# 软件开发计划

55200105 安姿铜

**修订历史记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2023/10/1 | <1.0> | 初始软件开发计划 | 安姿铜 |
| 2023/10/14 | <1.1> | 增加历史修订记录 | 安姿铜 |
| 2023/11/1 | <2.0> | 细化项目资源配备，添加任务分配 | 安姿铜 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CCB评审** | **检查项描述** | **要求满足情况** | **是否满足情况** | **备注** |
| A001 | 配置项命名规范是否符合要求 | 是 | 是 |  |
| A002 | 配置项文档是否完整 | 是 | 是 |  |
| A003 | 有无修订历史表 | 是 | 是 |  |

目录

[1. 简介 3](#_Toc8941)

[1.1 目的 3](#_Toc11369)

[1.2 范围 3](#_Toc26657)

[1.3 概述 3](#_Toc24355)

[2. 项目概述 3](#_Toc9104)

[2.1 项目的目的、规模和目标 3](#_Toc8732)

[2.2 假设与约束 3](#_Toc10864)

2.3项目的可交付工件..............................................................................................................................4

2.4软件开发计划的演进.........................................................................................................................4

[3. 项目组织 4](#_Toc3902)

3.1组织结构 ..4

3.2外部接口................................................................................................................................................5

3.3角色与职责...........................................................................................................................................5

[4. 管理流程 5](#_Toc2656)

[4.1 项目估计 5](#_Toc29574)

[4.2 项目计划 6](#_Toc25314)

[5. 技术流程计划 8](#_Toc30535)

[5.1 开发案例 8](#_Toc17510)

[5.2 方法、工具和技术 8](#_Toc9071)

[5.3 基础设施计划 9](#_Toc7189)

[5.4 产品验收计划 9](#_Toc30228)

[6. 支持流程计划 10](#_Toc19635)

[6.1 配置管理计划 10](#_Toc18865)

[6.2 评估计划 10](#_Toc1044)

[6.3 文档计划 11](#_Toc25863)

[6.4 质量保证计划 11](#_Toc25925)

[6.5 问题解决计划 11](#_Toc16634)

[6.6 流程改进计划 11](#_Toc1918)

**1. 简介**

**1.1 目的**

项目的目的是开发一个火车票自动售票机模拟软件，旨在提供用户友好的界面和功能，使乘客能够方便地购买火车票、查询票价、选择座位等操作；使管理员能完成后台管理、资源分配和统计分析等功能。通过该软件，用户可以熟悉操作，提高使用真实平台时的购票效率。

**1.2 范围**

（1）功能需求：明确定义火车票自动售票机模拟软件的功能，包括购票、查询票价、选择座位、取消订单等功能。需求应具体明确，包括输入输出的数据格式、各功能的流程和操作逻辑等。

（2）界面设计：设计用户友好的界面，使用户能够直观地进行购票操作。界面应符合用户习惯，布局合理，并提供必要的帮助和提示信息。

（3）性能要求：确定软件的性能要求，如响应时间、并发处理能力、容错性等。保证软件能够在高负载、并发访问情况下稳定可靠地运行。

（4）数据管理：设计数据库结构和关系，确保数据的可靠存储和快速检索。包括用户信息、车次信息、票价信息、座位信息等数据的管理和维护。

（5）安全性和权限管理：确保软件的安全性，包括用户信息的保护、管理员权限的管理等。采用密码加密、访问控制等方法，防止未经授权的访问和恶意操作。

（6）测试计划：制定详细的测试计划，包括单元测试、集成测试、系统测试和用户验收测试等，以确保软件的质量和功能的正确性。

（7）发布和部署：制定发布计划，选择合适的部署环境和方式，确保软件能够顺利上线并投入使用。同时，建立持续的技术支持和维护机制，及时修复漏洞和故障。

**1.3 概述**

在软件开发计划中，包括项目概述、项目组织、管理流程、技术流程计划和支持流程计划等主要内容。

文档的组织方式可以采用分节和子节的方式进行，每个部分都有自己的标题和内容。可以使用类似目录的方式将各个部分列出，并标明其在文档中的页码或链接。

**2. 项目概述**

**2.1 项目的目的、规模和目标**

火车票自动售票机模拟软件项目的目的是开发一个模拟火车票自动售票机的软件系统，以提供给用户一个真实的购票体验。该软件可以模拟用户查询车次信息、选择座位、购买车票、支付等功能。

