

人工智能技术在软件开发中的应用与实践探索

黄明华，邱金红

广州理工学院，广东 广州 510000

DOI: 10.61369/TACS.2026020018

摘 要： 目前，以生成式 AI 为代表的科技革新，使机器不但可以完成事先设定的指令，而且可以领会自然语言的需求，并将其转变成良好的代码，这就意味着软件开发由原来的“手工编写”转向了“意图驱动”的智能化时代。但是目前的实践大多只是一点工具的尝试，并没有对 AI 全链路赋能软件工程进行系统的探索，在如何处理 AI 生成代码的可解释性、保证复杂的业务逻辑准确性、重构传统的 DevOps 流程等各方面还存在着许多未知的问题。因此对人工智能在软件开发中具体的运用机制进行剖析，建立科学的实践评估体系，对于促进软件产业高质量发展有十分急迫的现实需求。

关 键 词： 人工智能技术；软件开发；应用；实践

Application and Practical Exploration of Artificial Intelligence Technology in Software Development

Huang Minghua, Qiu Jinhong

Guangzhou Institute of Science and Technology, Guangzhou, Guangdong 510000

Abstract : At present, technological innovations represented by generative AI have enabled machines to not only execute pre-set instructions, but also understand natural language requirements and convert them into high-quality code. This signifies that software development has shifted from the traditional "manual coding" to an intelligent era driven by "intent". However, most current practices are merely tentative uses of tools, without systematic exploration of AI's full-link empowerment of software engineering. There remain numerous unresolved issues in aspects such as addressing the interpretability of AI-generated code, ensuring the accuracy of complex business logic, and reconstructing traditional DevOps processes. Therefore, analyzing the specific application mechanisms of artificial intelligence in software development and establishing a scientific practical evaluation system are urgently needed to promote the high-quality development of the software industry.

Keywords : artificial intelligence technology; software development; application; practice

一、人工智能（AI）技术概述

人工智能（AI）是目前最前沿、最热门的技术之一，正在以前所未有的速度改变着全球经济、社会及文化氛围。依靠机器学习、计算机语言，达成超出人类精确度和速度的各种拟人操作，给各个行业的发展带来新的机会，计算机科技领域也是一样^[1-3]。基于大语言模型的 AI 编程助手可以很好地完成代码补全、错误检测、模块生成等任务，极大地提高了开发效率，并且减少了人为错误的发生。在测试环节中，AI 可以自动创建测试用例、预测边界条件、预测潜在缺陷，从而达到智能化质量保证的效果，运维阶段依靠 AIOps 实现异常检测、根因分析、自愈修复，提高系统的稳定性。可见，AI 也由辅助工具变成了协同开发者，加快了交付速度，而发了创新潜能，使软件工程进入了人机共生的新时代。

二、人工智能技术在软件开发中的应用现状

目前，人工智能技术已经深入到软件开发生命周期的各个环节当中，大大地改变了传统的研发方式。在代码生成和辅助编写方面，以大语言模型为基础的智能编程助手已经成为开发者们的标配工具，可以按照自然语言描述或者上下文代码来实时生成函数、单元测试甚至是完整的模块，极大地提高了编码效率并且降低了语法错误率^[4]。软件测试领域，AI 技术由原来的脚本执行向智能化发展转变，用强化学习自动生成覆盖边缘案例的测试数据，用计算机视觉做 UI 自动化回归测试，可以准确找到缺陷的原因，大大缩短了调试时间。进入到需求分析和系统设计阶段，AI 开始做一些文档整理、架构推荐的工作，可以从大量的历史项目中提取出最好的实践来帮助技术人员做技术选择和模块解耦^[5]。另外，在运维监控上，机器学习算法被用来做日志异常检测、性能

瓶颈预测以及故障自愈，从而达到由被动响应变为主动预防的目的。虽然目前的 AI 还没有完全取代人类工程师的决策，但是在重复性强、规则明确的编码、测试和文档维护等工作中，自动化程度已经很高，形成了以人机协同为主导的主流开发方式，大大缩短了软件交付的时间，也引起了关于代码版权、安全性和开发者技能转型等一系列的讨论。整体上呈现出技术快速更新和应用领域不断扩大的发展态势。

