基于 AI 伴学的软件测试实验教学平台构建与实践

李绘卓　唐峻　范勇　潘娅

西南科技大学计算机科学与技术学院，绵阳 621010

摘　要　针对《软件测试技术》实践课程评价不一、评阅耗时、反馈滞后的教学痛点，本文提出了“教师导学+工具裁定+AI 伴学”三元协同教学模式，研发了离线、在线双模实训教学平台。平台集成多语言工具链，覆盖单元测试、自动化测试、接口与性能测试等全栈场景，自动采集代码覆盖率、编码规范等量化数据。结合大模型搭建 AI 伴学模块，通过定制化提示词完成问题诊断与学习引导，构建“诊断—反馈—改进—验证”实训闭环。经 140 名学生教学实践验证，学生脚本正确率、测试规范达标率分别达 94%、95%，大幅减少实验迭代次数、缩短教师评阅时长。该模式有效落实教学减负增效，提升学生工程实践能力，为计算机实践课程数字化改革提供可复制的实施路径。

关键词　软件测试技术，AI 伴学，测试工具链，三元协同教学

Construction and Practice of an AI-Assisted Software Testing Experimental Teaching Platform

Hui Zhuo Li　Jun Tang　Yong Fan　Ya Pan

College of Computer Science & Technology Southwest University of Science and Technology
Mianyang 621010, China
huizhuoli@foxmail.com

Abstract—To address the long-standing pedagogical pain points in the practical course of Software Testing Technology—including inconsistent assessment standards, time-consuming grading, and delayed feedback—this paper proposes a tripartite collaborative teaching model featuring "teacher-guided instruction, tool-based assessment, and AI-assisted learning", and develops an offline-online dual-mode practical training platform. The platform integrates a multi-language toolchain supporting full-stack testing scenarios including unit testing, automated testing, API testing, and performance testing, and automatically collects quantitative metrics such as code coverage and coding compliance. An AI-assisted learning module powered by large language models (LLMs) is developed, which identifies root causes of coding errors and delivers targeted learning guidance via customized prompts, forming a closed-loop practical training workflow of "diagnosis-feedback-improvement-verification". Validated through a teaching experiment involving 140 students, the proposed model achieves a 94% correctness rate of test scripts and a 95% compliance rate with testing specifications, while significantly reducing the number of experimental iterations and shortening teachers' grading time. It effectively reduces teachers' workload while improving teaching efficacy, enhances students' engineering practice competence, and offers a replicable implementation pathway for the digital transformation of computer science practical courses.

Keywords—Software Testing Technology;AI Companionship; Testing Tool chain; ternary collaborative teaching model

1　引　言

《软件测试技术》作为软件工程的核心实践课程，具有工具链碎片化、语言生态多元及工程属性强的特点。当前教学普遍面临“三高两难”困境：即工具链复杂度高、人工评阅耗时高、认知浅层化率高、教师精细指导难、评价标准统一难。而在传统的评阅模式下，学生的工程实践能力难以得到实时校准与提升。为此，国内外引入了主流测试工具及各类教学平台，但仍存在一些显著局限，如：通用型平台（如对分易、雨课堂）缺乏针对测试流程的深度适配，而专业型平台（如慕测平台）则因架构固化，难以贴合本校差异化教学目标与日常教学场景[1]。

针对上述问题，我校于2020-2024年开展了教学信息化建设，研发了基于IntelliJ IDEA和Eclipse的本地化测试实践辅助插件，配合Web服务端构成“插件采集+服务端汇总”的评测模式。在该模式中，学生在本地IDE中执行测试用例，插件自动采集测试通过率、覆盖率等指标，并将测试脚本和结果数据上传至Web服务

*基金资助：2024-2026 年四川省高等教育人才培养质量和教学改革重点项目”AI 赋能软件开发课程群教学改革与探索”（JG2024-0719）；四川省自然科学基金重大项目（2025ZNSFSC0005）

