

敏捷项目管理中的质量和进度控制要点

李哲愚

中国电子科技集团公司第十四研究所 南京 210000

摘要：随着敏捷项目管理方法的普及，质量控制与进度控制日显重要，成为项目制胜的关键。本文剖析了敏捷项目中质量与进度的管控精髓，在质量管理方面，强调了客户深度参与、持续集成流程、自动化测试技术、代码严格审查以及团队知识共享等实践；在进度管理上，提出了目标设定、计划周密制定、实时进度追踪、定期效果评估和灵活应变等策略。通过融合实用方法与高效工具，阐释了如何在多变环境下确保项目质量、控制进度，从而实现客户价值的提升与团队效率的提高。

关键词：敏捷项目管理；质量控制；进度控制；持续集成；自动化测试

1. 敏捷项目管理定义

敏捷项目管理，作为一种深刻体现人本精神的管理方法，其核心在于通过迭代与增量的方式推动项目进程，从而实现对变化的迅速响应。这一理念的根基可追溯到 2001 年颁布的《敏捷宣言》，其中明确倡导个体与互动、可用的软件、与客户的紧密协作以及对变革的敏锐洞察，相较于传统模式下的工具依赖、僵化流程和固定计划，更具灵活性和前瞻性。通过巧妙地将庞大项目拆解为若干小型、可控的迭代周期，敏捷方法使团队得以高频次地交付成果，并借助持续的客户反馈来不断优化工作流程和提升产品质量。此外，敏捷原则所强调的适应性、透明度和高效沟通，不仅为质量控制和进度管理构筑了坚实基础，更在软件开发领域展现了广泛应用价值，其影响力还逐渐渗透至市场营销、产品设计等多个行业领域。

2. 敏捷项目管理中的质量控制要点

2.1 客户参与和需求管理

在敏捷项目管理的实践中，客户在此框架下的角色远超出了单纯的需求提供者，他们深入融入团队，成为项目全生命周期的紧密伙伴，展现在用户故事的精心构建、优先级的审慎排序以及迭代过程的细致评审之中。用户故事，作为捕捉客户需求的关键工具，采用平易近人的非技术语言，透过“用户角色 - 行为 - 目标”的清晰结构，助力团队精准把握需求的本质。故事点的引入，为工作量的量化评估提供了有力支持，使得资源的合理分配与开发周期的准确预估成为可能。而优先级的明确排序，则进一步确保了高价值任务

能够优先得到执行，从而最大化项目的整体效益。为确保需求的准确无误，团队与客户之间的沟通频率被提升至新的高度。借助 Jira、Trello 等先进的协作工具，团队能够实时记录和跟踪需求的每一个细节，从而确保项目进程的透明度和可追溯性。此外，面对客户需求的不可避免的变化，团队建立了一套灵活且高效的计划管理机制。当需求变更发生时，敏捷框架所特有的迭代机制使得团队能够迅速在下一个周期做出相应调整，而无需担心整体进度的紊乱。这种以客户为中心的敏捷机制，不仅提升了项目的交付质量，更增强了客户对项目成果的认同感和满意度。

2.2 持续集成与持续交付

持续集成（CI）与持续交付（CD）在敏捷项目管理中占据着举足轻重的地位，它们是提升项目质量、减少失误的关键实践。持续集成的核心理念在于将开发者的代码改动频繁且无缝地集成至主分支，借助自动化的构建与测试流程，迅速识别并解决代码冲突与潜在缺陷。这一过程离不开强大的工具链支持，如 Jenkins、GitLab CI/CD 或 CircleCI 等，它们能够在每次代码提交后即时触发构建与测试，从而持续保障代码的稳定性和可用性。持续交付则是对持续集成理念的进一步延伸，它侧重于确保经过验证的代码能够顺利、安全地部署至生产环境。通过精心设计的自动化部署流水线，团队能够显著降低人为操作所带来的潜在风险，并大幅提升交付效率。此外，为了维护团队协作的有序性，实施 CI/CD 过程中还需辅以严格的代码分支管理策略，如广泛采用的 Git Flow 模型。同时，通过监控构建成功率、部署频率等关键

