旅游信息管理系统

配置管理计划

版本 1.0

修订版历史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2022/10/18 | V1.0 | 初步确立 | 王景 |
|  |  |  |  |

目录

[1. 简介](#1.                  Introduction)

[1.1 目的](#1.1               Purpose)

[1.2 范围](#1.2               Scope)

[1.3 定义、首字母缩写词和缩略语](#1.3               Definitions, Acronyms and Abbreviations)

[1.4 引用](#1.4               References)

[1.5 概述](#1.5               Overview)

[2. 软件配置管理](#2.                  Software Configuration Management)

[2.1 组织、职责和接口](#2.1               Organization, Responsibilities and Interfaces)

[2.2 工具、环境和基础设施](#2.2               Tools, Environment and Infrastructure)

[3. 配置管理活动](#3.                  The CM Program)

[3.1 配置标识](#3.1               Configuration Identification)

[3.1.1 标识方法](#3.1.1          Identification Methods)

[3.1.2 项目基线](#3.1.2          Project Baselines)

[3.2 配置和变更控制](#3.2               Configuration and Change Control)

[3.2.1 变更请求的处理和审批](#3.2.1          Change Request Processing and Approval)

[3.2.2 变更控制委员会 (CCB)](#3.2.2          Change Control Board (CCB))

[3.3 配置状态统计](#3.3               Configuration Status Accounting)

[3.3.1 项目介质存储和发布进程](#3.3.1          Project Media Storage and Release Process)

[3.3.2 报告和审计](#3.3.2          Reports and Audits)

[4. 里程碑](#4.                  Milestones)

[5. 培训和资源](#5.                  Training and Resources)

[6. 分包商和厂商软件控制](#6.                  Subcontractor and Vendor Software Control)

配置管理计划

# 1.                  简介

## 1.1               目的

本文档目的在于对旅游管理系统项目进行软件配置管理即采用统一且有效的方法进行软件变更和版本管理，为该系统的研发提供了一套管理办法和活动原则以此来对旅游管理系统项目进行标识、组织和控制修改，提高软件质量，降低软件开发成本，使错误降为最小并最有效地提高生产效率。由于在软件建立时变更是不可避免的，而变更加剧了项目中软件开发者之间的混乱。软件配置管理贯穿于整个软件生命周期，应用于整个软件工程过程，其作用就是为了标识变更，控制变更，确保变更正确实现并向其他有关人员报告变更。

本文档内容主要参考研发中心相关的ISO程序和制度文档，并在这基础上整理成适合本项目的软件配置管理，为项目经理、配置管理员及相关人员提供日常的配置管理操作步骤。

## 1.2               范围

该软件开发过程中需要被管理的对象如下：

1. 软件建立过程中所有的工程代码。
2. 项目管理产生的文档：初始软件项目计划，软件需求规范，用户界面模型，详细软件项目计划，软件测试计划，配置管理计划，设计文档，软件测试设计，测试报告，配置库，交付产品，用户手册。
3. 实验输出生产过程所包含的原始图、表或其他类似文件。
4. 小组评审文件及教师评阅文件；
5. 其他相关参考资料。

## 1.3               定义、首字母缩写词和缩略语

软件配置管理：

简称SCM（Software Configuration Management 的缩写），是在项目开发中，标识、控制和管理软件变更的一种管理。配置管理的使用取决于项目规模和复杂性以及风险水平。软件的规模越大，配置管理就显得越重要。

基线(BaseLine)：

是项目储存库中每个工件版本在特定时期的一个“快照”。它提供一个正式标准，随后的工作基于此标准，并且只有经过授权后才能变更这个标准。建立一个初始基线后，以后每次对其进行的变更都将记录为一个差值，直到建成下一个基线。

配置管理员：

项目组中负责配置管理工作的角色，该角色可以兼职。在某一开发阶段通过评审或某一质量检查点通过审核后，配置管理员负责统一添加或修改相关文档的最新有效版本以及审批人签字。