*通讯作者：李绘卓　huizhuoli@foxmail.com

端，实现了一定程度的自动化评测。

然而，该模式仍存在三方面明显局限：一是分析粒度较粗，评价主要依赖通过率和覆盖率，缺乏对代码规范、断言质量、接口响应及性能指标的多维度裁定；二是语言与测试类型覆盖有限，高度依赖特定IDE环境，主要面向Java语言的单元测试，难以适配Python、C++等多语言生态及Web自动化、接口与性能测试等全栈场景；三是反馈机制呈单向断层，系统仅返回“对错”数据，缺少针对性的问题归因与修改建议，导致学生“知错难改”，教师仍需投入大量精力进行二次人工答疑。

鉴于此，本文提出“教师导学-工具裁定-AI伴学”三元协同教学模式，并自主研发了配套的AI伴学评价系统。这种“模式牵引、系统落地”的改革路径，有效实现了从“人工判分”到“智能伴学”的范式转型[2]，为提升学生工程实践能力提供了可复制的数字化解决方案，系统形态的演化对比如表1所示。

表 1 系统形态的演化对比

阶段	形态	核心技术	主要局限
传统人工	线下作业 + 人工批改	无	反馈慢 标准不一
插件服务端模式	IDE 插件 +Web 服务端	本地测试采集、通过率/覆盖率统计	多语言/全栈支持弱、反馈单向/兼容性差
三元协同模式	多语言 AI 伴学平台	多工具链集成、大模型诊断、全流程闭环	——

2 “三元协同”教学模式构建与评价机制

为解决传统教学中教师低效纠错、评价标准不一、学生缺乏个性化指导等问题，本文构建了“教师导学+工具裁定+AI 伴学”的三元协同教学模式。该模式明确了三方职能边界与协同逻辑，形成了“高阶引领—刚性度量—柔性辅助”的教学闭环，并将教学重心从“结果判定”转移至“过程优化”[3]，具体三元协作流程如图 1 所示。

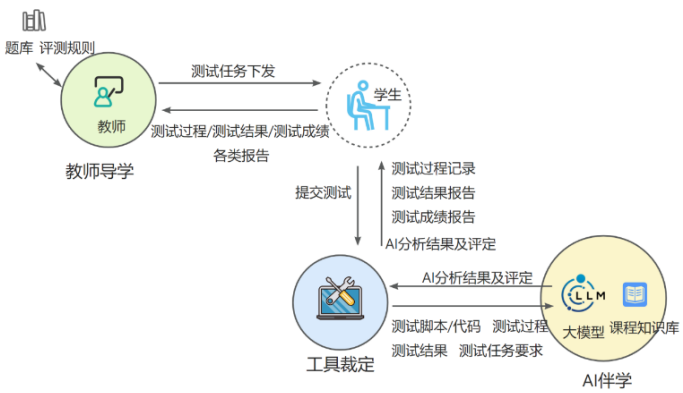


图 1 三元协同 AI 伴学模式

教学过程中，【教师导学】聚焦任务设计与高阶思维引导，【工具裁定】依托标准化测试平台实现代码质量、性能指标等客观维度的自动校验，【AI伴学】则通过多模态交互实时诊断学习障碍、动态推送适配资源。三者数据互通、反馈互促，共同支撑形成“诊断—反馈—改进—验证”的螺旋式能力成长机制：

1. 教师导学（高阶引领）与自主实践

在“三元协同”架构中，教师作为高阶引导者，通过配置评分规则与个性化发布任务来主导教学方向，促使学生开展自主探索。教师端依托可配置的多语言评分引擎与多维度题库，实现个性化实验发布与全流程管控。这种“教师导学”模式使教师能从低价值的脚本查错中解放，转而聚焦测试策略设计与学情分析，重点培养学生的工程决策能力。

2. 工具裁定（刚性执行）与客观反馈

系统依托测试框架与工具构建刚性执行层，针对不同测试类型设定差异化指标集，遵循“执行-采集-报告-评分”流程自动化裁定实验结果，消除人为评分波动；在系统工具支持下，学生可在本地开发测试环境中，采用指定测试工具链完成实验任务。系统工具自动调度多类型实验测试脚本执行，记录测试过程、收集测试结构数据，确保实验结果的可重复性与数据准确性，同时依据预设评分规则完成自动化评分，生成涵盖白盒、Web自动化、Web Api及性能测试等类型的多维度成绩分析报告，为学生提供即时、客观的反馈，全面保障评价标准的一致性与客观性。