指标，团队可以实时评估 CI/CD 实践的有效性。

2.3 自动化测试

在敏捷项目管理中，自动化测试是保障项目质量和提升效率的关键工具，更通过系统化、自动化的验证机制，有效减少了人为错误，提升了测试覆盖率。自动化测试主要包含三大类型：单元测试、集成测试以及端到端测试，它们分别从不同层面确保了项目的质量。单元测试专注于验证最小功能单元的正确性，常常借助 JUnit、TestNG 等成熟框架来进行；集成测试则着眼于各模块之间的协同工作，确保它们在组合后能够正常运作；而端到端测试则通过模拟用户的实际操作行为，全面检验系统从输入到输出的整体流程是否顺畅无阻。为了实现自动化测试的高效运作，选择合适的工具和框架显得至关重要。例如，针对 Web 端测试，Selenium 提供了强大的支持；而在验证 API 接口方面，Postman 则展现出了其独特的优势。此外，设计可维护性强的测试用例也是成功实施自动化测试的基础，需要在设计测试用例时避免过度复杂化和不必要的重复依赖，同时确保测试代码能够随着业务逻辑的变化而及时更新。

2.4 代码审查与知识共享

代码审查的核心宗旨在于通过深入细致的审查，挖掘出代码中潜在的问题与风险，如语法错误、编码风格的不一致，或是性能上的瓶颈等。在代码审查的实施过程中，通常采用自动化与人工相结合的方式，以确保审查的全面性与深入性。例如，可以利用 SonarQube 这类工具进行静态代码分析，自动检测出代码中的常见问题。同时，团队成员之间的相互审阅与反馈也是不可或缺的环节，它能够更加精准地定位并解决那些自动化工具难以发现的问题。为了确保代码审查的高效与规范，需制定明确的标准与流程。例如，遵循 PSR-12 或 Google Java Style 等行业编码规范，能够使代码保持高度的一致性与可维护性。此外，审查流程应与项目的开发周期紧密相连，确保在代码合并至主分支之前完成必要的审查工作，从而避免问题的累积与扩散。值得一提的是，代码审查不仅仅是一种技术层面的提升手段，它更是团队内部知识共享与传递的桥梁。在审查过程中的交流与讨论，使团队成员有机会学习到更多的最佳实践与设计模式，从而提升整个团队的技术实力。为了进一步巩固与拓展团队的知识库，还可以定期组织技术分享会，并将代码规范文档化，以便为持续优化代码质量提供有力的长期支持。

2.5 持续改进与精益思维

持续改进的实施，往往以迭代回顾会议为轴心展开。每当一个迭代周期落幕之际，团队成员便会齐聚一堂，共同回顾本次迭代的成果与经验，深入挖掘值得发扬的成功实践，以及亟待改进的不足之处。为确保改进措施能够落到实处，应借助 SMART 原则，即目标必须具体、可测量、可实现、与其他目标相关且具有明确的截止日期来制定详尽的行动计划。在持续改进的道路上，度量指标是评估改进成效的重要依据，如通过观测量板流程中的平均交付周期，或是代码质量相关的缺陷密度等指标，能够为下一步的改进方向提供科学的数据支撑。与此同时，精益思维要求减少浪费，将精力聚焦于真正创造价值的工作上。无论是过于繁琐的流程、低效的工具，还是资源的无端耗费，都需要持续识别和剔除的非增值活动。此外，应勇于尝试新的工具或实践方法，如测试驱动开发（TDD）或行为驱动开发（BDD），并通过收集与分析数据来验证它们的实际效果，借此提升项目的灵活性与执行效率，助力团队营造出一种积极向上的学习型文化氛围。

3. 敏捷项目管理中的进度控制要点

3.1 明确项目目标和范围

在敏捷项目管理中，确立清晰的项目目标和范围奠定了项目的基石，也为进度把控提供了指向。若目标含糊或范围不明，项目便可能频繁变更路径，徒增时间成本与资源浪费。制定目标时，应综合考虑客户需求、业务价值及项目实施的可行性，并以具体可量化的成果为指引。例如，在电商应用开发中，可将“提升用户下单转化率”作为目标，并进一步细化为“三个月内，借由新增推荐功能，使转化率提升 15%”。同时，定义范围时既要涵盖主要功能，也要明确排除某些内容，如将部分功能延后至后续迭代。利用需求文档和用户故事地图等工具，可协助团队与客户共同理清需求脉络。为防范围失控，团队可采取“一增一减”策略，即每新增一需求，便舍弃一低优先级需求，以保持范围的动态稳定。目标与范围一旦界定，后续规划与进度监控便有了明晰的框架，从而有效降低因目标偏离而导致的项目延误风险。