配置标识（Configuration Identification）：

对软件项目在开发过程中的资源进行标识，以便识别。

配置检查（Configuration Audit）：

对软件配置管理过程中的行动进行检查。

## 1.4               引用

《研发中心配置管理制度》

《产品的标识与可追溯性程序》

《开发手册》

## 1.5               概述

配置管理计划书主要是用来介绍本旅游管理项目过程中的配置管理部分的计划方案，主要包括以下几个部分的内容：

第一章节用来简要介绍配置管理计划的主要含义和目的，并指明软件开发过程中需要被管理的对象以及给出定义、缩略语等的含义。

第二章节主要介绍该项目完成过程中各个成员的具体分工，对于配置管理所用的工具、环境和基础设施，这是本项目过程中主要针对配置管理部分所采取的方法和实验工具和对本项目组在后续实验过程中的要求。

第三章节主要描述配置管理活动即具体的配置管理计划，包括该项目的标识规范、项目基线以及配置变更所遵循流程和状态控制，明确配置管理的进度安排，对后续工作进行工作量与工作时间的预计。

# 2.                  软件配置管理

## 2.1               组织、职责和接口

职责分配表：

|  |  |  |  |
| --- | --- | --- | --- |
| 实验编号 | 配置管理任务 | 具体说明 | 负责成员 |
| 实验一 | 确立成员分工与成员职责 | 《人员分工表》 | 全体成员 |
| 实验二 | 需求分析员挖掘分析项目需求并完成文档 | 《软件需求规范》 | 张仕易 |
| 实验二 | 项目经理制定总体项目计划，分析员完成软件初步设计 | 《初始软件项目计划》  《设计文档》 | 沈晨琳、孟娇 |
| 实验三 | 项目经理设计详细项目计划，界面设计员完成界面设计模型、测试员设计测试计划  配置员完成系统配置库设计 | 《详细软件项目计划》  《用户界面原型》  《软件测试计划》  《配置管理计划》  配置库 | 沈晨琳、殷志谦、陈明惠、王景 |
| 实验四 | 代码编写 |  |  |
| 实验五 | 测试员设计测试用例，并对系统进行测试，完成测试分析报告 | 《软件测试设计》  《测试报告》 | 陈明惠 |
| 实验六 | 交付产品，提供用户手册 | 《用户手册》 | 沈晨琳 |

配置库设计：

在本项目的实施过程中，将配置库分为开发配置库、受控配置库和产品配置库三种。

|  |  |  |
| --- | --- | --- |
| 类别 | 内容 | 权限 |
| 开发库 | 开发人员编写与修改代码 | 项目经理和开发人员具有所有权限。 |
| 受控库 | 测试版本代码存放的地方，需要开发人员提交测试申请来修改 | 各成员都具有读写权限 |
| 产品库 | 测试通过后版本存放的地方，需要测试报告来驱动修改 | 配置管理员具有所有权限 |

目录结构：

1项目名称

——1.1库名

——1.1.1个人工作区：

——代码：存放项目代码

——文档：存放各类文档

需求分析

概要设计

详细设计

软件测试

项目交付

——报告：存放项目的各类报告、评审意见

——1.1.2小组工作区：

——代码：存放项目代码

——文档：存放各类文档

需求分析

概要设计

详细设计

软件测试

项目交付

——报告：存放项目的各类报告、评审意见

——1.1.3提交区：

——代码：存放项目代码

——文档：存放各类文档

需求分析

概要设计

详细设计

软件测试

项目交付

——报告：存放项目的各类报告、评审意见

## 2.2               工具、环境和基础设施

服务器软件环境

|  |  |
| --- | --- |
| 软件名称 | 作用 |
| Windows10 | 操作系统 |
| Git、GitHub | 配置管理软件 |

服务器硬件环境

|  |  |
| --- | --- |
| 硬件名称 | 说明 |
| 网络 | 局域网 |
| 服务器 | PC服务器 |
| 客户机 | 项目组成员各自的计算机 |