三、人工智能技术在软件开发中的应用与实践策略

（一）智能编码助手

人工智能技术正在改变软件开发的编码阶段，把传统的逐行编写方式变成智能辅助和人机协作的新模式。现代集成开发环境已经把基于大语言模型的代码生成工具深度整合进来，可以瞬间生成完整函数逻辑、类结构或者微服务架构，还可以在开发者输入前半句代码的时候准确预测后面的内容，大大减少重复性劳动和语法错误。智能编码辅助不是简单的文本补全，而是在大量开源代码库、最佳实践的基础上，根据上下文语境来推荐出符合项目规范的变量名、设计模式、异常处理等^[6]。开发者从繁杂的底层实现中解脱出来，把更多的精力投入到业务逻辑的革新以及系统架构的改善当中，从而让软件交付周期明显缩减。同时人工智能还可以对代码的质量进行审查，在编写代码的过程中就会自动给出性能瓶颈、安全漏洞、内存泄漏等问题以及对应的重构建议，边写边检的方式使质量控制提前到了开发的最前面，从而有效地降低后期测试和维护的成本^[7]。以 GitHub Copilot 在全球知名金融科技公司的应用为例，企业将 AI 编程助手深度集成至核心交易系统的开发流程中，结果显示后端微服务代码的自动生成率提升至40%，单元测试覆盖率从65%跃升至92%，且因语法错误导致的返工率降低了70%。显然，智能编码助手能显著缩短交付周期，从源头处提升金融级软件的安全性与稳定性。未来智能编码、智慧编程还会继续发展，实现创意与逻辑的高效转化。

（二）自动化测试革新

人工智能技术在软件测试、质量保证方面的应用，使软件测试由被动防御变为主动预测，创建起更为坚固、智能的测试体系。传统的测试用例编写依靠人工经验，不能包含所有的边缘情况，维护成本也较高，但是基于人工智能的测试工具可以根据代码变更历史和业务逻辑智能地产生包含正常路径、异常分支、边界条件的高覆盖率测试用例，模拟出人类测试人员无法想到的各种复杂的交互。用机器学习算法，可以不断监测测试执行的结果，自动找出并分类失败的原因，分为代码缺陷、环境变化、测试脚本本身的问题等，大大降低了误报率，并且加快了故障的发现速度。值得一提的是，AI 具有自我修复测试脚本的功能，当应用程序界面元素发生细微的改变或者接口参数发生变化的时候，智能代理会自动更新对应的选择器或者断言逻辑，从而保证持续集成流水线的稳定，不会因为 UI 频繁的迭代而造成测试脚本大面积失效的情况出现。另外，利用历史缺陷数据训练出的深度学习模型可以预测新的代码会带来多少 bug，把更多的测试资源投入到

高风险模块上，达到动态优化测试策略的目的^[8]。在国内头部电商平台的双 11 大促系统中，测试团队就引入了基于计算机视觉与强化学习的 AI 测试工具，成功解决了传统脚本因页面元素频繁变动而大规模失效的痛点。这一工具在两周内自动识别并修复了超过 3000 处因 UI 重构导致的定位失败，将回归测试执行时间从原来的 5 天压缩至 8 小时，并精准预测了 3 个潜在的并发崩溃风险点。此案例充分验证了 AI 技术在应对高频率迭代场景下，实现测试脚本自我进化与质量主动防御的巨大价值。可见智能化的改造提高了软件发布的频率和可靠性，让质量内建成为开发流程里不可或缺的关键部分。