该项目中将包含核心的火车票自动售票机功能，主要涵盖车次查询、座位选择、票价计算、购票流程和订单管理等关键模块。

项目目标如下：

1. 实现基本的购票流程：确保用户能够进行车次查询、座位选择、票价计算、购票和支付等主要功能，并顺利生成有效的订单。
2. 保证系统的稳定性和安全性：对软件进行充分的测试和调试，确保系统在正常情况下能够稳定运行，并保护用户敏感信息的安全。
3. 用户友好的界面和操作体验：设计直观友好的用户界面，提供简化的操作流程，使用户能够轻松理解和使用系统的各项功能。

**2.2 假设与约束**

假设：

假设存在可靠的数据源：假设可以从一个可靠的数据源（如铁路局提供的API）获取实时的车次信息、票价和座位数据。

约束：

1. 预算约束：项目的开发预算有限，需要在预算范围内完成开发工作，并优化资源利用以确保项目的可行性。
2. 人员约束：项目所需的开发人员和测试人员人手有限，应为团队中的六人安排合理的工作量，并确保项目进度和质量。
3. 时间约束：项目计划要求在两个月内完成，因此需要制定详细的时间表，并合理安排各项工作任务和里程碑，以确保按时交付。
4. 设备约束：项目所需的开发和测试设备为小组成员个人电脑。
5. 技术约束：项目需要使用可靠的开发框架和工具，并遵循最佳实践和行业标准，确保软件的稳定性、安全性和可维护性。

**2.3 项目的可交付工件**

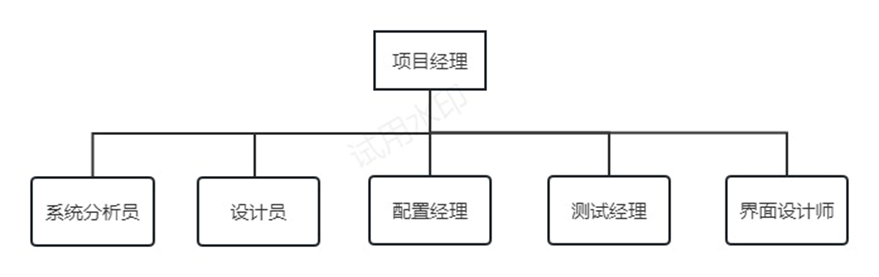
|  |  |  |  |
| --- | --- | --- | --- |
| **工件** | **目标交付日期** | **交付角色** | **交付人员** |
| 软件需求规约文档 | 2023.10.11 | 系统分析员 | 赵欣宇 |
| 软件开发计划 | 2023.10.11 | 项目经理 | 安姿铜 |
| 用户界面设计 | 2023.10.11 | 界面设计师 | 罗香雨 |
| 配置管理计划 | 2023.10.25 | 配置经理 | 狄锦文 |
| 软件测试计划 | 2023.10.25 | 测试经理 | 孟庆涛 |
| 软件设计文档 | 2023.11.08 | 设计员 | 郑佳策 |
| 详细软件开发计划 | 2023.11.08 | 项目经理 | 安姿铜 |
| 测试设计文档 | 2023.11.22 | 测试经理 | 孟庆涛 |
| 测试报告 | 2023.12.06 | 测试经理 | 孟庆涛 |
| 软件配置库 | 2023.10.06 | 配置经理 | 狄锦文 |
| 用户手册 | 2023.12.06 | 界面设计师 | 罗香雨 |
| 程序代码 | 2023.12.13 | 程序员 | 安姿铜、郑佳策 |

