL 在程序设计教学中的应用[J]. 教育现代化, 2016, 3(27): 165-166+171.
- [11] Github (2024) Git-Quick-Stats. <https://github.com/git-quick-stats/git-quick-stats>