（三）系统自愈运维

人工智能对于软件运维阶段的应用，是软件运营管理由人工响应型向自动化、智能化自愈型转变的关键一步，大大提高了系统的稳定性以及可用性。随着分布式架构以及微服务环境的不断复杂化，传统的依靠阈值告警、人工排查的运维方式已经不能满足要求了。AIOps 平台通过大量的日志、指标、链路追踪数据进行采集，使用无监督学习算法自动建立系统的基线模型，可以及时发现微小的异常波动，在故障出现之前发出准确的预警，将问题消灭在萌芽状态之中。当系统出现问题时，AI 引擎可以马上联系多维数据，自动生成原因分析，从上千个监控指标里找出引起问题的主要部分或者代码提交，把平均修复时间从小时级缩减到分钟级^[9]。中国移动相关数据中心在核心网管系统中部署了基于规则引擎与机器学习的自愈平台，重点解决服务器宕机与网络拥塞问题。数据显示，该系统上线一年内，累计自动执行故障诊断与恢复操作 3,200 余次，成功将网络中断事件的平均修复时间（MTTR）从人工介入的 40 分钟降低至 3 分钟以内，故障自愈率达到 85%。特别是在一次核心交换机板卡故障中，系统在 120 秒内自动完成流量迂回与备用链路切换，保障了全省 2,000 万用户的通信业务零感知，全年因此减少业务损失约 1,500 万元，显著提升了大规模基础设施的运维效率与稳定性。对运维知识库的不断学习、沉淀，智能助手可以给运维人员提供自然语言交互的查询、分析能力，快速找到解决方案，协助制定容量规划，使运维团队从繁琐的救火工作中解脱出来，专注于架构演进和技术革新，逐步构建起一个高度韧性、自适应的现代化软件运行环境。

（四）人机协作新范式

人工智能技术对软件开发全流程的渗透，不但是技术实现方式的改变，而且是团队协作模式、项目管理方法、伦理治理结构等各方面的全面革新。人机协作的新常态下，开发团队角色分工也随之变化，初级工程师依靠 AI 助手迅速成长为能够处理复杂任务的全栈开发者，资深架构师则更多的被定位为 AI 训练师、审核者，也就是对问题边界做出界定、对生成的代码质量进行评判、对整个系统方向进行把控。因此，项目管理也更加敏捷、数据化，依据以往的数据以及目前的进度来准确预估项目的风险，并且可以对资源进行合理的安排，还可以自动产生技术文档和用户手册，从而减轻了沟通的成本以及文档的维护工作。但是技术的飞跃也产生了新的问题和伦理考虑，开发者要警惕过度依赖 AI 工具造成的技能衰退，也要注意生成代码里隐藏的版权纠纷、算

法偏差以及数据隐私泄露等风险^[10]。建立完善的 AI 使用规范和审查机制就显得十分必要，企业要制定出明确的指南，在什么样的情况下可以使用 AI 生成代码、怎样进行安全性验证、知识产权归属问题应当如何处理等方面做出规定。未来软件开发将是一种人

类智慧与机器算力的融合，人类做富有创造性的想法、做出道德上的判断、做复杂的决定，机器去做繁重的执行、验证、优化等工作，让我们一起创建更加高效、可信、可持续发展的软件生态体系！

参考文献

[1] 李世玫, 余爱华, 王苗苗, 等. 基于人工智能压力性损伤图片分析系统的精细干预在神经内科重症患者中的应用 [J]. 河北医药, 2024, 46(20): 3194-3196+3200.

[2] 袁凯峰, 束梅玲. "人工智能+" 视域下高职电子信息类专业复合型人才培养新探 [J]. 南方职业教育学刊, 2024, 14(05): 31-39.

[3] 徐伟杰. 基于三维重建技术及人工智能的堵煤预警系统应用 [J]. 科学技术创新, 2024, (20): 44-47.

[4] 马晔风, 陈楠, 崔雪彬. 生成式人工智能技术如何影响专业型工作?——来自软件工程行业的早期证据 [J]. 劳动经济研究, 2024, 12(03): 3-34.

[5] 高倩. 人工智能在益智类计算机软件开发中的应用研究 [J]. 信息与电脑 (理论版), 2024, 36(11): 193-196.

[6] 王卫峰. 人工智能在软件管理与维护中的应用前景探讨 [J]. 信息与电脑 (理论版), 2024, 36(11): 218-220+228.

[7] 宋璞璇. 人工智能技术在信息化产品智能化设计与开发中的应用 [J]. 信息与电脑 (理论版), 2024, 36(07): 49-51.

[8] 薛梦丹. 基于人工智能的计算机应用软件开发技术应用分析 [J]. 中国高新科技, 2023, (13): 40-42.

[9] 毕然, 徐彤彤, 迟恺. 基于深度学习平台的人工智能软件开发与应用——以飞桨 (PaddlePaddle) 在火灾烟雾检测场景的应用为例 [J]. 警察技术, 2023, (03): 12-16.

[10] 张晓川. 人工智能在益智类计算机软件开发中的应用研究 [J]. 微型电脑应用, 2020, 36(09): 163-165.