Service Science and Management 服务科学和管理, 2026, 15(6), 1315-1319
Published Online November 2026 in Hans. https://www.hanspub.org/journal/ssem
https://doi.org/10.12677/ssem.2026.156153

Hans汉斯

浅谈人工智能驱动的软件测试新范式

杨　怡*，党鹏飞，雷虚竹，张笑妍，李春晓

32021部队，北京

收稿日期：2026年10月19日；录用日期：2026年11月12日；发布日期：2026年11月20日

摘　要

随着人工智能与大模型技术的快速发展，软件测试领域正经历深刻变革，逐步从传统的手工与自动化测试向智能化、自适应方向演进。AI技术凭借其在自然语言理解、逻辑推理与内容生成等方面的显著优势，为测试各环节提供了新的动力和创新解决方案。多智能体协同、自适应测试、无监督学习等创新方法的应用，大幅提升了测试效率、覆盖率和准确性，推动软件质量保障体系向更智能、更高效的方向发展。本文系统探讨了AI驱动下软件测试的新范式、实践路径与面临挑战，并对未来技术趋势进行展望，旨在为行业的智能化转型提供理论指导和实践参考。

关键词

人工智能，大模型，软件测试，智能化，新范式

A Brief Discussion on the New Paradigm of AI-Driven Software Testing

Yi Yang*, Pengfei Dang, Xuzhu Lei, Xiaoyan Zhang, Chunxiao Li

Unit 32021, Beijing

Received: October 19, 2026; accepted: November 12, 2026; published: November 20, 2026

Abstract

With the rapid advancement of artificial intelligence and large model technologies, the field of software testing is undergoing a profound transformation, progressively evolving from traditional manual and automated testing towards intelligent and adaptive directions. Leveraging its significant advantages in natural language understanding, logical reasoning, and content generation, AI technology injects new momentum and innovative solutions into various testing phases. The application of innovative methods such as multi-agent collaboration, adaptive testing, and unsupervised learning has

*通讯作者。

文章引用: 杨怡, 党鹏飞, 雷虚竹, 张笑妍, 李春晓. 浅谈人工智能驱动的软件测试新范式[J]. 服务科学和管理, 2026, 15(6): 1315-1319. DOI: 10.12677/ssem.2026.156153

substantially enhanced testing efficiency, coverage, and accuracy, thereby driving the software quality assurance system towards a smarter and more efficient future. This paper systematically explores the new paradigms, practical pathways, and challenges of AI-driven software testing, while also providing an outlook on future technological trends. It aims to offer theoretical guidance and practical references for the intelligent transformation of the industry.

Keywords

Artificial Intelligence, Large Models, Software Testing, Intelligentization, New Paradigm

Copyright © 2026 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

在数字经济加速渗透的今天，作为各行各业核心竞争力关键载体的软件，其质量直接关系到企业运营效率、用户体验乃至行业安全，而传统软件测试长期面临手工测试占比高导致的效率低下、复杂系统下测试覆盖不完整(行业统计显示传统测试方法平均仅能覆盖 60%~70%的系统逻辑路径)、测试执行周期冗长难以匹配现代软件敏捷开发与快速迭代节奏、资深测试人员经验知识难以系统化传承导致团队能力参差不齐以及频繁需求变更使测试用例维护成本居高不下等诸多瓶颈，这些问题严重制约了软件质量保障体系的升级。

随着 DeepSeek、千问、GPT 系列等开源与商用大模型的相继发布而使软件工程领域经历前所未有的深刻变革并正式迈入以智能化为核心特征的软件工程 3.0 时代，这一技术浪潮对传统软件工程流程与开发模式的重新定义及让软件测试领域突破性重构成为可能，结合 BIS 年度经济报告显示的 AI 编程工具使程序员生产效率提升超 50%，大模型凭借卓越的自然语言理解能力、逻辑推理与生成能力及高质量内容创作能力为解决行业痛点提供强有力的智能化辅助与增量价值，推动软件测试从被动响应式向主动预判式、从单点自动化向全流程智能化转型[1]。

