**<项目名称>**

**配置管理计划**

**版本 <1.0>**

**修订版历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 14/9/2024 | 1.0 | 配置管理计划的创建 | 李福硕（配置经理） |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

### [1. 简介](#_1.                  简介)

[1.1 目的](#_1.1               目的)

[1.2 范围](#_1.2               范围)

[1.3 定义、首字母缩写词和缩略语](#_1.3定义、首字母缩写词和缩略语)

[1.4 参考资料](#_1.4参考资料)

[1.5 概述](#_1.5               概述)

[2. 软件配置管理](#_2.软件配置管理)

[2.1 组织、职责和接口](#_2.1组织、职责和接口)

[2.2 工具、环境和基础设施](#_2.2工具、环境和基础设施)

[3. 配置管理活动](#_3.                  配置管理活动)

[3.1 配置标识](#_3.1               配置标识)

[3.1.1 标识方法](#_3.1.1          标识方法)

[3.1.2 项目基线](#_3.1.2          项目基线)

[3.2 配置和变更控制](#_3.2               配置和变更控制)

[3.2.1 变更请求的处理和审批](#_3.2.1          变更请求的处理和审批)

[3.2.2 变更控制委员会 (CCB)](#_3.2.2          变更控制委员会 (CCB))

[3.3 配置状态统计](#_3.3               配置状态统计)

[3.3.1 项目介质存储和发布进程](#_3.3               配置状态统计)

[3.3.2 报告和审核](#_3.3.2          报告和审核)

[4. 里程碑](#_4.                  里程碑)

[5. 培训和资源](#_5.                  培训和资源)

[6. 分包商和厂商软件控制](#_6.                  分包商和厂商软件控制)

**配置管理计划**

# 1.简介

## 1.1目的

## 配置管理计划的目的在于确保在项目或系统开发过程中，配置与变更请求管理控制对项目工件的变更，并且维护项目工件的完整性，实现“同时更新”，“有限通知”，“多个版本”以确保产品的正确性、一致性和可追溯性。

## 1.2范围

本配置管理计划负责在开发过程中的控制、状态统计、审计和审查、生产、过程管理等活动。

模型包括流，配置标识，基线，产品目录结构等。

此配置管理计划还影响其他文档和工作产品，如软件测试设计、配置库、用户手册等，这些文档需遵循配置管理计划进行更新和管理，以确保其与配置项的版本和状态保持一致。

## 1.3定义、首字母缩写词和缩略语

CCB：变更控制委员会。

CM：配置管理。

## 1.4参考资料

课件PPT《3.2 配置管理——配置管理计划》

任务0《工具选择说明》

## 1.5概述

文档的结构按以下方式组织：

**2**：软件配置管理，描述了执行配置管理（CM）活动的组织结构、职责和相关接口，以及实现CM功能所需的工具、环境和基础设施。

**3：**配置管理活动，详细介绍了配置标识、配置和变更控制、配置状态统计等方面的具体活动和流程。

4：里程碑，确定与项目或产品 CM 工作相关的内部里程碑和客户里程碑。

5：培训与资源，实施指定的 CM 活动时所需的软件工具、人员和培训。

6：分包商和厂商软件控制，并入在项目环境外部开发的软件。

# 2.软件配置管理

## 2.1组织、职责和接口

|  |  |  |
| --- | --- | --- |
| 角色 | 职责 | 接口 |
| 变更控制委员会（李福硕，吴卓宇，魏汝鑫） | 对版本变更的审批与监督，确保版本变更控制的正确和一致。 | 对任意成员提交的新版本进行审核，确保符合前后一致性，并与其进行沟通版本的控制问题。 |
| 配置经理（李福硕） | 为每个工作人员项目的每个开发人员创建一个工作区环境（根据不同的情况制定开发工作区或集成工作区），定期建立基线以确保各开发人员的工作保持同步。在项目过程中，应该在每次迭代结束点（次要里程碑），以及与生命周期各阶段结束点相关联的主要里程碑处定期建立基线。 | 项目经理：需要详细软件开发计划，并定期向项目经理汇报版本控制情况及进度。  测试经理：将配置管理计划交付给测试经理。  开发人员：为他们开发工作区，基线将合并自从上次建立基线以来开发人员已经交付的工作。 |
| 开发人员 | 参与项目的开发人员将基线所代表的各版本的目录和文件填入各自的工作区。变更一旦并入基线，开发人员就采用新的基线，以与项目中的变更保持同步。 | 变更控制委员会：开发人员可以根据项目所采用的标准在其中实施并测试代码，并提交版本。 |