**2.4 软件开发计划的演进**

|  |  |  |
| --- | --- | --- |
| **版本号** | **提议版本** | **符合标准** |
| **1.0** | 初始计划版本 | 项目目标、范围和计划已经明确，并与相关利益相关方达成共识 |
| **1.1** | 调整时间安排、资源分配 | 时间安排和资源分配符合可行性和项目实际情况，未超出预算 |
| **1.2** | 修改界面设计和用户体验 | 界面设计经过用户测试和反馈优化，并满足用户友好和易用性的要求 |
| **1.3** | 进行系统测试和质量保证 | 软件经过充分的测试，包括功能测试、性能测试和安全测试，并符合质量标准 |
| **1.4** | 编写系统维护文档和用户手册 | 维护文档和用户手册编写完整、准确，便于后续维护和使用 |
| **1.5** | 进行上线前演练和生产环境上线 | 上线前演练顺利进行，生产环境上线符合稳定性和可用性的要求 |
| **1.6** | 发布维护计划和持续改进 | 发布维护计划，确保后续的维护工作和持续改进，并适应变化的需求 |

**3. 项目组织**

**3.1 组织结构**

****

|  |  |  |  |
| --- | --- | --- | --- |
| 管理人员 | 项目经理 | 安姿铜 | 管理整个项目计划和资源分配 |
|  | 分析员 | 赵欣宇 | 分析用户需求、撰写需求文档并与开发团队沟通协调 |
|  | 测试经理 | 孟庆涛 | 管理测试部门的工作并监督测试流程 |
|  | 配置经理 | 狄锦文 | 管理软件配置、部署及环境问题，确保系统可靠稳定运行 |
| 团队成员 | 设计师 | 郑佳策 | 主要工作内容有系统需求分析，系统架构设计，系统详细设计，技术选型，制作设计文档 |
|  | 界面设计师 | 罗香雨 | 负责软件界面设计、交互设计和视觉效果 |
|  | 开发工程师 | 安姿铜、郑佳策 | 负责系统开发、编码实现和技术支持 |
|  | 数据库管理员 | 安姿铜、郑佳策 | 负责数据库的设计、优化和维护 |

**3.2 外部接口**

|  |  |  |  |
| --- | --- | --- | --- |
| **外部组织名称** | **内部联系人** | **外部联系人** | **联系方式** |
| 客户 | 项目经理 | 客户经理 | 邮件、电话、面谈 |

**3.3 角色与职责**

|  |  |  |
| --- | --- | --- |
| 项目经理 | 安姿铜 | 1. 负责整个项目的规划和管理，确保项目按时、按质地完成。 2. 协调和分配团队成员的任务和资源。 3. 监督项目进展，解决问题和风险管理。 4. 与客户和外部组织进行沟通和协调。 |
| 分析员 | 赵欣宇 | 1. 分析用户需求，撰写需求文档和功能规格说明书。 2. 协调用户和开发团队之间的沟通，确保需求的准确性和一致性。 3. 参与系统设计和架构讨论，并提供技术指导。 |
| 设计师 | 郑佳策 | 1. 负责软件界面设计、交互设计和视觉效果。 2. 根据需求和用户反馈，制定用户友好的界面设计方案。 3. 与开发团队密切合作，确保设计能够顺利实现。 |
| 测试经理 | 孟庆涛 | 1. 制定测试计划和策略，确保系统质量和稳定性。 2. 设计测试用例和执行测试，检测和报告缺陷。 3. 管理测试资源和进度，与开发团队合作解决问题。 |
| 配置经理 | 狄锦文 | 1. 负责软件配置、部署和环境管理。 2. 确保软件系统的正确配置和可靠运行。 3. 处理配置变更和版本控制，管理软件发布流程。 |
| 界面设计师 | 罗香雨 | 1. 负责软件界面设计、交互设计和用户体验。 2. 根据用户需求和产品定位，设计易用和具有吸引力的界面。 3. 与开发团队密切合作，确保设计理念的实现。 |

