**项目质量控制**

（一）软件项目跟踪控制计划：

针对本项目可以采用目标问题度量法进行软件质量跟踪控制。该方法通过设定明确的质量目标和问题度量标准，对软件开发过程中的质量进行连续监视和控制，确保软件质量符合预期要求。

1.软件项目跟踪对象

·项目风险：包括技术风险、进度风险、资源风险等。

·项目进展：各阶段任务的完成情况。

·开发活动进展：开发团队的工作进度和效率。

·开发活动问题：开发过程中遇到的技术问题和非技术问题。

·项目展望：对未来项目发展的预测和规划。

2.软件项目跟踪基础

·软件开发计划：包括活动安排、进度计划、资源和人员计划、成本计划等。

·软件项目实际实施过程：记录实际进度、面临的问题和解决方案。

3.软件项目跟踪方式

·成立项目跟踪小组PTT，由项目组成员或负责人组成，负责跟踪项目进展和协调各方工作。

·指定项目跟踪负责人，负责协调项目进度的跟踪工作，确保信息的准确性和及时性。

·定期召开项目跟踪会议，每周一次，了解项目实施情况，汇报问题，制定解决方案。

4.质量目标和度量标准

·需求满足度：确保软件功能满足用户需求，需求变更得到有效管理。度量标准包括需求实现率、需求变更控制率等。

·代码质量：确保代码规范、可读性强、易于维护。度量标准包括代码覆盖率、代码重复率、代码复杂度等。

·测试质量：确保测试用例全面、测试过程规范、测试结果准确。度量标准包括测试通过率、缺陷密度、测试覆盖率等。

·性能表现：确保软件在响应时间、吞吐量、资源利用率等方面表现良好。度量标准包括响应时间、吞吐量、CPU使用率等。

（二）质量阶段跟踪控制：

|  |  |
| --- | --- |
| 项目阶段 | 质量跟踪控制措施 |
| 项目启动阶段 | 质量目标设定：明确项目的质量目标，如系统的稳定性、安全性、易用性等，并设定可量化的质量指标。  质量计划制定：制定详细的质量计划，包括质量检查点、测试策略、审计计划等，确保后续阶段有章可循。 |
| 需求分析阶段 | 需求质量评审：对需求文档进行评审，确保需求的完整性、准确性、一致性和可测试性。  需求变更管理：建立需求变更控制流程，对需求变更进行记录、评估和批准，确保变更得到妥善处理。 |
| 概念设计阶段 | 设计质量评审：对系统设计文档进行评审，确保设计符合需求规格，具有可行性、可扩展性和可维护性。  风险识别与应对：识别设计过程中可能存在的风险，并制定相应的应对措施。 |
| 技术审评阶段 | 技术可行性评估：评估所选技术方案的可行性，包括技术成熟度、兼容性、性能等。  技术文档审查：审查技术文档，确保其清晰、准确、完整，便于后续开发和维护。 |
| 软件开发阶段 | 代码质量监控：通过代码审查、代码走查等方式，确保代码质量符合编码规范和质量标准。  单元测试执行：对每个模块进行单元测试，确保其功能正确性和边界条件处理。 |
| 集成测试阶段 | 集成测试计划：制定详细的集成测试计划，包括测试环境、测试用例、测试数据等。  缺陷跟踪与管理：使用缺陷跟踪系统记录、跟踪和修复测试过程中发现的缺陷。 |
| 性能优化阶段 | 性能测试：对系统进行性能测试，包括响应时间、吞吐量、资源利用率等，确保系统满足性能要求。  性能调优：根据性能测试结果，对系统进行性能调优，提高系统性能。 |
| 用户验收测试阶段 | UAT计划制定：制定用户验收测试计划，明确测试目标、测试范围、测试环境等。  用户反馈收集：收集用户对系统的反馈意见，确保系统满足用户需求。 |
| 上线准备阶段 | 上线前质量检查：对系统进行全面的质量检查，确保系统稳定、安全、可靠。  回滚计划制定：制定上线后的回滚计划，以应对可能出现的紧急情况。 |
| 正式上线阶段 | 上线监控：对系统上线过程进行监控，确保上线过程顺利。  上线后质量跟踪：对系统上线后的质量进行跟踪，收集用户反馈，及时处理可能出现的问题。 |
| 后期维护阶段 | 系统监控：对系统进行持续监控，确保系统稳定运行。  问题处理：及时处理用户反馈的问题，确保系统质量。  持续改进：根据用户反馈和质量监控结果，持续优化系统，提高系统质量。 |

1.贯穿整个项目周期：

·设立质量指标：在项目启动阶段，明确质量目标，如系统稳定性、响应时间、用户满意度等，并设立可量化的质量指标。

·定期监控与评估：从需求分析到后期维护的每个阶段，都要进行定期的质量监控和评估，确保项目活动符合既定的质量标准和要求。

·风险识别与应对：通过质量评审、审计和测试，及时发现潜在的质量风险，制定并实施应对措施。

·持续改进：根据质量监控结果，不断调整和优化项目管理流程，提升项目整体质量。

2. 需求过程审计

在需求分析阶段进行：

·需求文档审查：审查需求文档是否完整、准确、清晰，确保所有需求都已明确并记录。

·需求一致性检查：检查需求之间是否存在冲突或不一致，确保需求的一致性和可行性。

·利益相关者沟通：与项目利益相关者（如用户、业务代表等）进行沟通，验证需求的正确性和完整性。

·需求变更管理：建立需求变更控制流程，确保需求变更得到适当的评估、批准和记录。

3. 设计说明书审计

在概念设计和技术审评阶段进行：

·设计文档审查：审查系统设计说明书、数据库设计、接口设计等文档，确保设计符合需求规格说明。

·技术可行性评估：评估设计的技术可行性，包括技术选型、架构合理性、性能预测等。

·安全性与合规性检查：确保设计符合相关的安全标准和法规要求。

·评审会议：组织专家和设计团队进行评审会议，对设计方案进行充分讨论和评估。

4. 代码评审

在软件开发和集成测试阶段进行：

·代码审查：对编写的代码进行逐行审查，检查代码质量、可读性、可维护性和性能。

·单元测试验证：确保每个模块都经过充分的单元测试，验证其功能正确性和边界条件处理。

·代码走查：通过代码走查活动，发现潜在的问题和错误，如内存泄漏、资源未释放等。

·集成测试反馈：在集成测试阶段，根据测试结果对代码进行必要的调整和优化，确保系统整体性能和质量。

1. 综合管理

·质量报告：在每个阶段结束后，编制质量报告，总结质量目标的实现情况、存在的问题和改进措施。

·持续改进机制：建立持续改进机制，鼓励团队成员提出改进建议，不断优化项目管理流程和质量控制措施。