3. AI伴学（柔性反馈）与迭代优化

基于大模型构建“AI伴学”柔性反馈机制[4]，通过定制Prompt注入测试脚本、执行过程记录、执行结果与评判规则等上下文，结合课程知识库规避模型幻觉，提供思路引导与根因分析，不直接生成代码，在保留学生独立思考空间的同时实现实时辅助[5],参考提示词框架如表2所示。AI引擎对实验过程进行多维度诊断，精准定位失败根因并给出优化策略，引导学生进入“工具裁定→AI伴学→修正迭代”的闭环，强化对测试知识的深度内化。下面为prompt范例。

【角色】：软件测试助教

【上下文】：{学生测试代码} + {执行日志} + {评分规则} + {考察知识要点}

【约束】：不直接给出代码，只给思路与根因提示

【任务】：分析失败原因，给出修改建议

总的来说，通过构建“工具裁定结果→触发 AI 诊断→生成个性建议→学生迭代→学情可视→教师调整重点”的全流程闭环管理，系统有效减少了实验迭代次数并提升了完成质量，显著增强了软件测试实践教

学的效率、客观性与教学质量，有力支撑了从“人工判分”向“智能伴学”的范式转型。

3 实验教学平台设计

3.1 多工具链集成与多模式部署架构

为支撑软件测试教学中存在的多语言生态并存、工具链碎片化及高校网络环境复杂多样的现实需求，系统在实现逻辑上划分为用户交互层、测试执行评测层、测试工具链层及题库资源层，同时支持离线与在线双模式运行（如图 2 所示），旨在实现评测流程标准化、环境适配灵活化及教学管理轻量化。

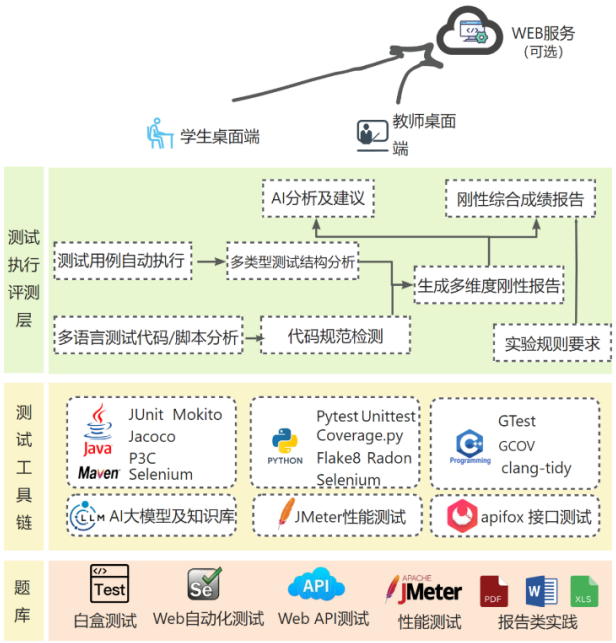


图 2 三元协同 Ai 伴学系统设计

系统架构设计如下：

1. 用户交互层：双端协同的教学触点

该层面向教师/学生提供交互入口。其中教师端侧重于实验全生命周期管控与多语言生态（Java/Python/C++）题库建设。使教师从繁琐的脚本查错中解放，聚焦高阶导学。

学生端则依托桌面端部署优势，支持零配置接收实验任务，并在本地开发环境中完成测试执行与结果查看，确保教学操作的便捷性与低门槛。

2. 测试执行评测层：多工具链的刚性裁定引擎

作为系统的核心执行中枢，该层负责调用底层测试工具链，屏蔽异构测试工具链的差异性，实现评测流程的刚性自动化。针对Java生态，系统集成JUnit、Mockito、Jacoco及P3C等工具，支持单元测试、覆盖率统计与代码规范检测；针对Python生态，集成Pytest/Unittest、Coverage.py及Flake8/Radon；针对C++生态，集成GTest、GCOV及clang-tidy。此外，系统还纳入Selenium（Web自动化测试）、JMeter（性能测试）与Apifox（Web Api测试），分别支撑对应测试场景。该层通过统一的任务调度接口与标准化结果解析器，将不同工具产生的XML、JSON或HTML输出转化为系统内部统一的数据结构，确保评价标准的客观统一[6]。

