低代码开发平台在信息技术项目中的优势与挑战

钱浙民
杭州钱南原水有限公司　浙江杭州　310001

摘　要：低代码开发平台（LCDP）作为一种以可视化拖拽、预制组件和自动化脚本为核心的开发工具，正逐步重塑信息技术（IT）项目的开发模式。本文从效率提升、成本优化、敏捷响应等维度分析其在 IT 项目中的核心优势，同时探讨技术局限性、数据安全、平台锁定等现实挑战，并提出混合开发模式、合规框架构建、生态开放化等应对策略。研究表明，低代码平台通过降低技术门槛、缩短交付周期，有效支撑企业数字化转型，但需在灵活性与稳定性、创新与合规之间寻求平衡，以实现可持续价值。

关键词：低代码开发平台；信息技术项目；开发效率；数字化转型；敏捷迭代

引言：

随着数字化转型加速，企业对 IT 项目的交付速度、迭代能力和成本控制提出更高要求。传统代码开发模式因周期长、技术门槛高、跨部门协作难等问题，难以满足快速变化的业务需求。低代码开发平台（Low-Code Development Platform，LCDP）应运而生，其通过“少编码、多配置”的方式，将复杂逻辑封装为可视化组件，使开发者（甚至业务人员）能快速构建应用系统[1]。据 Gartner 预测，到 2025 年，70% 的企业新应用将通过低代码平台开发，低代码已成为 IT 项目降本增效的关键工具。

1 低代码开发平台在 IT 项目中的核心优势

（1）开发效率呈指数提升：传统开发模式下，从需求分析到代码编写、测试、部署全流程需手动完成，耗时且易出错。而低代码开发平台通过三大核心机制缩短周期。一是可视化开发，开发者借助拖拽式界面和模块化组件（如表单、流程引擎、数据模型等）构建应用，减少超 70% 重复编码工作；二是引入自动化工具链，实现代码自动生成、一键部署和版本管理等功能，将部署周期从以周为单位压缩至以日为单位；三是并行协作机制，业务人员可参与原型设计，减少“需求 – 开发”沟通成本，实现“业务 – IT”协同开发。例如，某制造业企业用低代码平台开发供应链管理系统，开发周期从 6 个月缩至 45 天，效率提升近 400%[2]。这节省了时间和资源，让企业能更快响应市场变化，提升竞争力，使低代码平台成为企业数字化转型重要工具。

（2）降低技术门槛，释放业务创新潜力：通过降低技术门槛，我们可以充分释放业务创新的潜力。低代码平台的出现打破了传统意义上的“技术垄断”，使得那些非专业开发者，例如业务分析师和运营人员，也能够借助可视化工具自主构建轻量级应用。这种现象催生了一种新的开发模式，即“citizen developer（公民开发者）”模式。在这种模式下，简化编程逻辑成为可能，因为开发者无需掌握复杂的编程语言，而是通过配置化规则（例如条件判断、数据联动等）来实现业务逻辑。此外，低代码平台还赋予了业务部门更大的自主权，使得销售、财务等部门能够快速开发出定制化的工具，例如客户管理表和报销流程等，从而减少了对 IT 团队的依赖。最后，低代码平台还能够加速创新试错的过程，使得业务需求能够快速转化为原型，并通过小步迭代的方式验证其可行性，从而大大降低了创新的成本。

（3）敏捷响应业务变化，支持迭代优化：低代码平台天然契合敏捷开发理念，支持“快速构建 – 测试 – 反馈 – 优化”的闭环，从而确保业务需求能够迅速得到满足并持续改进。动态调整能力：通过可视化界面直接修改组件属性或流程逻辑，无需重构代码，适应业务规则频繁变化，使得业务团队能够灵活应对市场和客户需求的变化。灰度发布与回滚：支持增量更新和版本回溯，降低系统变更风险，确保在发布新功能时能够逐步验证其稳定性和可靠性，一旦出现问题可以迅速回滚到上一个稳定版本。数据驱动决策：内置数据分析模块，可实时监控应用运行数据，为迭代优化提供依据，帮助团队根据实际数据进行科学决策，从而不断提升应用性能和用户体验。

