 DOI: https://doi.org/10.12349/iser.v7i6.10790

Research on the Practice and Application of AI-Assisted Technology in Software Functional Automated Testing

Fan Si

Tonghao Information Industry Co., Ltd., Beijing, 100071, China

Abstract

Addressing the challenges of frequent functional changes and complex interface coupling in continuous delivery, traditional manual testing struggles to meet high-frequency iteration requirements. This paper proposes an AI-assisted software functional automation testing solution that integrates learning paradigms with engineering architecture design, establishing a three-tier architecture comprising the data layer, model layer, and application layer. Methodologically, supervised and reinforcement learning are employed for intelligent test case generation, with uncertainty gating implemented to ensure release quality; cross-modal defect detection combining computer vision and natural language processing enhances observability; and cluster-driven result analysis generates governance checklists. Results demonstrate that this solution effectively addresses the limitations of traditional testing without altering the release schedule, achieving simultaneous improvements in testing efficiency and software delivery quality.

Keywords

AI-assisted testing; reinforcement learning; intelligent use case generation; cross-modal defect detection; continuous delivery closed-loop

AI 辅助技术在软件功能自动化测试中的实践与应用研究

思璠

通号信息产业有限公司，中国 · 北京 100071

摘　要

面向持续交付中功能频繁变更与接口耦合复杂的问题，传统人工测试难以适配高频迭代需求，本文提出将学习范式与工程架构协同设计的AI辅助软件功能自动化测试方案，构建数据层、模型层、应用层三层架构。方法上，利用监督与强化学习进行用例智能生成，设置不确定性门控保障上线质量；以计算机视觉与自然语言处理联合的跨模态缺陷检测提升可观测性；通过聚类驱动的结果分析形成治理清单。结果表明，该方案能在不改变上线节奏的前提下，有效弥补传统测试短板，实现测试效率与软件交付质量的同步提升。

关键词

AI辅助测试；强化学习；用例智能生成；跨模态缺陷检测；持续交付闭环

1 引言

软件系统在云原生与业务互联网化推动下进入高频发布常态，需求快速演进导致接口契约、界面结构与依赖服务持续漂移，给传统脚本化自动化带来设计滞后、维护高开销与弱异常漏检等挑战。与此同时，企业在持续交付流水线中需要将测试活动前置并闭环化，以在稳定节奏下获得更高缺陷可观测性与资源利用效率。为此，本文面向企业级应用提出 AI 辅助的软件功能自动化测试思路，将监督学习与强化学习与工程架构协同，形成自下而上的数据采集与特征库、面向用例生成与缺陷检测的模型层、服务于调度与回归的应用层，并通过统一特征接口与在线训练打通数据到执行的闭环。研究旨在探索可迁移、可解释且低维护成本的测试能力，支撑多模块多版本的持续演化。

2 AI 辅助软件功能自动化测试的理论基础与技术架构

鉴于企业级应用持续交付场景中功能变更频繁且接口耦合复杂，AI 辅助软件功能自动化测试需把学习范式与工程架构进行协同设计。从测试知识获取机制来看，监督学习把历史缺陷标注、用例执行轨迹与界面结构特征进行映射学习，用于生成高价值用例与异常分类；强化学习把测试环境抽象为状态与动作序列，以覆盖增益与缺陷触发作为回报信号，驱动策略在多轮执行中不断优化。由此推导出三层技术架构 [1]。数据层负责采集代码差异、接口契约、运行日志与界面树，并进行清洗、匿名化与特征标准化，构建可复用的特征库。模型层承载用例生成模型与缺陷检测模型，

【作者简介】思璠（1990–），女，中国北京人，本科，工程师，从事通信信息研究。

提供统一特征接口与在线训练服务，并把不确定性评估输出给上层。应用层面向流水线与回归场景进行调度与执行，结合结果分析模块对失败归因、稳定性画像与覆盖缺口进行判定，再把执行反馈回传至数据层与模型层，形成持续演化的闭环。

3 AI 辅助软件功能自动化测试的关键技术实践

3.1 AI 驱动的测试用例智能生成

在电商平台交易链路的高频变更情境下，本文把测试用例生成问题建模为序列决策任务，并以支付模块为具象场景来组织算法与工程流程。围绕订单创建至支付完成的端到端路径，命名实体识别被用来把需求文档中的关键要素抽取为结构化标签，该技术指把金额区间、币种、支付渠道、分期参数、优惠策略、用户分层、设备形态以及网络状态等文本要素转化为可计算特征，从而把状态空间构造成界面位置、上下文会话、依赖服务可用性以及风控信号的组合，把动作空间构造成选择渠道、填写边界值、注入异常码、切换网络条件以及触发重试的原子操作序列。

