配置管理计划

**目 录**

[配置管理计划 1](#_Toc26772)

[一．引言 2](#_Toc17180)

[二．产品目录 4](#_Toc14745)

[三．配置管理的主要目标 6](#_Toc1108)

[四．配置管理活动负责人的职责 7](#_Toc6209)

[五．实现流程 7](#_Toc17417)

[六．配置标识 8](#_Toc32308)

[七．基线 8](#_Toc31390)

1. **引言**

1.配置计划的目的：

* 版本控制： 维护应用程序的不同版本，以确保能够跟踪和恢复旧版本，同时允许新功能的开发。
* 质量控制： 确保在应用程序的不同部分中维护高质量和稳定性，以降低错误和故障的风险。
* 协同开发： 促进多个团队成员协同开发，确保他们能够合理共享代码和资源。
* 可维护性： 为了将来的维护和更新，确保代码和文档的可维护性。
* 追踪变更： 记录和跟踪代码和文档的变更，以便了解什么、为什么和如何变更。
* 备份和恢复： 提供灾难恢复机制，以便在数据丢失或损坏时能够快速恢复。
* 合规性： 遵守法规和合同要求，以确保项目的合规性。

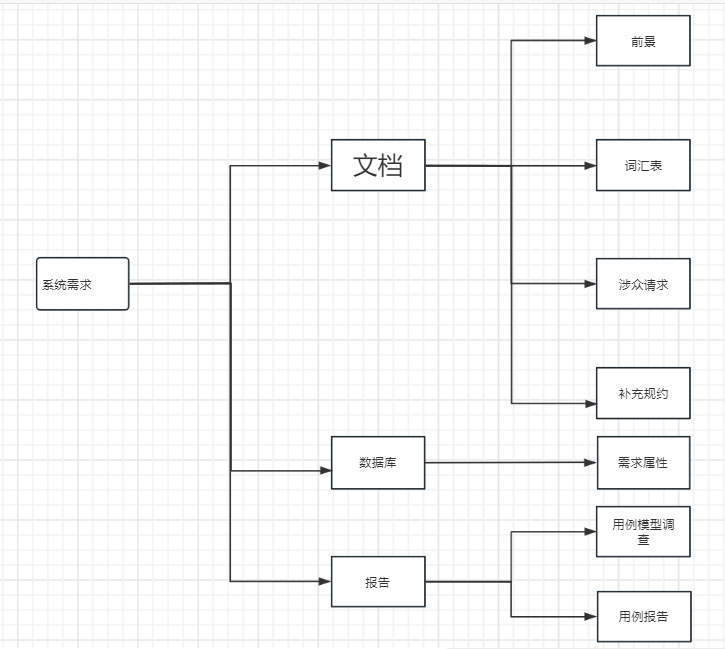
2.配置计划的范围：

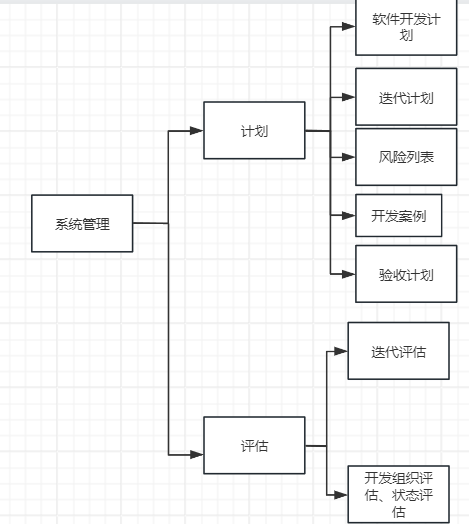
* 源代码管理： 管理应用程序的源代码，包括所有的代码文件和依赖项。
* 文档管理： 管理项目文档，包括用户手册、技术文档、设计文档等。
* 配置文件： 确保应用程序的配置文件被适当管理，以确保不同环境的配置一致性。
* 测试用例和数据： 管理测试用例、测试数据和自动化测试脚本的版本，以支持测试流程。
* 基线版本： 创建和管理基线版本，以便能够回溯到已确认的版本。
* 第三方库和依赖： 管理项目使用的第三方库和依赖，以确保版本一致性。
* 问题和变更跟踪： 跟踪和记录问题报告和变更请求，以确保问题得到适当解决。
* 合规性文档： 管理与法规和合同要求相关的文档，以确保合规性。
* 备份和灾难恢复： 确保配置管理计划支持数据备份和恢复操作。

3.版本号

|  |  |  |
| --- | --- | --- |
| **发布版1.0** | **演示版** | * 此版本是ATM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、取款、存款、转账、余额查询等核心功能。 * 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ATM模拟软件的改进版本，引入了更多功能和改进。 * 具有友好的用户界面 * 增加了用户打印凭证、查看流水、修改密码等功能。 |
| **发布版3.0** | 正式版 | * 此版本是ATM模拟软件的正式发布版，具备完整的功能和稳定性。 * 进行了性能优化，提高了系统的响应速度和用户体验。 |

1. **产品目录**





1. **配置管理的主要目标**

1.主要目标：

* 提高软件质量： 通过有效的配置管理，确保应用程序的不同部分具有高质量、稳定性和可靠性，以减少错误和故障的出现。
* 维护版本控制： 确保能够管理和维护应用程序的不同版本，以便跟踪和控制新功能、错误修复和改进的开发。

1. 主要目的：

* 支持合规性： 确保项目符合相关法规、标准和合同要求。这可能涉及到跟踪和记录合规性相关的信息，以便在审计时提供必要的证据。
* 确保可追溯性： 跟踪和记录配置项的变更，包括什么、为什么和如何发生的变更。这有助于审计、问题解决和历史追溯。