3. 题库资源层：结构化实验资产库

底层题库资源是支撑多样化实验的基石。系统构建了包含白盒测试、Web自动化测试、Web Api测试、性能测试及测试报告类实践（PDF/Word/Excel）在内的多维题库体系，以及自定义测试评判规则。这种结构化设计不仅满足了教师针对不同教学阶段（基础训练、综合实战）的组卷需求，也为学生提供了从测试设计、脚本开发到报告交付的全链路工程化训练环境。

4. 离线与在线双模式部署结构

针对高校校园网带宽波动、部分实验室无外网连接及设备异构性强等实际痛点，系统在架构层面引入“离线与在线双模式”部署结构，以保障教学活动的连续性与稳定性。离线模式下，系统以教师端与学生端桌面程序方式执行，可辅助第三方平台（如对分易、雨课堂等）分发和导入学生实践作业，支持本地任务接收、脚本执行与报告生成，结果可本地缓存，待网络恢复后同步。在线模式下，配合桌面程序实现成绩与报告数据的云端同步，支持多终端访问、学情集中管理与数据可视化分析。

两种模式通过统一的数据格式与接口规范实现无缝切换，教师端可灵活配置实验发布模式，学生端自动适配当前网络状态。系统核心评测引擎与题库资源在两种模式下保持一致，确保教学评价标准统一。

通过上述分层设计与双模式部署，系统不仅解决了多语言、多工具链集成的技术复杂性，更从工程实践角度有效适配了高校多样化的信息化教学环境，为软件测试实践教学的稳定开展提供了坚实的技术底座。

图3为教师端功能界面截图，涵盖题库管理与学生作业管理两大核心模块。学生端界面则聚焦于实验成果的反馈与迭代，图4为学生端操作界面。

3.2 标准化题库建设

在实验资源建设方面，课程围绕软件测试核心能力培养目标，系统化设计了8个层次递进的实验项目，覆盖测试管理、黑盒测试、白盒测试、单元测试、Web自动化测试、Web API接口测试、性能测试以及Web系统综合测试等关键实践环节，形成了较为完整的实验体系。

配套建设了50余个标准化实验题库及对应的评阅规则，为自动化判定与过程性评价提供统一标准。同

时，针对每个实验题目构建了结构化的提示词体系，将实验任务、预期结果及常见错误类型等信息进行规范化封装，为AI伴学模块提供稳定的上下文输入，使其能够给出更具针对性的分析与指导，从而实现“工具裁定+AI引导”的有机融合[7]。

在此基础上，课程搭建了被测Web平台，内部集成4个典型Web服务系统，支持本地启动。该设计既便于学生在可控环境中反复开展测试实践，又有效避免了公网依赖和性能测试过程中对公共服务器造成的压力，显著提升了实验的可控性与教学效率。