从序列优化角度来看，策略需要在覆盖收益、缺陷触发概率以及脚本稳定性之间进行权衡，因此本文设定一个面向覆盖增益与错误可观测性的奖励函数，并把指示类约束直接嵌入惩罚项以抑制无效探索。模型训练阶段先借助历史执行轨迹与接口契约进行冷启动，再借助半仿真支付网关以及可控的风控与网络扰动进行在线自适应更新，同时把不确定性阈值门控作为候选用例上线的约束。与手工方式相比，手工依赖经验驱动的场景枚举以及脚本重复维护，往往在规格频繁演进时出现滞后，而策略学习能够把实体约束转化为动作优先级并在状态转移中持续校正，因而在规模化生成、去重与参数化覆盖方面表现出更强的工程可移植性。支付模块语境下，人工与强化学习方法在产出速率、变更适应以及维护开销等维度的结构化对比。

3.2 AI 辅助的测试执行与缺陷检测

鉴于持续交付背景下接口与界面持续演进，测试执行被组织为调度、环境准备、数据注入、代理执行与监控收集的闭环，并把缺陷可观测性前置到调度策略之中。本文把计算机视觉与 NLP 嵌入执行路径：视觉侧在 UI 层抓取渲染帧与界面树，借助关键点匹配、组件分割与 OCR 校验，把错位、遮挡与文案漂移量化为异常分数；日志侧在容器与网关采集多源日志，依靠模板归一与语义编码，把错误模式与跨调用因果线索转化为低维表示，并在时序上与用例步骤对齐后送入检测模型，以便把界面与日志的跨模态证据进行联合判定[2]。

立足某电商平台的订单模块，训练数据来自订单创建到支付指引的端到端轨迹，工程上把截图对与 DOM 序列对齐到可观察日志，构建视觉与文本双塔编码器，借助对比学习区分正常与弱异常，再以自训练扩展样本，并把规则引擎产生的违约信号作为辅助监督。部署阶段由执行代理在灰度环境注入边界数据与网络扰动，实时进行推理与告警，并把异常上下文封装进回放通道以支撑复现与归因。

3.3 AI 驱动的测试结果分析与优化

面向回归周期内海量执行产物研判，本研究把测试结果分析组织为以 K-means 聚类为核心的缺陷画像流程。K-means 聚类被定义为把样本划分为 K 个簇以最小化簇内平方误差的划分方法，向量化特征由缺陷置信度、视觉异常分数、日志语义偏离、调用链一致性反向量化、用例类型、环境指纹与时间窗口构成，并以轮廓系数这一聚类效果评估指标和簇内离散度来选取 K。聚类完成后由知识库对簇心进行命名，再把簇标签映射到覆盖缺口与修复优先级，形成可执行的治理清单。结合某厂商的用户管理模块，聚类结果呈现五类高频缺陷类型，分别是令牌过期处理不一致引发强制登出，角色权限矩阵配置错误导致细粒度授权缺失，表单边界校验不足造成非法输入上穿，并发下重复创建用户导致数据重复，国际化资源加载漂移触发界面语义错配。针对上述簇标签，本研究把优化建议拆解为两类动作，一类面向测试体系，包括把表单校验策略与后端契约对齐并把边界分布注入用例，构建基于权限字典的矩阵快照比对，建立认证中间件模拟器注入过期与异常分支，并对易抖动路径设置稳定性评分阈值门控；另一类面向研发流程，包括把用户创建接口加入幂等键与去重校验，把权限变更事件纳入一致性校验与异步补偿，把国际化资源包纳入规则校验与灰度发布。最终把簇标签与用例优先级、修复里程碑和训练样本回流形成闭环，在不改变交付节奏约束下提升治理聚焦度与资源分配有效性[3]。

4 AI 辅助软件功能自动化测试的应用效果与案例验证

4.1 案例场景与数据准备

鉴于位于某厂商在 CRM 客户管理模块中持续引入线索转化与审批链重构，本研究把数据准备工作限定在历史用例与缺陷记录两个来源来开展。数据采集环节依靠企业内部的 TestRail 与 Jira 实例，拉取 2024-01 至 2025-12 期间与客户信息、线索转化、角色权限、审批配置相关的历史用例 3,286 条与执行轨迹 19,472 次，同时从缺陷库抽取 2,318 条经产品确认归档的记录，补充包含模块标识、环境指纹、步骤序列、参数边界与变更版本信息，并把需求基线与接口契约版本映射到统一的变更时间轴以便后续对齐处理。

围绕数据质量与隐私合规，预处理过程被组织为清洗、脱敏与特征抽取三个子流程。清洗阶段把版本作废的用例按变更时间轴剔除，把步骤文本进行分词与词形还原后计算序列相似度，相似度达到 0.85 及以上的样本合并归档，并把跨环境字段统一到可比口径。脱敏阶段对姓名、手机号与邮箱等直接标识字段施加带盐的 SHA-256 散列，对地址经由高斯噪声扰动在 100 m 尺度内替换，同时保留分布结构以维