2. AI 大模型助力构建软件测试新形态

2025 年以前大模型技术在软件测试领域实践主要集中于测试用例生成、测试执行、测试报告及缺陷修复等环节并通过自动化和智能化手段提升测试效率与准确性，今年以来其技术增长点正逐步向测试需求分析这一上游环节延伸[2]，在需求洞察与拆解方面基于历史项目数据和行业经验精准识别用户需求潜在意图、将复杂需求拆解为可执行测试子项并补充行业内隐性需求及标准规范以确保测试覆盖全面性，通过构建关联图谱直观展示需求间依赖关系并厘清系统整体逻辑，在应对测试需求变更方面通过智能识别变更范围、分析评估影响程度、量化变更后工作量、生成风险报告以辅助支撑决策并同步调整测试用例与数据确保测试一致性，在软件质量保证方面通过对需求完整性勘察、对模糊表述量化识别潜在逻辑冲突并提前预警高风险点，在测试双向映射方面通过自动关联需求与对应测试用例、测试脚本生成覆盖矩阵确保测试充分性并将缺陷与原始需求关联精准定位问题根源以提升缺陷修复效率，这些能力的结合使大模型在测试全生命周期中发挥更关键的作用。

3. 多智能体协同加速软件测试智能化进程

在大模型时代，软件测试智能化水平正获得显著提升，其发展走向呈现出由基于大模型的单点能力

强化逐渐朝着基于智能体的全流程深度赋能演进的态势。这一演进不只体现在测试效率的提高方面，更在于测试智能化程度的加深，让测试过程变得更为精准、高效且自动化。目前，借助多智能体协同，在常规功能验证、单元测试、API 测试、GUI 测试、回归测试等简单测试场景当中，已能展现出优异表现[3]。单元测试智能体运用多维分析来提升代码质量，通过逻辑检查，验证代码在预期输入情况下执行是否正确，同时保证所有分支路径都能被覆盖；通过边界检查，评估系统对典型输入、边缘输入以及无效输入的响应状态，确保系统具备鲁棒性；通过错误处理检查，查看输入错误时系统是否能提供清晰的用户提示或者避免崩溃，以此提升用户体验；另外，通过对象检查，监控代码运行之后持久对象状态的更新状况，确保数据保持一致性。代码评审智能体依托大模型超强的上下文理解能力、自主规划能力以及自主审查能力，通过“智能体 + 传统软件 + 测评人员”的三方协同机制来提升评审的质量和效率，能够智能解析代码仓库并生成详细的代码说明文档，助力快速理解代码逻辑；可以自动识别潜在问题、安全风险以及性能隐患，提前规避缺陷；还能针对问题缺陷给出优化建议，并完成代码自动修正，大幅减少人工干预[3]。未来，随着多智能体技术的进一步发展，软件测试将实现从单一环节优化到全流程智能化的跨越，推动测试效率与质量实现双重提升。

4. 智能化软件测试应用实践路径

智能化、大模型技术在软件测试行业的潜在价值已得到充分验证，实践落地也为企业带来了显著的实质效益。据中国信息通信研究院专题报告中指出，受访企业落地智能测试工具后，助力测试效率提升的程度呈现向更高水平迁移的趋势，2025 年测试效率提升 30%以上的企业占比达到 66.23%，较 2024 年提升 17.97% [4]。在实践探索中，行业逐步积累了丰富经验，华为经历敏捷化、云化、平台化、服务化等多阶段演进，最终在 DevOps 体系下构建了覆盖全生命周期的智能测试能力，通过分析其向敏捷测试转型的实践之路[5]，可大致分为四阶段。

第一阶段为整体性评估阶段，企业需系统梳理内部测试流程，拆解从需求分析、用例设计、测试执行、缺陷管理到报告生成的全流程，评估各环节的标准化程度、自动化覆盖率及流水线化水平，如通过统计手工测试占比、自动化脚本复用率等指标判断流程成熟度，同时全面盘点包括服务器配置、GPU 数量、存储容量等在内的现有算力资源，评估其对大模型部署与运行的支撑能力，此外分析企业内部大模型技术的应用现状，明确已具备的技术基础与缺失的核心能力，综合判断是否具备智能化落地的必要条件，对暂不满足条件的企业，可优先通过云服务租用算力资源、引入轻量化智能测试工具等方式逐步推进。

第二阶段为基于评估结果深入分析以往测试工作中如需求理解偏差导致的测试遗漏、复杂场景测试用例设计困难、回归测试效率低下等重难点问题、流程瓶颈及核心痛点，结合企业业务特点与测试需求准确定位如需求分析、用例生成、缺陷定位等智能化技术核心应用场景，明确选择开源大模型二次开发、直接引入商用智能测试平台或采用混合模式的建设模式，制定明确各阶段目标、任务、时间节点及责任部门的详细实施步骤，并重点研究包括 API 接口设计、模型参数调优、调用效率优化等确保有效支撑工程化场景实际落地应用的大模型调用策略的方案设计阶段。

