

基于大语言模型 RAG 的软件工程项目管理探索

卢军 杜翠 徐露 黄健 李代伟

Lu Jun, Du Cui, Xu Lu, Huang Jian

(成都信息工程大学软件工程学院 四川 成都 610225)

(School of Software Engineering, Chengdu University of Information Technology, Sichuan Chengdu 610225, China)

摘要: 大语言模型已经深刻地改变了软件工程。本文全面且深入地对基于大语言模型检索增强生成 (RAG) 技术在软件工程项目管理中的应用研究, 深入阐述了其在需求管理、进度跟踪、风险管理等软件工程项目管理关键环节的具体应用方式, 以及由此带来的显著改进。同时, 通过引入详实的实际案例进行深度分析, 有力地验证了该技术的有效性, 为软件工程项目管理向智能化方向转型提供了坚实的理论与极具价值的实践参考。

关键词: 大语言模型; RAG; 软件工程; 项目管理

一、引言

在软件工程项目管理中各种频繁变化的因素为软件工程项目管理的稳定性和连续性带来了巨大挑战。项目进度把控缺乏

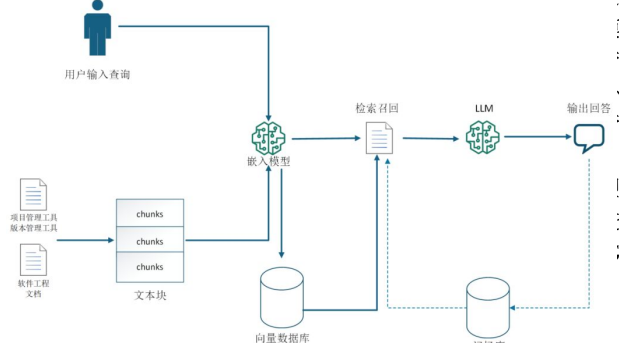


图 1 在软件工程项目管理中应用大语言模型 RAG

如图 1 所示, RAG (Retrieval-Augmented Generation) 技术即检索增强生成技术, 它通过将外部知识源整合到大型语言模型 (LLM) 的生成过程中来提高其能力。

RAG 从外部数据库或文档中检索相关信息, 并使用这些信息生成更准确、更符合上下文的响应。如图 1 所示, 通过在 RAG 中集成软件工程项目管理相关的文档与代码, RAG 极大地提升了 LLM 应用于软件工程项目管理的能力。通过在 RAG 中引入软件工程项目文档与代码, 包含项目需求文档、设计文档、技术规范、Git 信息、项目源代码等, 也可以是专业领域的数据库, 如软件行业的技术知识库、开源代码库等, LLM 能够访问软件工程项目的实时信息, 从而提高其响应的相关性和准确性。从自身软件工程项目知识源中检索到相关的算法描述、代码示例以及应用案例等信息, 大语言模型结合这些信息, 能够给出更详细、更符合实际需求的解答, 有效弥补了大语言模型在知识更新不及时和本软件工程项目知识不足的问题^[3]。

三、基于大语言模型 RAG 的软件工程项目管理应用

(一) 需求管理

(1) 需求理解与分析: 在软件项目需求获取阶段, 基于 RAG 的系统能够利用其强大的文本处理能力, 快速读取并解析需求文档, 结合大语言模型对这些内容进行深入理解和分析, 生成清晰、详细的解释。大语言模型还可以通过可视化的方式, 如生成流程图或思维导图, 将复杂的需求逻辑直观地呈现给开发人员, 帮助他们更好地理解需求, 减少因理解偏差导致的开发错误。

(2) 需求变更管理: 当需求发生变更时, 系统首先对变更内容进行语义分析, 理解变更的核心要点和影响范围。然后, 通过检索与之相关的原有需求、设计文档以及代码实现, 利用

大语言模型的推理能力, 全面评估变更对项目进度、成本和技术实现的影响, 评估所需的开发工作量和时间, 从而为项目管理者提供准确的决策依据。

(二) 进度跟踪

(1) 任务进度预测: 基于大语言模型 RAG 的系统可以通过对项目历史数据、当前任务状态以及团队成员工作效率等多维度信息的综合分析, 实现对项目任务完成时间的精准预测。大语言模型可以结合项目文档与源代码、Git 信息数据等数据, 运用时间序列分析、机器学习等算法, 预测该任务的剩余时间, 为项目管理者提供多种可能的进度调整方案, 如增加资源、调整任务优先级等, 帮助他们及时做出合理的决策。