3.2 合理的项目进度计划

相较于传统项目计划对初期详尽预测的过度依赖，敏捷项目管理则推崇动态规划，以便项目计划能根据项目进展和实时反馈进行持续优化。在制定计划之初，就应根据项目

的具体范围和团队的执行能力，明确每个迭代周期内应完成的交付目标，如优先实现的一组用户故事。利用看板之类的工具，可以通过清晰的列状态（待办、进行中、已完成）来实时监控任务的进展情况。而燃尽图则能以一种直观的方式展示剩余工作量如何随时间推移而减少，从而协助团队及时识别出可能存在的延误风险。在规划过程中，应全面考量团队的实际执行力和任务的复杂程度，并运用时间盒技术为每项任务设定明确的时间限制，以杜绝不必要的拖延现象。同时，计划中还应预留出一定比例的缓冲时段，如分配 10%–20% 的时间来应对可能出现的变更请求或技术难题。通过采纳滚动式规划策略，可促进团队聚焦于当前迭代周期内的任务，同时为未来的迭代预留出充分的调整空间。

3.3 实时跟踪项目进度

实时跟踪进度关乎管理，是一种主动的风险防控手段，助力团队敏锐捕捉偏差、迅速调整策略。在敏捷实践中，这依赖于精准数据与高效沟通的双重支撑。借助自动化工具 Jira 或 Azure DevOps，团队得以实时掌握任务进展，洞悉哪些任务正顺利推进，哪些已遭遇瓶颈。这种透明化跟踪有效减少信息延迟，提升团队对全局进度的把控力。关键指标如速度和任务完成率，为团队提供实际生产率的客观衡量。一旦发现迭代完成率低于预期，便迅速揭示出需求理解或技术方案的潜在问题，促使团队即刻介入、深入剖析。每日站会作为跟踪的核心，通过简洁的三问法快速定位问题、调配资源。同时，与客户保持实时同步，确保他们对项目进展了如指掌，显著增强项目的透明度和信任度。

3.4 定期进行项目进度评估

定期评估项目进度，既是对工作成果的全面检视，也是发现和解决根本性的重要途径。每次评估都应聚焦于两大核心目标：验证目标的达成情况，以及审视当前工作方法的效能。数据在此过程中发挥着关键作用，如分析燃尽图以预测剩余工作量与计划时间的契合度，或利用速度数据来评估团队生产率的稳定性。评估会议应由产品负责人主导，并邀请开发、测试团队及项目利益相关者共同参与，以确保讨论内容的深入和全面。在评估中，应重点关注实际存在的问题，深入剖析未完成的任务背后的原因，无论是技术难题、需求变动，还是团队协作的障碍。当然，评估的结果不应仅

停留在问题层面，而应转化为切实的行动指南，如调整任务优先级、重新配置资源或改进工作流程。通过将定期评估融入每个迭代周期，团队能够在快速迭代中实现精准调整与持续进步，从而确保项目进度始终在掌控之中。

3.5 灵活应对变化

灵活应变能力既是敏捷项目的核心，也是保障项目进度的关键。在项目中，需求与环境的变动难以避免，而如何应对变动则决定着项目的成败。敏捷方法借助迭代式开发与动态优先级设定，为变动提供了应对空间。例如，客户中途提出新需求时，团队可迅速调整产品待办列表，将新需求纳入下一迭代周期，确保当前任务不受干扰。这种快速响应机制避免了混乱调整，而自组织团队的每个成员均能根据变动迅速调整工作计划，甚至主动提出更高效方案。同时，灵活性需有明确界限，以防无节制满足需求导致项目范围失控或资源匮乏。

4. 结语

本文从质量和进度两个核心角度，深入剖析了敏捷项目管理的关键要点与实践策略。在质量控制上，通过客户的深度介入、持续集成交付、自动化测试及代码严格审查等措施，确保了项目成果的高可靠性与一致性。而在进度管理方面，则运用明确目标设定、科学规划布局、实时进度追踪以及灵活应变等策略，保障项目在多变环境中的平稳推进。展望未来，随着技术的不断革新与敏捷理念的更广泛应用，如何巧妙结合智能工具、数据驱动分析及高效团队协作，将成为进一步优化敏捷项目管理的关键所在。

参考文献：

- [1] 姚佳人. 敏捷项目管理方法与传统项目管理方法的比较研究 [J]. 机械工业标准化与质量, 2023(12)
- [2] 周爱龄. Z 项目敏捷开发进度管理研究 [D]. 电子科技大学, 2024
- [3] 蒋帆. 敏捷项目管理中的质量和进度控制要点 [J]. 建设监理, 2024(07)

作者简介：

李哲愚（1992–），性别：男，民族：汉，籍贯：山东省枣庄市，学历：硕士研究生，职称：工程师，研究方向：项目计划与控制、项目管理的敏捷方法。