第三阶段为部署实施阶段，核心是构建适配测试场景的专属大模型能力。首先通过系统化收集企业内部测试用例库、代码仓库、缺陷报告、需求文档、行业标准规范等数据，经数据清洗去除冗余及错误信息、数据筛选选取高价值数据、数据标注提供监督信号，构建高质量领域数据集[6]；其次基于选定基础大模型，采用微调、提示工程、RLHF (基于人类反馈的强化学习)等技术进行调优训练，使模型学习测试领域专业知识与业务逻辑，提升特定测试场景性能；最后搭建测试环境，完成智能测试工具与现有测试流程及开发工具的集成，进行小范围试点测试以验证模型有效性与稳定性，并根据试点结果调整优化。

第四阶段为持续优化阶段，智能化测试需建立有别于传统软件测试一劳永逸模式的动态调优机制以维持高水平数据质量、保障输出结果稳定性，包括建立数据更新机制定期收集新的测试数据、缺陷案例、需求变更信息等以持续扩充与优化训练数据集，建立模型性能监控体系实时监测模型在测试准确率、响应速度、用例覆盖率等关键指标上的表现并在指标下滑时及时触发调优流程，跟踪行业技术发展动态引入最新算法模型与技术方法对现有智能测试系统进行升级迭代，以及建立反馈机制收集测试人员的使用意见与建议以持续优化系统的易用性与实用性。

5. 智能化软件测试面临的挑战与应对策略

尽管大模型在初期实践、单一场景测试领域的提质增效效果显著，但在覆盖多个关联测试场景的综合性任务中，其有效赋能仍面临协同机制、模型可信、数据安全与工程适配四个方面的挑战。

多智能体协同机制不完善导致其在应对复杂场景时缺乏成熟的协作方案。究其根源：一是不同智能体在功能定位、数据格式与通信方式上存在天然差异，缺乏统一的通信协议与数据交互标准，导致信息传递效率低、上下文数据不一致；二是智能体间的任务分配与协作逻辑灵活性不足，依靠预先编排的静态流程执行顺畅，但依据测试场景进行自适应调整的能力较弱，面对需求变更或缺陷分布超出预设路径时，成功率将大幅降低。因此，一是推动智能体间统一通信协议与数据交互标准的制定与落地，通过标准化的消息语义、任务描述与结果回传格式，消除异构智能体之间的协作壁垒；二是引入具备动态规划能力的编排框架，由协调智能体依据测试进展、缺陷分布与资源占用情况实时重新分配任务，实现由静态流水线式协作向自适应式协作的演进，进而支撑软件测试从单一环节优化迈向全流程智能化的跨越。

大模型决策过程缺乏可解释性导致对其作出的测试结果存在可靠性质疑。究其根源：大模型的表征空间与人类可理解的逻辑体系之间存在语义鸿沟，模型输出缺乏与测试依据、需求条目之间的显式追溯链条，使结果既难以验证、也难以审计。在金融、医疗、航空航天等对软件质量要求极高的关键领域，这种不可解释性直接构成了智能化测试应用推广的信任障碍。因此，应将可解释性作为智能化测试系统的工程化要求而非附加功能：一方面引入可解释 AI (XAI)技术，对模型的推理链路进行显式呈现，使每一条生成用例、每一个缺陷定位结论均可回溯至对应的需求条目或代码位置；另一方面建立“模型输出–人工确认–结果反馈”的可信使用机制，在关键质量决策环节保留人工复核节点，并利用复核数据持续校准模型行为。随着可解释测试体系的成熟，模型的透明度与可追溯性将成为其进入高可信度领域的准入条件，也将反过来促进模型自身的优化。

企业在构建高质量领域数据涉及敏感事项与模型训练要求矛盾导致存在泄露风险。究其根源：大模型的训练与推理过程本质上要求数据的集中化处理与流动，而真实的敏感数据一旦进入模型，便存在泄露与被滥用的可能；同时，跨企业、跨机构的数据共享与协同训练面临数据隔离难题，“数据不出域、价值可共享”的矛盾症结尚未得到系统性解决。因此，一是推行模型私有化部署与数据脱敏处理，确保敏感数据不出企业边界；二是引入联邦学习等技术，实现“数据不动、模型动”的协同训练模式，在保障数据隔离的前提下聚合多方数据价值；三是建立覆盖数据采集、标注、训练、推理全链路的安全审计机制，对数据的来源、用途与流向实施可追溯管理。打通数据安全与价值实现双链路是智能化测试在关键行业规模化推广的前提条件。