# 3.                  配置管理活动

## 3.1               配置标识

### 3.1.1          标识方法

**文档命名规范：**

文档名称\_date\_vx.y.扩展名

其中文档名称指代该文档标准全称，date表示该文档完成日期，vx.y表示该文档演化版本号，扩展名代表文件扩展全名。

**版本号演化规范：**

该软件过程中，版本号保留在文档文件名中，其形式为Vx.y。其中v代表version；x为整数，代表主版本号；y也为整数，代表次版本号；x从0开始，y从0开始，遇到小的变更次版本号加一，遇到大的变更主版本号加一，且次版本号置为0。v1.0一般代表一个文档的初稿完成版本。v0.x一般为未完成的文档（或者文档初稿各部分），而未完成的文档各部分编号按照项目成员提交的先后顺序编号为v0.1,v0.2,v0.3等,若出现了修改版本，则版本号演化为v0.11, v0.12等。而每次修改均应更新版本号，尽量避免对历史版本文档的直接修改。

### 3.1.2          项目基线

|  |  |  |
| --- | --- | --- |
| 基线类别 | 内容 | 授权人 |
| 需求基线 | 在需求分析规格说明书通过评审后建立，此时客户的需求和产品需求一个是全面、清晰、准确并且文档化的。  包含文档：  《软件需求规范》 | 需求分析员 |
| 设计基线 | 在详细设计通过评审后建立。设计基线建立后开发人员可以进行代码开发。在开发过程中必然会遇到需求变更和设计变更的活动，这些变更需要被完整记录并且变更的内容要及时反应到需求文档和设计文档中。保证需求和设计文档内容完整的有效办法是指定文档的唯一责任人。  包含文档：《详细软件项目计划》《配置管理计划》《软件测试计划》《用户界面原型》 | 分析员、配置员、界面设计员 |
| 测试基线 | 是开发人员完成开发后，将软件系统交给测试人员测试时对之前所有开发成果的标识。建立测试基线需要设计、开发人员提供《功能清单》和源码（含ant编译脚本）。在建立测试基线时，根据测试人员的要求，设计、开发人员还应该提供相应的讲解和培训。  包含文档：《测试报告》、《软件测试设计》 | 测试员 |
| 发布基线 | 是在测试人员完成测试工作后建立。建立测试基线时，测试中发现的所有bug应该已经fixed或者未fixed但不影响系统使用。未fixed的bug作为遗留问题被记录下来。  产品发布后，所有产品的安装根据用户需求从已经发布的版本中选择或者进行增量开发。  包含文档：《用户手册》 | 项目经理 |

## 3.2               配置和变更控制

### 3.2.1          变更请求的处理和审批

**具体要求：**

对该项目各个子系统及其专用支持软件的基线及其集成系统的任何修改，必须得到项目经理的批准并在本项目配置经理处备案才能进行配置更改。

更改完成后的文档和代码等，需得到项目经理认可，提交给配置管理员后，由配置管理员签入受控配置库。受控配置库中的文档，在文档末尾必须有修改记录部分，包括修改人、修改日期、修改内容等项，每次对于受控配置库中文档的修改，必须填写这些项。

**变更流程：**

1、变更申请：变更人向项目经理提出申请，并提交申请报告，报告应记录变更的提出人、日期、申请变更的内容等信息。  
2、变更评估：项目经理组织其他相关人员对变更的影响范围、严重程度、经济和技术可行性进行系统分析。  
3、变更决策：由CCB决定是否实施变更。若变更审核不通过，拒绝变更请求后需要给出拒绝理由。  
4、变更实施：CCB指定变更工作人员并制定变更计划，变更工作人员在受控状态下实施变更。  
5、变更验证：由配置管理人员或受到变更影响的人对变更结果进行评价，确定变更结果和预期是否相符、相关内容是否进行了更新、工作产物是否符合版本管理的要求。  
6、沟通存档：将变更后的内容通知可能会受到影响的人员，并将变更记录汇总归档。如提出的变更在决策时被否决，其初始记录也应予以保存。