图3(a) 教师端题库界面

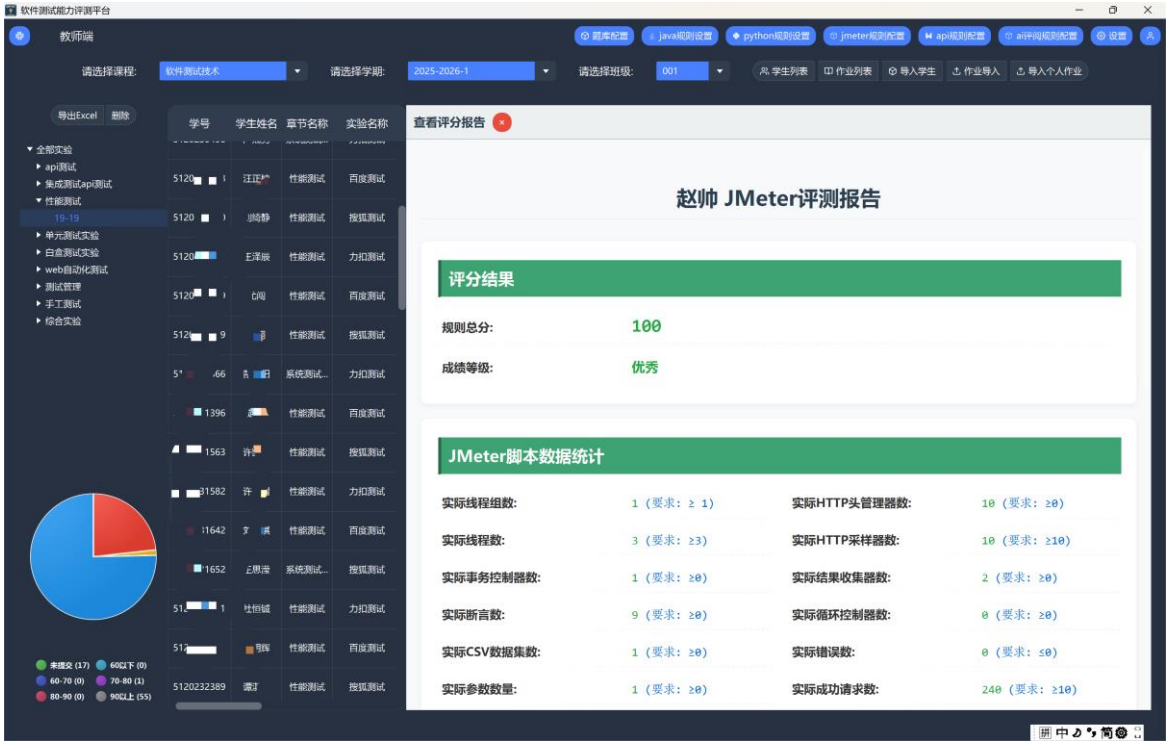


图3(b) 教师端学生作业界面

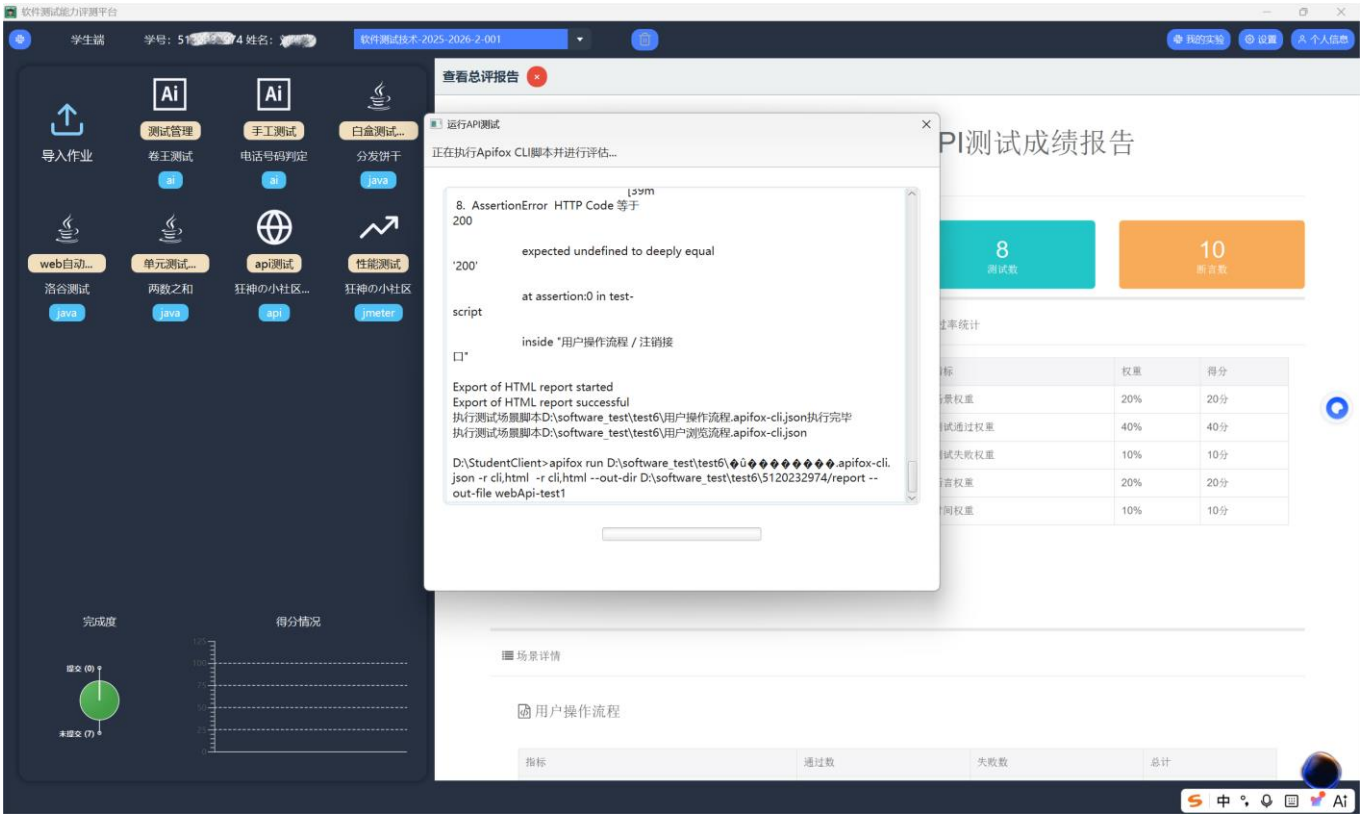


图4 学生端脚本执行界面

4 教学实践与效果分析

该模式于2024-2025学年秋季学期在软件工程专业2个教学班（共140名学生）中实施应用，目前采集了涵盖测试过程记录、代码/脚本质量分析、测试执行结果及综合评价的等多维核心数据，并与传统人工模式及插件服务端模式（我校既往模式）进行对比（如表2所示）。实践表明，依托T+0分钟实时反馈周期及智能伴学机制，学生不仅显著提升了测试工具应用能力和软件开发全周期测试能力，还在AI精准指引下有效降低了单次实验的平均迭代次数，多项测评指标显著优化。与此同时，教师教学效能显著提升。

表 2 不同教学模式下能力培养成效对比

评价维度	核心指标	传统人工模式	插件服务端模式	三元协同模式	能力提升体现
使用现代工具能力	自动化脚本编写正确率	75%	82%	94%	学生能更熟练地运用测试工具链解决复杂工程问题，工具应用能力显著增强。
软件开发全过程测试能力	测试覆盖率、代码规范性、接口测试指标、性能测试指标等	72%	80%	95%	学生不再满足于“跑通”，而是追求测试的充分性与规范性，工程素养提升。
实证研究与分析能力	实验平均迭代次数	–	5	3	借助AI伴学即时反馈，学生能更快定位根因，实证分析与问题解决效率大幅提升。
教师效能	人均评阅耗时（h/百人）	10	5	2	教师从低阶查错中解放，有更多精力投入到高阶教学设计与指导中。

5 结束语