持统计有效性。

4.2 效果指标与对比分析

鉴于该厂商在 CRM 客户管理模块持续引入线索转化与审批链重构，本研究把评估口径固定为同一双周回归窗口，以效率、检出率与成本三类指标进行对比分析。效率维度以流水线端到端耗时为主体，覆盖调度排队、环境拉起、数据注入以及执行回收等环节；检出率维度以经产品确认归档的有效缺陷占比为度量，并把误报占比纳入稳定性侧证；成本维度以人力投入的人天数与脚本维护工作量为核算口径。依托统一特征接口与在线训练服务，把用例生成与执行代理集群整合到一体化流水线后，效率侧出现显著压缩，检出率在保留可复现性的前提下稳定抬升，成本侧的人力与返工量同步下降，具体数据见表 1。

从机制层面展开解释，效率改善主要来源于用例智能生成把准备时段压缩，并且调度编排器在资源侧进行优先级重排，促使关键路径得到提前执行；检出率的提升与跨模态缺陷判定有关，视觉日志联合编码降低弱异常漏检，同时聚类驱动的归因把重复告警合并回收，进一步净化有效缺陷比例；成本的下降与脚本迁移开销减少密切相关，统一特征接口把需求实体的变更映射为参数级调整，使得脚本层的改动集中在轻量化封装，另外反馈管理器把无效路径及时下线，降低回归周期内的冗余执行次数 [4]。

表 1 某 CRM 客户管理模块测试效果对比表

指标	口径说明	传统基线	AI 辅助方案	相对变化
单次回归总耗时	端到端壁钟时间 h	22.0	12.1	-45.0%
用例准备时长	用例设计与参数化时长 h	8.0	3.4	-57.5%
缺陷检出率	确认缺陷占执行条目比例	50.0%	64.0%	+28.0%
误报率	告警中经核验的误报占比	6.5%	4.9%	-24.6%
人力投入	双周回归周期人天	30	21	-30.0%
关键路径覆盖率	客户信息与转化链路关键用例覆盖占比	72.0%	84.0%	+16.7%

4.3 实践经验与优化方向

鉴于该厂商在 CRM 客户管理模块的多版本并行演进，本研究把数据质量当作驱动策略稳定性的核心约束来使用。进一步观察发现，覆盖率达到 85% 以上时，用例生成与缺陷检测在迁移到相邻版本时的波动明显收敛，缘由在于实体参数空间、状态行为组合以及环境指纹三类分布得到充分表征。需重点关注的是，覆盖不是样本数量的简单累加，而是把边界条件、弱异常以及跨服务耦合纳入可观测轨迹的结构化刻画，因此需要依靠分层抽样、稀有场景增强与双人一致性标注来压降噪声，把需求基线与接口契约对齐到统一时间轴，并把不确定性门控前置以拦截低置信候选 [5]。

由此推导出两类优化方向。本研究立足融合视角，把检索增强的用例生成模型、强化学习策略、规则约束解算器以及视觉日志双塔进行融合，选用门控加权的元学习器在回归窗口内自适应分配权重，校准信号来自离线回溯与灰度反馈。面向跨平台适配，构建统一交互语义与特征接口，把点击、输入、滑动以及通道切换等原子动作映射到 Android、iOS、Web 与小程序的控件语义，依靠适配器层完成定位与容错，配合容器化执行与能力探测来维持设备矩阵的一致性，并把数据契约、覆盖阈值与回滚策略固化为治理规则以支撑持续演化。

5 结语

本文验证了三层架构与策略学习在企业级场景的有效性，用例智能生成、跨模态缺陷检测与聚类归因共同构成闭环提升路径。实践表明，数据质量与覆盖结构化刻画是稳定性的关键，应将不确定性门控与反馈管理前置。未来将沿多模型融合与元学习权重自适应、跨平台统一交互语义与容错适配、隐私保护与可解释评估等方向深化，进一步提升通用性与工程可用性。

参考文献

[1] 杨浠,熊盼,郑旭飞,吴海林,郭伟.AIGC辅助软件单元测试的研究[J].人工智能科学与工程,2024,(04):31-41.

[2] 雷雪.人工智能在计算机应用软件开发中的应用探索[J].中国信息界,2025,(06):129-131.

[3] 李晖,辛华.基于RPA+AI的多目标自动化软件测试系统设计[J].粘接,2025,52(07):144-146.

[4] 黎晖,于宏宇,张绍平,林柯军.基于框架的Web服务软件自动化测试技术[J].兵工自动化,2024,43(08):43-46+79.

[5] 马静芳,郑宇.浅谈AI技术在软件测试中的应用[A].天津市电子工业协会,天津市电子工业协会2025年年会论文集[C].逸兴泰辰技术有限公司;星际空间(天津)科技发展有限公司;:天津市电子工业协会,2025:55-57