**4. 管理流程**

**4.1 项目估计**

项目需要3名开发人员，2位测试人员，均由小组内成员负责；硬件设备和软件均使用小组成员现有个人电脑，无额外费用；无其他费用。

参考过去类似项目的数据，包括成本、进度和资源使用情况，进度估计如下所示，

|  |  |
| --- | --- |
| 需求收集和分析阶段 | 一周 |
| 设计和开发阶段 | 六周 |
| 测试阶段 | 两周 |

重新估计时间点和情况如下所示：

1. 项目启动和计划阶段结束后：在这两个阶段，可以根据整体项目规划的反馈和需求分析阶段的完善，进行成本和进度的重新评估。
2. 项目中的里程碑节点：在完成系统设计、编码阶段的结束、测试阶段的结束等关键项目节点处，进行重新估计。
3. 变更控制：当需求发生变更或范围调整时，需要根据变更内容对成本和进度进行重新评估。

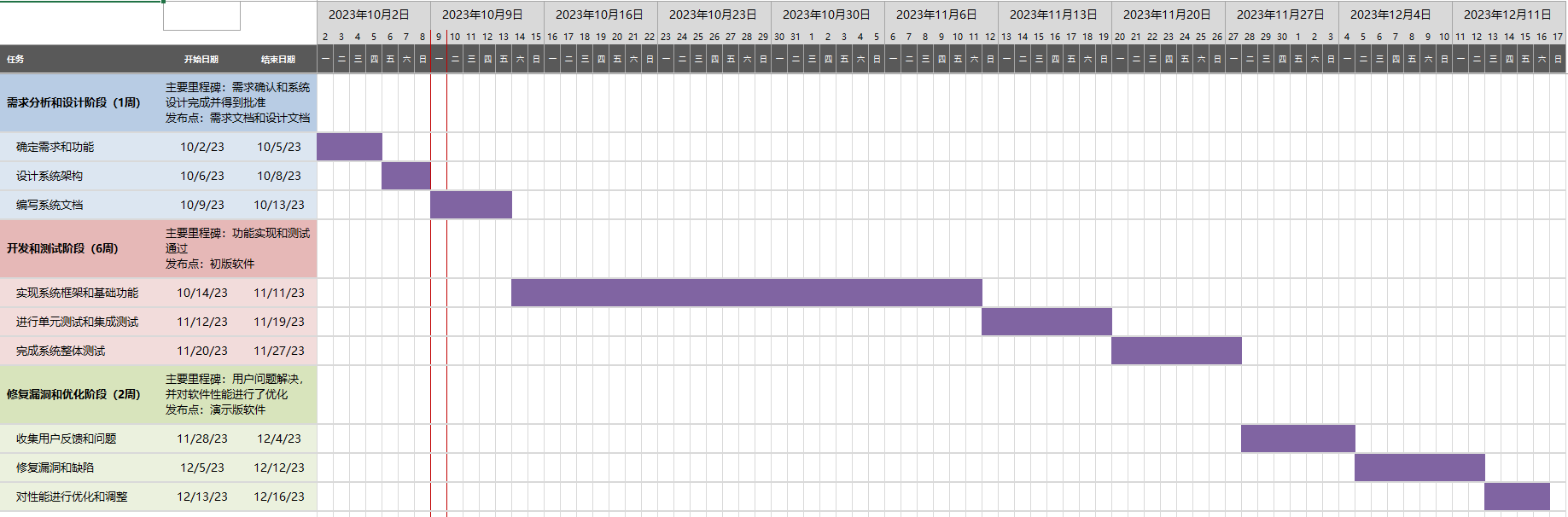
**4.2 项目计划**

**4.2.1 阶段计划**

阶段计划如下所示：



甘特图如下所示：



**4.2.2 迭代目标**

|  |  |
| --- | --- |
| 迭代1 | 1. 设计界面原型：创建火车票自动售票机的用户界面原型，包括主页面、购票页面和支付页面等。 2. 实现基本功能：实现用户登录、车次查询、座位选择和订单生成等基本功能。 3. 进行单元测试：对已实现的功能进行单元测试，确保其正确运作。 |
| 迭代2 | 1. 完善购票流程：优化购票流程，包括添加乘客信息、选择座位、填写联系人信息等。 2. 增加支付功能：集成第三方支付接口，实现在线支付功能。 3. 添加订单管理：提供订单列表和订单详情页面，用户可以查看和管理自己的订单。 4. 进行系统整体测试：对已实现的功能进行系统整体测试，确保各个模块之间的协调运作 |
| 迭代3 | 1. 引入票价策略：根据不同的日期、车次和座位类型制定票价策略，并在购票过程中实时计算票价。 2. 添加退票功能：为用户提供退票服务，包括退票申请、退款计算和退票流程。 3. 集成身份验证：集成身份验证系统，确保用户的身份信息安全。 4. 进行性能优化：对系统进行性能优化，提高响应速度和用户体验。 |