本文针对软件测试实践教学“三高两难”痛点，构建“教师导学+工具裁定+AI伴学”三元协同教学模式，研发多语言、多工具链兼容的AI伴学评价系统，形成刚柔并济的实时评价与迭代学习闭环。经教学实践验证，该模式显著缩短评测反馈周期，提升学生实验完成质量与代码规范水平，有效降低教师机械评阅负担，推动实践教学从人工判分向智能伴学转型。不足之处在于AI伴学中存在个性化程度不足与课程AI知识库建设仍需加强。未来将扩展测试场景与工具生态，完善课程AI知识库建设，并面向更多工科实践课程推广验证，持续完善数据驱动的智能教学范式。

参 考 文 献

[1] 边奕心,赵松,朱晓.基于慕测平台的软件测试混合式实验教学模式初探[J].计算机教育,2020(11):170-173,182.

[2] 易凯谕,韩锡斌.从混合教学到人智协同教学:生成式人工智能技术变革下的教学新形态[J].中国远程教育,2025,45(4):85-98.

[3] 刘妍,李琳,陈文俊.AI 伴学准备好了吗?——GAI 赋能高校课堂人机协作教学设计与实践研究[J].电化教育研究,2025,46(12):113-121.

[4] 高利鹏,蔡文静,郑炜.大模型与知识图谱驱动的软件测试智能辅导系统[J].软件导刊,2025,24(11):49-53.

[5] 王华,吴敏敏.基于生成式 AI 的软件测试教学模式研究[J].电脑与电信,2025(3):67-71.

[6] 刘小飞,崔海燕,兰丽.AIGC 赋能软件测试技术课程教学探索与实践[J].西部素质教育,2025,11(12):10-13,61.

[7] 吴伶琳.AIGC 驱动的高职软件测试教学资源人机协同开发研究[J].办公自动化,2025,30(21):50-52,98.