大模型的算力成本与工具链兼容性导致推广受阻。究其根源：智能化测试尚处发展初期，工具生态呈现碎片化格局，缺乏统一的接口规范，而通用大模型的资源开销又显著超出中小企业的承受能力。高昂的算力成本是企业应用的突出障碍。同时，新引入的大模型工具与现有测试工具链之间存在兼容性问题，不同工具在数据格式与接口标准上的不统一，导致测试流程衔接不畅、数据无法有效流转。因此，应做好“降本”与“融合”的统筹协调，在算力侧，借助模型蒸馏、量化压缩与推理优化等技术降低算力

消耗，可采用云服务租用、轻量化模型部署等方式降低企业的初期投入门槛；在工具链侧，推动测试工具 API 标准化，利用中间件屏蔽底层数据格式与接口差异，并构建“传统自动化工具 + 大模型能力”的混合架构，使新旧工具实现无缝协作，避免推倒重建带来的成本浪费。由此形成的标准化、绿色集约的技术架构，将成为智能化测试普惠化落地的关键支撑。

在应对上述挑战的过程中，还要重点推动“无监督学习驱动的未知缺陷检测”与“面向复杂应用场景的多模态测试”两项技术的发展，现有测试大多依赖有监督学习，需要大量标注数据且难以识别未见过的缺陷类型；未来，无监督学习将通过分析代码变更轨迹与系统执行行为构建正常运行模式，进而精准识别偏离正常模式的异常行为，配合异常检测模型与聚类分析算法，实现对潜在逻辑漏洞、性能瓶颈与安全风险的前瞻性发现，在复杂系统的隐性缺陷检测中尤为重要。在智能硬件、元宇宙、自动驾驶等新兴领域快速发展的背景下，软件需要处理文本、图像、语音、视频、传感器数据等多模态信息；多模态测试通过融合多源数据真实模拟用户交互与实际应用场景，可将自然语言描述的测试需求精确转化为 GUI 操作指令、API 调用序列、传感器数据输入等测试行为，完成端到端 GUI 功能、系统集成能力与用户体验的全面验证；其与虚拟现实(VR)、增强现实(AR)技术的结合，还将构建起沉浸式测试环境，进一步提升测试的真实性与全面性。

6. 结束语

本文系统探讨了 AI 驱动下软件测试的新范式、实践路径与挑战对策。研究表明，AI 大模型正推动软件测试从被动响应式向主动预判式、从单点自动化向全流程智能化转型；四阶段实践路径为企业落地提供了可操作的指引；而多智能体协同、模型可解释、数据安全与算力适配四大挑战的破解，则依赖于标准化协作框架、可解释 AI、联邦学习与绿色集约架构等技术的协同演进。展望未来，随着自适应测试、无监督学习、多模态验证等技术的持续突破，软件测试将演进为贯穿全生命周期的智能内核。这一转型不仅改变测试的工作模式，更将推动测试人员从传统手工执行者向测试策略制定者、智能模型优化者与质量风险管理者转变。为促进智能化测试的健康发展，行业需加强理论与工程实践结合、建立统一标准、加大关键技术研发投入并培养复合型人才，从而支撑复杂软件的高质量交付，为数字经济发展筑牢质量保障基石。

参考文献

[1] 景奕昕. 基于 AI 技术的软件测试系统研究[J]. 高科技与产业化, 2024(11): 39-40.

[2] 姜红德. 一站式 AI 测试平台提升软件质量[J]. 中国信息化, 2022(8): 33.

[3] 颜媚, 段虎才, 郭琛. AI 在软件测试领域中的应用与发展[J]. 信息通信技术与政策, 2018(10): 65-70.

[4] 中国信息通信研究院人工智能研究所. AI4SE 行业现状调查报告(2026 年) [EB/OL]. https://www.caict.ac.cn/kxyj/qwfb/ztbg/202604/P020260402599647343221.pdf, 2026-03-11.

[5] 华为云. DevOps 敏捷测试之道[EB/OL]. https://support.huaweicloud.com/bestpractice-testman/cloudtest_14_0001.html, 2026-05-11.

[6] 马静芳, 郑宇. 浅谈 AI 技术在软件测试中的应用[C]//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. 天津: 天津市电子工业协会, 2025: 55-57.