**四．配置管理活动负责人的职责**

配置工作的活动负责人由配置经理完成，确保配置工作正确的执行以满足项目的质量、版本控制、合规性和可追溯性需求，其职责如下：

* 配置项标识： 管理和维护配置项的唯一标识，确保每个配置项都具有正确的标识，以便在整个项目期间进行跟踪。
* 版本控制： 确保不同版本的配置项得到正确的版本控制，包括创建版本、标签和分支。
* 问题跟踪： 管理和跟踪问题报告、变更请求和审计信息，以确保问题的解决和审计要求的满足。
* 合规性和审计： 确保项目合规性文档和配置项得到正确的管理和记录，以满足法规和合同要求。
* 配置项审查： 审查配置项的状态，以确保它们符合项目的质量标准和标准规定。

**五．实现流程**

（1）使用Git或其他版本控制工具来管理应用程序的源代码，以确保版本控制。

（2）创建版本标签来标识每个重要的版本，以便快速回溯和发布管理。

（3）建立代码审查流程，以确保代码质量和合规性。

（4）管理文档的版本，确保技术文档和用户手册的正确性。

（5）跟踪和记录错误报告、变更请求和审计信息，以确保问题的跟踪和问题解决。

（6）建立备份和恢复策略，以确保数据的完整性和可用性。

（7）迭代流程：

|  |  |
| --- | --- |
| **迭代版本** | 目标 |
| **迭代1** | 获取项目需求分析，制定项目计划表，完成项目概要设计，完成用户界面原型设计以及实现，让项目有一个初步的轮廓 |
| **迭代2** | 进行详细设计，实现部分核心功能，如用户登录、管理员登录 |
| **迭代3** | 提交最终需求分析，完成所有详细设计，实现ATM模拟软件的左右核心功能，如：取款、存款、查看余额、查看流水 |
| **迭代4** | 提交最终详细设计、进行单元测试以及集成测试，发现缺陷并修复，发布演示版本、收集反馈以便于在下一次迭代中改进 |
| **迭代5** | 添加附加功能，如：打印凭条等。进行安全措施，提高程序的安全性，如用户密码加密等等，增强用户界面友好性，发布Beta版本、收集反馈以便于在下一次迭代中改进 |
| **迭代6** | 系统测试，修复发现问题并进行性能优化，进行最后部署和配置。发布正式版，结束项目 |

**六．配置标识**

（1）版本号标识

* 标签标识：唯一标识特定的配置项。
* 序列号标识： 为每个配置项分配一个唯一的数字序列号，类似于问题追踪系统中的问题编号，这可以用于跟踪和管理不同配置项。
* 文件路径标识
* 日期时间戳标识： 使用日期和时间戳来标识配置项

1. 管理和维护配置标识的原则：

* 唯一性： 确保每个配置项标识都是唯一的，不会与其他配置项标识发生冲突。如果使用手动分配标识的方法，确保项目团队知道如何避免重复的标识。
* 标识命名规则： 定义一致的标识命名规则，以确保标识的清晰性和易读性。这可以包括前缀、格式和命名约定。
* 标识文档： 创建一个标识文档，其中包含每个配置项的标识和相应的描述。这有助于团队成员理解每个标识的含义。

**七．基线**

在项目中，项目经理和配置管理经理会授权基线的建立。授权者应该审查和批准建立基线的请求，以确保建立基线的时机和目的是符合项目需求和目标的。

1. 基线的作用

* 版本控制： 基线可以用于标记特定版本的应用程序，以确保在将来能够快速回滚到已发布版本。
* 质量控制： 基线可以用来记录通过了一系列测试和质量控制检查的版本。这有助于确保每个发布的版本都满足质量标准。
* 配置管理： 基线有助于捕获特定时间点的配置项状态，包括源代码、文档和配置文件。这使得能够追踪每个基线中的配置项的版本和状态。
* 可追溯性： 基线记录了每个版本的状态，使得可以追溯到以前的版本，了解问题、需求和功能的历史。

（2）基线授权流程

* 请求建立基线： 项目团队成员或相关方向项目管理或配置管理团队提交建立基线的请求。这个请求应包括基线建立的理由和目的，以及所需的配置项和版本。
* 审查和评估： 配置管理团队或项目管理团队对建立基线的请求进行审查和评估。他们将考虑建立基线的时机、目的、影响和合规性。
* 审批流程： 如果审查结果是积极的，请求建立基线将进入审批流程。审批流程通常涉及到以下步骤：

a. 技术审批： 技术团队审查和评估基线请求，以确保建立基线的技术可行 性和合理性。

b. 质量审批： 质量团队审查基线请求，以确保基线满足质量标准。

c. 项目经理审批： 项目经理审查并批准基线请求，以确保建立基线符合项 目计划和目标。

d. 合规性审批： 如果有适用的法规或合同要求，合规性团队审查并批准基 线请求，以确保符合合规性要求。

* 建立基线： 一旦获得所有必要的审批，配置管理团队或相关的团队成员将建立基线，捕获配置项的特定状态和版本。
* 文档和通知： 在建立基线后，相应的文档应该更新，包括基线文档和配置项状态。项目团队和相关方应该及时通知关于基线的建立。
* 授权记录： 所有授权和批准的记录应该妥善保存，以备将来的审计和追踪。
* 变更控制： 一旦基线建立完成，任何对基线进行的变更都应经过相应的变更管理流程和授权。不应随意更改基线。
* 审计和监督： 建立基线后，应定期进行审计和监督，以确保基线的一致性和可追溯性。

（3）基线建立时机：

* 生命周期目标里程碑（先启阶段）
* 生命周期构架里程碑（精化阶段）
* 初始操作性能里程碑（构建阶段）
* 产品发布里程碑（产品化阶段）