**4.2.3 发布版**

|  |  |  |
| --- | --- | --- |
| 发布版 1.0 | 演示版 | * 此版本是火车票自动售票机模拟软件的初始版本，旨在展示基本功能和界面设计。 * 功能包括用户登录、车次查询、座位选择、订单生成等核心功能。 * 用户可以使用演示账号登录并体验软件的基本功能，但无法进行真实交易。 |
| 发布版 2.0 | Beta 版 | * 此版本是火车票自动售票机模拟软件的改进版本，引入了更多功能和改进。 * 增加了购票流程的细化，例如添加乘客信息、选择座位、填写联系人信息等。 * 添加了订单管理功能，用户可以查看和管理自己的订单。 |
| 发布版 3.0 | 正式版 | * 此版本是火车票自动售票机模拟软件的正式发布版，具备完整的功能和稳定性。 * 引入了票价策略，根据不同的日期、车次和座位类型制定票价，并在购票过程中实时计算票价。 * 进行了性能优化，提高了系统的响应速度和用户体验。 |

**4.2.4 项目时间表**

|  |  |  |
| --- | --- | --- |
| **迭代/阶段** | **目标** | **目标日期** |
| 迭代1 | 设计界面原型 | 2023-10-11 |
| 实现基本功能 | 2023-11-11 |
| 单元测试 | 2023-11-19 |
| 迭代2 | 完善购票流程 | 2023-11-21 |
| 添加订单管理 | 2023-11-21 |
| 系统整体测试 | 2023-11-23 |
| 迭代3 | 性能优化 | 2023-11-27 |
| 发布版1.0 | 演示版 | 2023-11-27 |
| 发布版2.0 | Beta 版 | 2023-12-12 |
| 发布版3.0 | 正式版 | 2023-12-16 |

**4.2.5 项目资源分配**

项目资源如下所示，

|  |  |  |  |
| --- | --- | --- | --- |
| **人员类型** | **数量** | **担任成员** | **所需技能或经验** |
| 产品经理 | 1人 | 安姿铜 | 具备火车票销售业务和用户需求分析的经验；能够制定产品规划和功能需求；协调团队工作。 |
| 设计员 | 1人 | 郑佳策 | 熟悉用户界面设计原则和用户体验设计；能够创建界面原型和设计可用性测试。 |
| 分析员 | 1人 | 赵欣宇 | 有数据库设计和数据分析经验；能够进行系统需求分析和数据建模。 |
| 配置经理 | 1人 | 狄锦文 | 熟悉软件配置管理流程；能够管理开发、测试和发布环境的配置；具备版本控制和持续集成经验。 |
| 测试经理 | 1人 | 孟庆涛 | 有软件测试经验，熟悉测试策略和方法；能够制定测试计划和执行测试用例；善于进行缺陷跟踪和性能测试。 |
| 界面设计师 | 1人 | 罗香雨 | 精通用户界面设计工具和技术；熟悉网页设计和响应式设计；能够根据产品需求设计界面元素和布局。 |