## 2.2工具、环境和基础设施

管理工具：git

产品数据量的预期大小:300个文件以内，200MB以内

产品团队的分配：6台笔记本电脑，每个人一台

服务器和客户机的实际位置：其中一台可作服务器。

产品结构目录：

|  |  |  |  |
| --- | --- | --- | --- |
| **系统需求** | **模型** | **用例模型** | |
| **用户界面原型** | |
| **数据库** | **需求属性** | |
| **文档** | **用例规约** | |
| **补充规约** | |
| **系统设计与实施** | **模型** | **分析模型** | **用例实现** |
| **文档** | **软件开发计划** | |
| **各成员的初始项目计划** | |
| **系统测试** | **计划** | **测试计划** | |
| **测试用例** | **测试过程用户界面测试、性能评测、访问控制测试、安装测试** | |
| **测试结果** | **测试报告** | |
| **工具** | **开发环境工具** | **VScode，IDEA** | |
| **配置管理工具** | **Git** | |
| **项目管理工具** | **飞书** | |
| **可视化建模工具** | **亿图** | |
| **UI设计工具** | **Pixso** | |
| **缺陷管理工具** | **JIRA** | |
| **配置版本** | **各项工具需要配置的说明** |  | |

# 3.配置管理活动

## 3.1 配置标识

### **3.1.1标识方法**

|  |  |
| --- | --- |
| **PLN** | **软件开发计划** |
| **REQ** | **需求规约** |
| **USC** | **用例模型** |
| **SRC** | **源代码文件** |
| **TST** | **测试结果** |
| **DOC** | **文档** |
| **UIP** | **用户界面原型** |
| **SS** | **补充规约** |
| **TP** | **测试计划** |
| **IPP** | **初始项目** |
| **ISPP** | **初始软件项目计划** |
| **SD** | **系统设计与实施** |
| **SR** | **系统需求** |
| **ORS** | **总体需求规约** |
| **DSDP** | **详细软件开发计划** |
| **DDOC** | **设计文档** |
| **STD** | **软件测试设计** |
| **SRD** | **软件发布文档** |
| **UM** | **用户手册** |

格式为：<SYSTEM> <X.Y.Z> [.BL<#>].[NAME]（v）

|  |  |
| --- | --- |
| **<X>** | **代表主发布版本号（例如 1）** |
| **<Y>** | **整数（可选），代表次发布版本号（例如 1.0）** |
| **<Z>** | **整数（可选），代表备选发布版本号（修补程序、移植程序等,例如 1.0.1）** |
| **BL** | **代表基线（内部发布版本）** |
| **#** | **整数，代表内部发布版本号** |
| **NAME** | **各文件名** |

例如：助农系统 1.0（代表助农系统的1.0版本）

丰收宝盒 1.0.requirementBL1.**REQ(1)**（代表计划在 1.0 版中发布的助农系统的内部需求基线版本1的需求规约）(1)代表修改过的第一个版本，若未修改过，则不用书写。

### **3.1.2项目基线**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 阶段 | |  | 基线版本 | 授权人 | 包含内容 |
| 先启阶段 | | 已完成 | 需求基线1.0 | 项目经理 | 确定项目的目标、范围和业务需求。 |
| 精化阶段 | | 9.13—9.22 | 需求基线2.0  设计基线1.0  配置基线1.0  测试基线1.0 | 项目经理  系统分析员，设计员  配置经理  配置经理 | 1. 需求文档的第二版更新，细化和确认所有功能性和非功能性需求。需求已经根据前期的反馈进行了调整和优化。 2. 项目的详细设计，包括系统架构、模块划分、接口设计和用户界面设计等。 3. 系统的初步配置项，包括关键的配置文件、环境设置和版本控制策略。 4. 测试计划、测试用例、测试标准和测试环境的初步定义。 |
| 构建阶段 | 第一阶段 | 9.23-9.29 | 开发基线 1.0  测试基线 2.0  配置基线 2.0 | 项目经理  测试经理  配置经理 | 1. 编写编码规范、接口实现和文档说明，实现初始用户方面的功能模块 2. 更新的测试计划和测试用例，覆盖开发阶段所有实现的功能模块。 3. 系统配置项的进一步完善，包括环境配置、软件版本控制、数据库配置等。 |
| 第二阶段 | 10.1-10.14 | 开发基线 2.0  测试基线 3.0  配置基线 3.0 | 测试经理  配置经理  项目经理 | 1. 实现助农系统大部分功能模块，更新所有实现的功能模块。 2. 更新的测试计划，覆盖所有集成的功能模块和关键业务流程。此时测试基线应包含完整的系统测试、回归测试和集成测试。 3. 包括所有关键配置项的最终调整，包括环境配置、软件版本控制、数据库配置等 |
| 第三阶段 | 10.14-10.20 | 开发基线3.0  测试基线 4.0  配置基线 4.0 | 测试经理  配置经理  项目经理 | 1. 所有开发工作最终完成后的代码库，代码冻结，不再允许进一步开发。 2. 包括完整的系统测试、回归测试、验收测试和用户测试的结果。所有关键业务流程和功能模块经过严格验证，确保系统无重大缺陷。 3. 最终的配置项，包括环境配置、版本控制、数据库配置等。所有的配置文件和系统配置项已经锁定，并且准备好进行发布和部署。 |
| 产品化阶段 | | 10.21-10.24 | 发布基线1.0  操作维护基线1.0 | 项目经理  测试经理 | 1. 这是产品的正式发布版本，包括发布的软件包、配置文件、安装指南和用户手册等。所有的代码、配置、测试结果和文档已经确认，并且产品已经准备好交付给客户或用户。 2. 涵盖产品发布后的操作和维护信息，包括维护计划、技术支持文档、维护指南、系统监控和日志管理配置等内容，保证助农系统的安全。 |