（4）成本控制与资源优化：通过实施有效的成本控制措施和资源优化策略，企业可以显著降低运营成本并提高资源利用效率。具体来说，人力成本的降低是一个重要的方面。通过减少对高级开发工程师的依赖，企业可以利用普通开发者或业务人员来完成基础开发任务。这种转变使得人力成本得以降低 30%–50%，从而为企业节省了大量的人力资源开支。此外，维护成本的减少也是一个显著的优势。通过采用标准化组件和自动生成的代码，后期维护的难度大大降低，故障排查的效率提升了 60%。这意味着企业在维护和故障处理方面的投入将大幅减少，进一步降低了运营成本。最后，硬件资源的节省也是一个不可忽视的方面。部分低代码平台支持云原生架构，能够实现按需弹性扩展，避免了服务器资源的浪费。这种灵活的资源管理方式使得企业可以根据实际需求动态调整硬件资源，从而避免了不必要的硬件开支，进一步优化了资源利用。

2 低代码开发平台面临的现实挑战

（1）技术局限性，复杂场景适配能力不足：尽管低代码平台在快速构建标准化应用方面表现出色，但在面对复杂场景时，其能力却显得有些力不从心。具体来说，功能深度有限：这些平台提供的可视化组件虽然方便易用，但在面对高度定制化的需求时，如大规模并发处理和复杂算法的集成，往往难以满足。性能瓶颈：自动生成的代码可能会包含不必要的冗余部分，这会导致系统响应速度变慢，资源占用过高，从而影响整体性能。集成难度：与 legacy 系统（例如老旧的 ERP 系统或定制化的数据库）进行集成时，通常需要额外开发专门的接口，这不仅增加了开发工作量，而且在兼容性方面也存在较大的挑战。

（2）数据安全与合规风险，平台自身安全漏洞：低代码平台作为第三方工具，可能存在权限管理不严、数据加密不足等问题；这些漏洞可能导致敏感数据泄露，给企业带来严重的安全威胁。此外，由于低代码平台的普及，攻击者可能会针对这些漏洞进行专门的攻击，进一步增加了安全风险。合规性挑战：医疗、金融等行业对数据隐私（如 GDPR、HIPAA）要求严格，低代码平台的预置功能可能无法满足定制化合规需求；这些行业对数据的保护有着严格的规定，如果低代码平台无法提供足够的定制化功能来满足这些需求，企业可能会面临合规风险，甚至可能面临罚款和声誉损失。知识产权争议：部分平台对生成代码的所有权界定模糊，可能引发企业与供应商的法律纠纷；在使用低代码平台生成的代码时，企业可能会面临知识产权的争议，因为平台对生成代码的所有权界定可能不够明确，导致企业与供应商之间产生法律纠纷 [3]。为了避免这种情况，企业在选择低代码平台时，需要仔细阅读合同条款，明确代码的所有权和使用权。

（3）平台锁定与生态依赖、供应商绑定：不同平台组件格式和数据模型常不兼容，企业更换平台需重构应用程序，迁移成本高昂。功能迭代受限：企业依赖供应商技术路线图，新功能更新速度难满足业务快速发展需求。生态封闭性：部分平台限制第三方插件接入，扩展功能边界困难。平台锁定使企业选技术平台需谨慎，更换平台面临巨大时间和经济成本。供应商绑定加剧依赖性，企业技术更新和功能迭代受限，影响业务灵活性和竞争力。生态封闭性限制功能扩展，还可能降低企业市场创新能力和竞争力，使其无法充分利用外部资源和工具提升产品和服务。

（4）团队技能转型与组织阻力：在快速发展的 IT 行业，团队技能转型是不可忽视的挑战。IT 团队角色重构明显，传统开发者要从“代码编写者”转变为“平台配置者”和“复杂功能开发者”，这一过程中团队成员需学习新技能和工具，会有阵痛。同时，业务团队信任问题突出，部分业务人员怀疑低代码应用稳定性和安全性，倾向传统开发方式，影响团队转型。此外，跨部门协作存在摩擦，公民开发者与 IT 团队职责划分不清，导致需求冲突或重复开发，影响组织效率和协同。为克服这些挑战，组织要加强内部沟通，明确职责分工，提供培训支持，促进团队技能转型，建立业务团队对新技术的信任，确保跨部门协作顺畅。

3 应对策略：平衡优势与挑战的路径

（1）采用“低代码 + 传统代码”混合开发模式：这种模式可实现高效灵活的软件开发。开发过程分不同层次，各有特定开发方式与工具。标准化功能如表单、流程等，可通过低代码平台快速实现，其可视化开发环境降低开发门槛、提高效率，非专业人员也可参与。而复杂功能如核心算法、高并发模块等，仍需传统代码开发，以满足定制化和性能要求，开发后可集成到低代码平台，与其他功能无缝对接。API 开放化是重要环节，选择支持自定义 API 的低代码平台，可与 AI 引擎、物联网设备等外部系统集成，实现数据和功能交互。总之，该模式能利用低代码平台便捷性与传统代码