任务详细分配如下所示，

|  |  |  |  |
| --- | --- | --- | --- |
| **所属阶段** | **任务** | | **执行者** |
| 迭代1 | 业务建模 | | 罗香雨 |
| 制定项目计划和时间表 | | 安姿铜 |
| 需求分析 | | 赵欣宇 |
| 概要设计 | | 郑佳策 |
| 设计界面原型 | | 罗香雨 |
| 实现基本功能 | 定义UserMapper、TrainticketMapper、TrainseatMapper、OrdersMapper等模型类 | 安姿铜、郑佳策 |
| 实现类OrderController、OrderServiceIpml、UserController、UserServiceImpl | 安姿铜、郑佳策 |
| 实现边界类ConfirmOrder、DetailOrder、ChangeTicket、ManageUser、SelectUser、Login、Register | 安姿铜、郑佳策 |
| 单元测试 | | 孟庆涛 |
| 迭代2 | 完善需求分析 | | 赵欣宇 |
| 详细设计 | | 郑佳策 |
| 完善购票流程 | 对ConfirmOrder、ChangeTicket进行修改，并修改OrderController、OrderServiceIpml中相应的方法 | 安姿铜 |
| 添加订单管理 | 添加边界类Order并在OrderController、OrderServiceIpml中添加相应的方法 | 郑佳策 |
| 制定测试计划 | | 孟庆涛 |
| 制定配置计划 | | 狄锦文 |
| 系统整体测试 | | 孟庆涛 |
| 迭代3 | 增加附加功能 | 在UserController、UserServiceImpl 、OrderController、OrderServiceIpml中添加相应的方法 | 安姿铜、郑佳策 |
| 测试并修复缺陷 | | 孟庆涛 |
| 进行安全措施 | | 狄锦文 |
| 增加用户界面友好型 | | 罗香雨 |
| 发布并收集用户反馈 | | 安姿铜 |
| 移交阶段 | 进行系统测试 | | 孟庆涛 |
| 修复发现问题 | | 郑佳策 |
| 进行最后部署和配置 | | 狄锦文 |
| 编写用户手册 | | 罗香雨 |
| 组内评审 | | 安姿铜 |
| 最终评审并发布 | | 安姿铜 |

**5. 技术流程计划**

**5.1 开发案例**

开发一个能够模拟火车票自动售票机的软件，以方便用户查询、购买或退票，支持用户登录和注册、查询余票信息、购买车票、退票等功能。

**5.2 方法、工具和技术**

(1)方法：本项目将采用敏捷开发方法，包括迭代开发

(2)工具：

* 项目管理工具：MicroSoft Project
* 建模工具：Enterprise Architect
* 配置管理工具：Git
* 缺陷管理工具：BugZilla
* UI设计工具：MasterGo

(3)技术：

* 后端：SpringBoot
* 前端界面：Vue2
* 数据库：MySQL

**5.3 基础设施计划**

（1)项目目标和范围：确定开发一个火车票自动售票机模拟软件的目标和范围，包括主要功能和特性。

（2)人员配备：确定项目团队成员及其角色和职责；分配任务和工作量，确保适当的资源分配。

（3）开发方法：选择敏捷开发，制定开发流程和时间表，确定每个阶段的交付物和里程碑。

（4）技术基础设施：确定开发所需的硬件和软件环境，包括开发工具、版本控制系统、测试环境等。配置和测试开发环境，确保各项工具和系统正常运行。

（5）需求分析和设计：进行需求调研和用户分析，明确用户需求和期望。编写需求文档和用例文档，描述系统功能和使用场景，进行系统设计，包括界面设计、数据库设计和架构设计。

（6）开发和测试：根据需求和设计，进行软件编码和单元测试；进行集成测试，确保各个模块的正常协作和功能完整性；进行系统测试和性能测试，验证系统的稳定性和性能指标。

（7）文档编写和培训：编写用户手册和操作指南，帮助用户了解系统的功能和使用方法。编写技术文档和维护手册，方便后续开发和维护工作。进行培训，培训相关人员使用和维护系统。

