（规格为A4纸或A3纸折叠）

|  |
| --- |
| 佛山科学技术学院  实 验 报 告  课程名称 IT项目管理  实验项目 项目质量控制  专业班级 22计科3 姓 名 张利荣、徐文杰、周正浩、叶志鹏、黎叶豪 学 号 20220310311、20220310320、20220310317、20220310301、20200390139  指导教师 胡小生 成 绩 日 期 2024.11.1    **一、实验目的**  1．掌握软件项目质量跟踪控制的过程。  **二、实验内容**  1. 复习软件项目质量跟踪控制的方法；  2. 完成SPM项目质量目标的跟踪控制、需求过程审计、设计说明书审计、代码评审等；  3. 选择1个团队课堂上讲述SPM项目的质量目标跟踪控制、需求过程审计、设计说明书审计、代码评审过程。  **三、实验步骤**  **1. 复习软件项目质量跟踪控制的方法**   * **步骤1.1**：回顾项目管理中的质量控制概念，包括质量规划、质量保证和质量控制的区别。 * **步骤1.2**：学习常用的质量跟踪工具和技术，如质量指标、评审、审计、缺陷管理和持续改进等。 * **步骤1.3**：讨论如何制定可量化的质量目标，以及如何收集和分析相关数据。   **2. 确定SPM项目质量目标**   * **步骤2.1**：明确SPM项目的质量目标，包括功能性、性能、安全性和可维护性等指标。  1. **实验结果**   **1. 质量规划（Quality Planning）**   * **定义**：质量规划是确定项目质量目标和要求的过程，同时制定实现这些目标的具体策略和计划。 * **内容**：   + 确定质量标准和指标。   + 识别所需的质量管理活动。   + 制定质量管理计划，包括资源分配、时间安排等。   **2. 质量保证（Quality Assurance, QA）**   * **定义**：质量保证是确保项目和过程按照既定标准和规范进行的活动，重点在于预防问题的发生。 * **内容**：   + 监控和审查过程以确保符合质量标准。   + 实施过程改进，提升团队的工作效率和质量水平。   + 提供培训和支持，帮助团队理解质量要求。   **3. 质量控制（Quality Control, QC）**   * **定义**：质量控制是检测和评估项目结果的过程，以确保交付的产品符合质量标准，重点在于识别和纠正缺陷。 * **内容**：   + 进行产品的测量和评审（如代码审查、测试等）。   + 记录缺陷并跟踪其解决情况。   + 根据检测结果进行纠正措施和调整。   **总结区别**   * **目的**：   + 质量规划关注的是“计划怎么做”。   + 质量保证关注的是“确保过程的合规性”。   + 质量控制关注的是“评估和纠正结果”。 * **活动**：   + 质量规划是事先制定的策略。   + 质量保证是过程中的持续监督和改进。   + 质量控制是事后对产品质量的评估和问题处理。   **1. 项目概述**  **本项目旨在开发一套内控管理信息系统，目标是提升企业内部控制的效率与透明度，降低风险，确保合规性，并提升决策支持能力。为实现上述目标，必须重视项目交付过程中的质量控制，确保系统满足业务需求并具备良好的性能与安全性。**  **2. 项目质量目标**  **项目质量目标具体包括：**   * **功能性：**   + **系统应满足所有业务需求，包括财务管理、风险管理和合规监控等功能模块。**   + **所有功能必须经过业务用户验证，确保其符合实际操作流程。** * **性能：**   + **系统在正常使用情况下，应能在2秒内响应用户请求。**   + **在高峰使用情况下（例如月末结算），系统应支持至少500名用户同时在线，响应时间不超过5秒。** * **安全性：**   + **系统必须实现多层次的安全机制，包括用户身份验证、数据加密和访问控制。**   + **定期进行安全审计和漏洞扫描，确保系统防护措施的有效性。** * **可维护性：**   + **代码应遵循编码规范，便于团队成员理解和修改。**   + **系统架构应支持模块化设计，以便未来的功能扩展和技术升级。**   **3. 质量目标跟踪控制**  **3.1 制定质量标准**   * **质量标准文档：**   + **制定包含功能性、性能和安全性指标的详细质量标准文档，明确各项指标的衡量标准。