(2) 进度偏差分析: RAG 技术可以帮助系统全面检索项目日志、沟通记录、代码提交记录等信息, 结合大语言模型强大的自然语言处理和分析能力, 深入挖掘导致进度偏差的关键因素, 并针对这一关键因素提供相应的解决建议。

(三) 风险管理

(1) 风险识别: 基于 RAG 的风险管理系统具备强大的风险识别能力, 它可以全面读取项目相关的各类文档, 包括需求文档、设计文档、测试文档、用户手册等, 利用大语言模型对风险的理解和语义分析能力, 结合从行业风险知识库中检索到的常见风险类型和案例, 识别潜在的风险因素。通过这种方式, 系统能够帮助项目管理者提前发现潜在风险, 为风险应对争取时间。

(2) 风险应对策略生成: 在识别风险后, RAG 技术能够发挥其强大的知识检索和生成能力, 为项目管理者生成针对性的风险应对策略。系统首先从开发团队内部的风险案例库和行业最佳实践中检索以往项目的风险应对案例, 然后结合当前项目的实际情况, 如项目规模、技术架构、团队能力等, 利用大语言模型的生成能力, 制定出符合项目特点的风险应对策略。

(四) RAG 与 Git 工具的结合

Git 作为广泛使用的版本控制系统, 将 RAG 技术与 Git 工具相结合, 能为项目管理带来新的助力^[4]。RAG 技术可以通过检索 git commit 信息, 将其与软件工程管理中的各个关键内容进行关联, 这种联系能够让 RAG 清晰地感知到每个需求是如何一步步转化为实际代码。Git 代码变更信息详细记录了代码的修改内容, RAG 可以分析这些变更信息, 结合需求变更管理中的内容, 判断代码变更是否准确实现了需求变更。RAG 利用 git commit history 信息还可以追溯项目在不同阶段的开发情况。当项目出现问题需要进行回溯分析时, RAG 能够根据 commit history 信息, 结合大语言模型的分析能力, 快速定位到可能导致问题的代码变更。

四、RAG 在不同开发模型中的应用

在软件工程领域，不同开发模型各有特点，而 RAG（检索增强生成）在其中均展现出独特价值。如表 1 所示，在不同的软件开发开发模型中，RAG 都能起到显著的作用。

从开发流程的角度来看，瀑布模型强调文档的规范性与完整性，RAG 能在需求变更或问题排查时，快速定位关键信息，提升文档处理效率，打破传统文档处理的僵化模式。敏捷开发注重灵活性，RAG 可对简略文档进行智能补充，适应频繁的需求变更，减少人工干预。迭代模型中，RAG 能够分析迭代阶段文档的变化与延续性，为团队理解项目演进提供有力支持。然而，RAG 在应用中也面临一些挑战，如对不同开发模型下文档格式与内容的适应性优化，以及如何在复杂的项目情境中进一步提升信息检索与生成的准确性等问题，值得后续深入研究与探索，以更好地发挥 RAG 在软件开发中的作用^[5]。