灵活性，实现高效、灵活且可扩展的软件开发，既适合搭建标准化功能，也能应对复杂功能开发，还能与外部系统灵活集成，为企业数字化转型提供支持。

（2）构建全流程安全合规体系：为了确保平台的安全性和合规性，我们将严格化平台选型，优先选用已经通过ISO 27001、SOC 2 等国际权威认证的平台[4]。在选择平台时，我们将全面评估其数据加密、权限管理及审计日志功能，确保这些关键功能能够满足我们的安全需求。此外，我们还将自定义合规规则，借助平台的扩展能力，开发合规插件，如数据脱敏、访问控制策略等，以满足行业特定需求。为了确保平台的持续安全，我们将定期进行安全审计，对低代码应用实施渗透测试和漏洞扫描，及时排查并修复安全隐患，确保平台的安全性和合规性。

（3）选择开放式平台，降低锁定风险：为了确保技术中立性，我们应当优先选择那些支持主流技术栈（如 Java、Python）并且兼容各种开源组件的平台。这样做的目的是为了避免因依赖私有协议而受到限制，从而降低技术锁定的风险。此外，在签订合同时，我们需要明确数据主权，确保在合同中清晰界定数据所有权、迁移权限及相关成本。这样可以保障企业对核心数据的掌控，避免在未来出现数据迁移或数据主权方面的纠纷。最后，我们还应当积极参与低代码平台的开发者社区，通过与社区成员的互动和合作，推动平台功能的持续迭代和标准化建设。这样不仅能够促进生态协同，还能确保平台的长期发展和企业的持续受益。

（4）推动组织能力与文化转型：通过实施人才双轨培养计划，我们致力于提升整个组织的技术能力和文化氛围。具体来说，我们将对 IT 团队进行系统的低代码平台培训，使他们能够熟练掌握这一新兴技术。与此同时，我们也将向业务团队传授基础的开发逻辑，以便他们能够更好地理解技术团队的工作，并在必要时提供支持。通过这种方式，我们旨在建立一个“IT+ 业务”协同开发的机制，促进跨部门的沟通与合作。为了确保低代码平台的顺利推广和应用，我们将采取试点先行、逐步推广的策略。首先，我们会选择一些非核心业务作为试点项目，例如内部管理工具等，通过这些项目的实践来验证低代码平台的价值和效果。在取得初步成效后，我们再逐步将低代码平台扩展至核心系统，确保其在关键业务中的稳定性和可靠性。另外，为了确保低代码平台的有序开发和应用，我们将建立一个完善的治理框架。这包括明确公民开发者的权限范围，确保他们在开发过程中不会超出既定的权限，从而避免潜在的安全风险。同时，我们还将制定应用上线的审批流程，确保每一个应用在上线前都经过严格的审查和评估，避免无序开发导致的系统混乱和潜在问题。通过这些措施，我们希望能够建立一个高效、有序且可持续发展的低代码开发环境[5]。

4 结论

低代码开发平台通过“可视化、自动化、平民化”的特性，为信息技术项目提供了效率革命的新路径，尤其在数字化转型背景下，成为企业快速响应市场、降低创新成本的核心工具。然而，其技术局限性、安全风险与组织挑战亦不容忽视。未来，企业需以“场景适配”为导向，通过混合开发、合规治理、生态开放和组织转型，最大化低代码平台的价值，同时规避潜在风险。只有将低代码工具与企业战略、技术架构、人才体系深度融合，才能真正释放其在 IT 项目中的变革力量，加速数字化转型进程。

参考文献：

[1] 郭星明 , 马荣飞 , 李金营 . 信息系统的低代码开发 [M]. 浙江工商大学出版社 :202304:274.

[2] 苏伟 , 冯宽 . 基于自动化引擎的工业软件低代码开发平台技术架构研究 [J]. 信息技术与标准化 ,2025,(05):8-12.

[3] 于潇 , 孙立生 , 孙梦椰 , 等 . 低代码开发平台对用户体验、个性化定制能力提升的创新研究 [J]. 科技创新与应用 ,2025,15(09):13-16.

[4] 吴金津 , 程玉柱 , 梁宇 . 基于低代码开发平台的普通话报名微系统设计与实现 [J]. 湖南邮电职业技术学院学报 ,2025,24(01):32-38.

[5] 刘平 . 基于低代码开发平台的教学案例研究 [J]. 电脑知识与技术 ,2025,21(03):145-147.

作者简介：钱浙民（1991-），男，汉族，浙江，湖州，本科，高级工程师，研究方向：信息化建设、企业数字化转型、物联网、人工智能。