**   + **各项指标应与项目需求紧密关联，并通过关键绩效指标（KPI）进行量化。** * **验收标准：**   + **明确各个阶段的验收标准，包括开发阶段、测试阶段和上线阶段的具体要求。**   + **验收标准应在项目初期与利益相关者进行确认，并在项目实施过程中定期回顾。**   **3.2 定期评估**   * **进度检查：**   + **每两周召开项目进度评审会，检查已完成任务与计划的符合度，识别项目延误的潜在原因。**   + **会议中记录进展情况、问题及解决方案，确保各项任务的透明度。** * **质量审查：**   + **在每个项目阶段结束后进行质量审查，确认交付物是否符合质量标准，审查内容包括功能测试、性能测试和安全测试的结果。**   + **通过质量审查生成的报告将作为项目阶段完成的依据。**   **3.3 质量审查会议**   * **会议频率：**   + **每月召开一次质量审查会议，重点讨论项目的质量状态、当前存在的风险及应对措施。**   + **邀请项目团队的各个成员参与，确保信息共享和多方意见的汇集。** * **记录与行动项：**   + **会议记录将包括待解决的问题、建议和相应的责任人及截止日期，以确保问题的跟进和解决。**   **4. 需求过程审计**  **4.1 需求收集与分析**   * **文档审查：**   + **定期审核需求文档，确保其完整性和清晰性，特别关注需求的可追溯性，确保每一项需求都有明确的来源和描述。**   + **需求文档需包括功能需求、非功能需求（如性能、可用性等）及相关的业务背景信息。** * **利益相关者反馈：**   + **定期与利益相关者沟通，确认需求的准确性，确保所有关键用户的需求都得到满足。**   + **通过会议或问卷收集反馈意见，及时调整需求文档。**   **4.2 变更管理**   * **变更请求流程：**   + **建立标准化的变更请求流程，确保所有变更都有记录，并经过合理的审批流程。**   + **每项变更请求需包含变更的理由、影响范围及实施建议。** * **影响分析：**   + **对每次变更进行影响分析，评估对项目整体进度、资源及预算的影响，确保变更的合理性与必要性。**   + **将变更分析结果纳入项目风险管理，必要时调整项目计划。**   **5. 设计说明书审计**  **5.1 设计文档评审**   * **设计一致性检查：**   + **审查设计文档与需求文档的一致性，确保设计方案充分满足需求，避免设计遗漏或误解。**   + **设计文档应包括详细的架构设计、数据流图及界面原型等信息，以便于开发和测试。** * **设计方案可行性：**   + **评估设计方案的技术可行性和业务适应性，确保设计方案能在预算内按时交付。**   + **对于关键设计决策，需进行技术评估，确保选择最佳方案。**   **5.2 设计符合性检查**   * **遵循编码标准：**   + **确保设计遵循行业编码标准及最佳实践，避免技术债务的积累，确保代码质量和可维护性。**   + **定期进行编码规范培训，增强团队对编码标准的认知。**   **6. 代码评审**  **6.1 代码审查流程**   * **同伴评审：**   + **每个开发人员在提交代码前，必须经过至少一名同事的审查，确保代码质量、逻辑正确性及符合设计要求。**   + **审查过程中应记录审查结果和改进建议，确保所有意见得到落实。** * **审查标准：**   + **制定代码审查标准，包括代码风格、逻辑复杂度、性能优化等指标，确保代码的一致性和可读性。**   **6.2 自动化测试**   * **测试覆盖率：**   + **确保关键功能的自动化测试覆盖率达到85%以上，包括单元测试、集成测试和端到端测试。**   + **定期审查测试用例，确保其有效性和完整性。** * **持续集成：**   + **引入持续集成工具，自动执行测试，及时发现和解决问题，提升代码交付的质量与速度。**   + **设定自动化测试的失败策略，确保关键功能在发布前经过充分验证。**   **7. 风险管理**  **7.1 风险识别与评估**   * **风险登记册：**   + **建立风险登记册，识别项目中可能遇到的风险，并定期更新风险的严重性及可能性评估。