生命周期各阶段结束点相关联的主要里程碑处建立基线。

## 3.2配置和变更控制

### **3.2.1变更请求的处理和审批**

### ****3.2.1.1. 变更请求的提交****

**发起人**：所有发现问题的人员，包括项目团队成员、客户、用户等，都可以提交变更请求。

**提交方式**：通过JIRA提交变更请求。变更请求应包含以下内容：

（1）变更请求描述：说明变更的原因、内容和预期结果。

（2）受影响的系统或模块：明确需要变更的系统部分。

（3）请求者的联系信息。

#### ****3.2.1.2. 变更请求的复审****

**复审人员**：变更控制委员会与开发人员

**复审内容**：

**（1）变更的合理性**：是否有必要进行该变更，或有无其他解决方案，分析该变更对项目各方面的影响。

**（2）变更的可行性**：技术上是否可行，团队是否有足够的资源和时间完成该变更。

**（3）复审结果**：

同意：变更有必要且可行，进入下一步处理。

**拒绝**：变更无必要或不可行，变更请求被拒绝，会向变更请求提交者发送理由。

**待定**：暂时无法判断是否可行，等待项目的进一步深入再做判断。

#### ****3.2.1.3. 变更请求的处理****

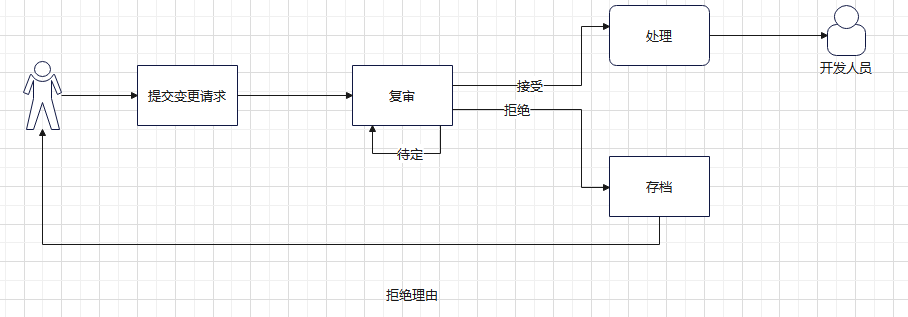
**负责人**：开发人员。

**处理步骤**：

**任务分配**：将变更请求细分为具体的任务，分配给相关人员。

**变更实施**：开发团队或技术人员按照批准的变更请求实施变更。实施过程中，及时记录进展和遇到的问题。

**变更测试**：在变更完成后，测试团队应对变更部分进行功能测试、回归测试和性能测试，以确保变更的有效性和系统的整体稳定性。

****3.2.1.4**** **变更流程**：

### **3.2.2变更控制委员会 (CCB)**

**3.2.2.1 CCB成员及资格标准：**

|  |  |
| --- | --- |
| CCB成员 | 成员资格标准 |
| 项目经理 | 具有项目整体管理的责任，熟悉项目的进度、资源分配、质量要求和主要风险。 |
| 配置经理 | 熟悉项目的配置管理策略，负责对项目配置项的控制和管理。 |
| 测试经理 | 对项目的测试计划和质量要求有深刻认识，能够评估变更对测试工作量和质量的影响。 |
| 变更部分的开发人员 | 对项目的技术架构和实现方案有深入了解，能够评估变更对系统架构、技术设计、代码和集成的影响。 |