（8）上线和部署：准备生产环境，包括服务器和网络设备的配置与部署。将软件部署到生产环境，进行系统的验证和性能优化。上线系统，投入使用，并进行必要的监控和维护。

**5.4 产品验收计划**

1. 验收标准：

* 确定软件的功能和性能验收标准，包括主要功能的正确性和稳定性。
* 确定性能指标，如响应时间、并发处理能力等。

1. 测试和验证：

* 进行回归测试，确保已修复的缺陷不再出现。
* 进行用户验收测试，邀请用户或代表用户进行系统测试，确认满足用户需求。

1. 缺陷修复：

* 根据验收测试结果，修复发现的缺陷和问题。
* 进行二次测试，验证修复后的系统功能和性能。

1. 文档和培训：

* 确认用户手册和操作指南是否完善且易于理解。
* 开展培训活动，确保相关人员理解和熟悉系统的使用方法。

1. 验收评估：

* 进行验收评估，根据验收标准对系统进行评价。
* 收集用户反馈和建议，以进一步改进和优化系统。

1. 产品正式交付：

* 确认系统达到了验收标准，准备正式交付。
* 完成交付文档和相关资料，包括软件安装指南、培训材料等。

**6. 支持流程计划**

**6.1 配置管理计划**

* 1. 配置项版本控制
* 将项目源代码、文档和其他相关配置项放入配置管理工具中进行版本控制。
* 使用版本标签、分支等功能来管理和控制配置项的不同版本。
  1. 变更管理和控制
* 建立变更管理流程，确保每一个配置项的变更都经过审批和记录。
* 设置变更管理委员会或者角色，负责评审和批准变更请求。
* 配置管理工具中的变更管理功能可以用来记录和跟踪变更请求的状态、审批信息和实施情况。
  1. 配置发布策略
* 制定配置发布策略，包括开发环境到测试环境的切换、测试环境到生产环境的切换等。
* 确定发布日期和时间，并且建立发布计划，统一协调配置的发布工作。
  1. 配置审计和验证
* 定期对软件项目的配置进行审计，确保配置项的正确性和一致性。
* 进行配置验证，验证系统在不同配置下的正确性和可用性。
  1. 配置管理文档
* 编写配置管理计划，明确配置管理的目标、任务和责任。
* 编写配置管理手册，包括配置项标识规范、变更管理流程、发布策略等。

**6.2 评估计划**

1. 目标和范围确认

* 明确软件开发的目标和范围。
* 评估软件的功能、性能和安全性等方面。

1. 流程评估

* 分析软件开发流程中可能出现的风险和问题。
* 评估各阶段的交付物和里程碑，确保按时交付。

1. 技术评估

* 评估开发所需的技术和工具环境是否满足需求。
* 确认软件架构和数据库设计是否符合规范和最佳实践。

1. 团队评估

* 分析团队技能和经验，确定是否需要进行培训和知识转移。
* 分析团队沟通协作情况，确保协作顺畅。

1. 风险评估

* 识别软件开发过程中可能遇到的风险和问题。
* 评估风险的严重程度和影响范围，制定风险应对计划。

1. 成本评估

* 评估软件开发的成本和预算，确保开发成本可控。
* 评估软件维护和升级的成本，制定合理的维护计划。

1. 质量评估

* 评估软件的质量，包括功能、性能、安全和易用性等方面。
* 制定测试计划，确保软件质量符合标准和用户需求。

1. 上线评估

* 评估软件上线后的稳定性和性能。
* 收集用户反馈和建议，确定是否需要升级和改进软件。

**6.3 文档计划**

1. 项目概述
2. 需求文档
3. 架构设计文档
4. 数据库设计文档
5. 接口文档
6. 用户手册
7. 开发文档
8. 测试文档
9. 维护文档

**6.4 质量保证计划**

（1）需求验证：与用户和相关利益相关方(整个团队模拟)确认需求。

（2）设计评审：进行设计评审，确保设计满足功能需求和系统架构。

（3）编码规范：制定编码规范，确保代码的可读性、可维护性和一致性。

（4）测试策略：制定全面的测试策略，包括单元测试、集成测试、系统测试和性能测试。

（5）缺陷管理：建立缺陷管理流程，追踪和解决缺陷，确保软件质量达到要求。

**6.5 问题解决计划**

（1）问题记录和归档

（2）问题筛选和优先级

（3）问题分析和解决方案

（4）问题跟进和反馈

（5）问题汇总和分析

**6.6 流程改进计划**

（1）确定目标和范围

（2）评估现有流程和问题

（3）制定改进方案和实施计划

（4）实施改进和监控效果

（5）持续改进