**   + **每个风险应记录其潜在影响、发生概率和应对措施，确保风险信息的透明度。** * **风险应对策略：**   + **针对每项识别的风险，制定相应的应对策略，包括规避、减轻、转移和接受等方案，确保项目进展的稳定性。**   + **定期评审风险应对措施的有效性，根据项目进展调整风险管理策略。**   **7.2 风险监控**   * **定期审查：**   + **每月审查已识别风险，确认应对措施的有效性，并进行必要的调整。**   + **通过风险监控工具，实时跟踪风险状态，确保团队及时响应。**   **8. 质量改进计划**  **8.1 反馈机制**   * **收集反馈：**   + **设立匿名反馈渠道，鼓励团队成员和利益相关者提出意见和建议，确保改进措施的全面性和有效性。**   + **定期分析反馈数据，识别共性问题，制定改进计划。** * **改进措施：**   + **针对反馈进行深入分析，制定相应的改进措施，并将改进结果纳入后续项目计划中。**   + **设定改进目标和时间框架，确保改进措施的落地。**   **8.2 培训与知识共享**   * **定期培训：**   + **每季度组织质量管理和最佳实践培训，提高团队成员的专业素养，确保团队对质量标准的理解和执行。**   + **邀请外部专家进行分享，拓展团队的视野和知识。** * **知识库建立：**   + **构建项目经验和最佳实践的知识库，供团队成员查阅和学习，促进团队内的知识共享。**   + **知识库应包含案例研究、常见问题解决方案和技术文档等，便于团队成员获取信息。**   **步骤1.3**：  **1. 制定可量化的质量目标**  **（1）明确质量指标**   * **功能性**：如需求实现的百分比、用户故事的完成情况。 * **性能**：如响应时间、处理速度、系统吞吐量等。 * **可靠性**：如故障率、平均无故障时间（MTBF）等。 * **可维护性**：如代码复杂度、文档完整性、缺陷修复时间等。   **（2）使用SMART原则**   * **具体（Specific）**：目标要明确，避免模糊不清。 * **可测量（Measurable）**：设定可量化的标准，如“减少缺陷率到5%以下”。 * **可实现（Achievable）**：目标需切合实际，基于团队能力和资源。 * **相关性（Relevant）**：确保目标与项目整体目标一致。 * **时间限制（Time-bound）**：为目标设定明确的截止时间。   **（3）参与相关方**   * 与项目团队、利益相关者和用户沟通，确保目标符合各方期望和需求。   **2. 收集相关数据**  **（1）定义数据来源**   * **需求文档**：验证需求是否得到实现。 * **测试报告**：记录功能测试、性能测试和用户验收测试的结果。 * **缺陷跟踪系统**：追踪和记录软件缺陷的数量和类型。 * **用户反馈**：通过调查、访谈等方式收集用户意见和建议。   **（2）自动化工具**   * 使用自动化测试工具、持续集成工具和项目管理软件收集实时数据，提高数据收集的效率和准确性。   **（3）定期审查**   * 定期召开项目评审会议，汇总质量数据，分析质量目标的达成情况。   **3. 分析相关数据**  **（1）数据整理与可视化**   * 使用图表、仪表盘等方式将数据可视化，帮助团队快速识别趋势和问题。   **（2）统计分析**   * 采用统计方法（如平均值、标准差、趋势分析等）评估数据，识别潜在的质量问题。   **（3）对比分析**   * 将当前数据与历史数据进行对比，分析质量目标的达成情况，并识别改进机会。   **（4）根本原因分析**   * 对于未达标的质量目标，进行根本原因分析，确定导致问题的根源，并制定改进措施。   **4. 持续改进**   * 根据数据分析结果，不断调整和优化质量目标和策略，形成持续改进的循环。   通过这些步骤，团队能够制定出明确的、可量化的质量目标，并通过有效的数据收集和分析，确保项目的质量不断提升。 |

注：1、实验报告的内容: 一、实验目的；二、实验原理；三、实验步骤；四、实验结果；五、讨论分析

（完成指定的思考题和作业题）；六、改进实验建议。

2、各专业可在满足学校对实验教学基本要求的前提下，根据专业特点自行设计实验报告的格式，所设

计的实验报告在使用前需交实践教学管理科备案。