表 2 不同软件开发开发模型中 RAG 应用的特点

开发模型	文档特点	RAG 在模型中的应用特点
瀑布模型	从需求分析到测试维护，各阶段均生成大量详细文档，这些文档规范、完整，记录项目全流程信息，如需求规格说明书、设计文档、测试计划与报告等，是后续阶段开展的重要依据。	1. 能快速检索和分析瀑布模型各阶段生成的大量文档，当出现需求变更或问题排查时，可迅速定位相关内容，提高决策效率。例如在设计阶段发现需求理解偏差，RAG 能从需求文档中精准提取关键信息辅助重新评估。 2. 在项目后期若需对文档进行修改或整合，RAG 可根据新需求智能生成修改建议或整合方案，突破传统文档处理的僵硬模式，提高文档处理的灵活性。
敏捷开发	重视可工作软件，文档相对简洁，主要记录关键信息，如用户故事、迭代计划等，多作为团队内部沟通协作工具，很多信息靠面对面交流传递。	1. 敏捷开发文档简略，RAG 可利用其强大的语言理解和生成能力，对简略文档进行智能补充和完善，比如根据用户故事和迭代过程，生成更详细的技术实现说明，方便新成员快速上手。 2. 面对敏捷开发中频繁的需求变更，RAG 可实时更新和调整文档内容，自动关联变更点与相关文档，确保文档与实际开发保持一致，无需人工手动繁琐修改。
迭代模型	每个迭代生成相应文档，内容随迭代更新完善，详细程度介于瀑布模型和敏捷开发之间，记录迭代关键信息，为后续迭代提供参考，但不像瀑布模型文档那样追求全面详尽。	1. 在迭代过程中，RAG 可对不同迭代阶段的文档进行智能分析和比对，快速识别出需求、设计等方面的变化和延续性，帮助团队更好地理解项目演进过程，如分析出迭代中功能改进的相关文档信息，助力后续迭代优化。 2. 当迭代中出现需求变更或调整迭代计划时，RAG 能智能调整文档结构和内容，自动生成变更日志和影响分析报告，提高文档处理的柔性和效率，保障项目文档的连贯性和准确性。

五、结论与展望

基于大语言模型 RAG 的技术在软件工程项目管理的需求管理、进度跟踪和风险管理等关键环节展现出了显著的应用价值。通过巧妙地结合外部知识源和大语言模型的优势，该技术能够高效处理海量的项目信息，准确理解项目需求和问题，提供精准的分析和决策支持，从而大大提高了项目管理的效率和准确性。在实际项目中，有效帮助项目管理者更好地应对各种复杂多变的挑战，保障项目的顺利进行。

未来的研究可以进一步拓展大语言模型 RAG 在软件工程项目管理中的应用场景。在团队协作管理方面，可以利用 RAG 技术实现智能的任务分配和团队成员之间的沟通协调，根据成员的技能水平、工作负荷和项目需求，合理分配任务，提高团队协作效率。在成本管理领域，通过对项目成本数据的分析和知识检索，结合大语言模型的预测能力，实现对项目成本的精准估算和动态监控，及时发现成本超支风险并提供应对策略。同时，随着数据安全和隐私保护问题日益受到关注，需要深入研究如何更好地整合不同类型的知识源，在保障数据安全和隐私的前提下，充分发挥 RAG 技术的优势。例如，采用联邦学习、同态加密等技术，实现数据在不同机构或部门之间的安全共享和协同处理，为软件工程项目管理的智能化发展提供更坚实的技术保障。

参考文献：

[1]Liu J A H .Adaptive Software Development: A Collaborative Approach to Managing Complex Systems[J].scitech book news, 2000.

[2]王海涛,师杨坤.基于大语言模型的查询扩展方法研究[J].计算机技术与发展[2025-01-25].

[3]张永健,陈可洲,王 涛,等.结合大语言模型和知识感知重构的图增强服务推荐方法[J].计算机科学与应用, 2024, 14(11):12.DOI:10.12677/csa.2024.1411217.

[4]Andersson R , Ericsson M , Wingkvist A .Mining Relations from Git Commit Messages : an Experience Report[J]. 2014.

[5]Shao Y , Huang Y , Shen J ,et al.Vortex under Ripplet: An Empirical Study of RAG-enabled Applications[J]. 2024.

作者简介：

卢军（1974-），男，汉族，四川成都人，博士，教授，主要研究方向：软件工程与人工智能、嵌入式系统与智能软件。

杜翠（1985-），女，汉族，河南柘城人，博士，副教授，主要研究方向：知识表示与推理等。

徐露（1994-），汉族，四川成都人，博士，讲师，主要研究方向：人工智能与数据分析等。

黄健（1976-），男，汉族，四川成都人，硕士，副教授，主要研究方向：计算机辅助工业软件等。

李代伟（1976-），男，汉族，四川达州人，博士，教授，主要研究方向：数据库与知识工程、人工智能、大数据分析与应用。

基金情况（若无基金可以不填写）

成都信息工程大学面向产教融合与科教融合的《嵌入式系统》课程教材与相关教学仪器建设、开源软件在软件工程教育教学综合改革中的实践与探索、人工智能驱动下的软件工程专业改革支持。