**3.2.2.2过程：**

**1.变更请求的初步筛选：**CCB 的相关成员对变更请求进行初步审查，确定是否具备必要性和可行性。初步筛选阶段的结果可能是接受、拒绝或需要进一步信息。

**2.变更请求评估会议：**召开 CCB 会议，讨论和评审待处理的变更请求。评审包括对变更的技术、成本、时间、资源和风险的分析。

**3.变更决策：决定是否通过变更请求。**

## 3.3配置状态统计：

### **3.3.1项目介质存储和发布进程：**

**3.3.1.1.保留策略:**

1)每天下班时将数据备份到U盘中。

2)及时上传至git

**3.3.1.2.事故处理和恢复计划：**

如果出现事故，采用U盘上的数据或github已上传的代码进行恢复。

**3.3.1.3.介质保留方式:**

介质保留方式:联机，脱机合并

类型:git，硬盘。

### **3.3.2报告和审核**

|  |  |  |  |
| --- | --- | --- | --- |
| 缺陷内容 |  | | |
| 存在时间 |  | 滞后时间 |  |
| 缺陷分布 |  | 趋势 |  |
| 危急程度 |  | 报告人 |  |
| 审核人 |  | 审核意见 |  |

# 4.里程碑

# ****4.1内部里程碑：****

**需求基线完成**：

**CM活动**：创建配置管理计划，确保需求文档的版本控制和存档，创建配置项清单以跟踪需求的变更。

**目标**：确认需求的完整性和一致性，为后续设计和开发奠定基础。

**时间点**：在项目启动后的需求分析阶段结束时。

**设计基线完成：**

**CM活动**：更新配置管理计划，确保设计文档的版本控制和存档，记录设计变更的历史，确保所有设计变更都经过审查和批准。

**目标**：确认系统设计的完整性和可行性，为开发阶段提供明确的指导。

**时间点**：在需求分析和设计阶段结束时。

**开发基线完成**：

**CM活动**：进行代码的版本控制和标记，更新配置管理计划以反映开发工作的完成，确保代码库的一致性和完整性。

**目标**：标志着开发阶段的结束，准备进入测试阶段。

**时间点**：在开发阶段结束时。

**测试基线完成**：

**CM活动**：更新配置管理计划，确保测试文档和环境的版本控制，记录测试计划的变更历史。

**目标**：确保测试阶段能够顺利进行，提供测试的规范和标准。

**时间点**：在测试全部结束之后。

**产品发布准备**：

**CM活动**：更新配置管理计划，记录产品的发布版本和配置，确保发布过程中的所有配置项得到适当管理。

**目标**：确保产品能够顺利交付并投入使用。

**时间点**：在产品化阶段结束时。

#### ****4.2客户里程碑：****

**需求确认**：

**CM活动**：记录需求确认的文档和版本，更新配置管理计划以反映客户确认的需求，跟踪需求变更的影响。

**目标**：确保客户需求的准确性和完整性，防止后续阶段的需求变更。

**时间点**：在需求收集阶段结束时。

**设计评审**：

**CM活动**：记录设计评审的结果和客户反馈，更新配置管理计划以反映设计变更，确保所有设计变更都经过客户审批。

**目标**：确保设计满足客户的期望和要求，获取客户对设计的认可。

**时间点**：在设计阶段结束时。

**用户验收测试**：

**CM活动**：记录用户验收测试的结果和反馈，更新配置管理计划以反映验收测试的状态，跟踪和管理发现的缺陷。

**目标**：通过客户测试确认产品的功能和性能，获得客户的最终验收。

**时间点**：在测试阶段完成后，产品发布之前。

**最终交付**：

**CM活动**：更新配置管理计划，记录最终交付的版本和配置，确保交付物的完整性和一致性。

**目标**：完成产品交付，确保客户能够开始使用产品。

**时间点**：在产品化阶段结束时。

**客户培训和支持**：

**CM活动**：更新配置管理计划以包含培训和支持活动的记录，跟踪培训和支持服务的需求和反馈，确保支持过程中的配置项得到适当管理。**目标**：确保客户能够正确使用产品，并获得所需的技术支持。

**时间点**：在产品交付后，培训和支持服务开始时。

#### ****4.3更新 CM 计划的时间点：****

#### **需求基线完成后，设计基线完成后，开发基线每一个阶段完成后，产品发布准备阶段后，客户验收和交付时。**

# 5.培训和资源

项目经理、测试经理、配置经理自行学习git和各种其他开发工具或管理工具来实施各项CM活动。

# 6.分包商和厂商软件控制

暂无。