### 3.2.2          变更控制委员会 (CCB)

**组成成员：**项目经理、系统分析员、测试员、配置员、开发人员、需求分析员

**遵循标准：**

提出申请：任意成员提出变更请求，需向项目经理提交申请，

审核：项目经理组织其他人员讨论，针对变更的目的、要求和影响来综合决策，

审核同意：一致同意变更请求后，项目经理安排相关的变更实施负责人和相关联的协作组织。审核不通过：拒绝变更请求后需要给出拒绝理由。

## 3.3               配置状态统计

### 3.3.1          项目介质存储和发布进程

配置状态报告对于大型开发项目的成功起着至关重要的作用。它提高了所有开发人员之间的通信能力，避免了可能出现的不一致和冲突。它通过支特创建和修改记录、管理报告配置项的状态或需求变化并审核变化来实现，它提供用户需要的功能，跟腙任意模式的软件项，提供完整的各种变化的历史版本和汇总信息。

配置状态报告的内容需要包括以下各项：

(1)各变更请求概要：变更请求号、日期、申请人、状态、估计工作量、实际工作量、发行版本、变更结束日期。

(2)基线库状态。

(3)发行信息。

(4备份信息。

(5)配置管理工具状态。

(6)配置管理培训状态。

在变更请求批准后，实施变更需要一段时间，要设置一种管理手段来反映变更所处的状态，这就是变更状态说明

它可供项目经理和CCB追腙变更的情况。状态说明的信息可以通过变更请求和故障报告得到，变更状态可分为活动正在

实施变更到、完成状态（已完成变更到和未列入变更状态3种。

### 3.3.2          报告和审计

[说明所需报告和配置审计的内容、格式和目的。

报告用于在项目和产品生命周期中的任意给定时间对“产品质量”进行评估。如果根据变更请求来报告缺陷，就可以提供一些有用的质量指标。因此，应提醒管理人员和开发人员多注意特别关键的开发领域。 缺陷通常按危急程度（高、中和低）分类。可以依据以下各项来报告缺陷：

         存在时间（基于时间的报告）：各种缺陷已经打开了多久？在生命周期中，从发现缺陷到修复缺陷有多长的“滞后时间”？

         分布（基于计数的报告）：在按照拥有者、优先级或修复状态划分的不同类别中各有多少个缺陷？

         趋势（与时间和计数有关的报告）：在一段时间内发现并修复的缺陷累计有多少个？发现并修复缺陷的频率为多少？就打开的缺陷和关闭的缺陷而言，它们之间的“质量差距”有多大？解决缺陷所用的平均时间为多长？]

|  |  |  |
| --- | --- | --- |
| 审核分类 | 审核内容 | 检查情况 |
| 发布审核 | 发布文档是否清楚地定义发布的范围，包括应被纳入的更改请  求？ |  |
|  | 所有已知缺陷/毛病(bug)是否已文档化？ |  |
|  | 是否有适当的文档，它标识重建该发布所需的环境（编译器版本、  OS 版本、compilation flags，等等）？ |  |
|  | 是否有适当的文档，它说明构成该发布的成分及成分的版本？ |  |
|  | 发布的所有项是否彼此同步（在时间上一致）？ |  |
|  | 是否采用正确存储库中的正确成分的正确版本生成发布？ |  |
| 存储库/配  置项审核 | 存储库是否按 SCM 计划定义？ |  |
|  | 项是否已经进入正确的库？ |  |
|  | 是否按 SCM 计划中规定的命名约定项命名? |  |
|  | 是否按照 SCM 计划，规定项的版本号？ |  |
|  | 是否按照 SCM 计划中规定的事件已经将所有项入库？例如：测试  完成、客户的评审意见已采纳 |  |
|  | 项是否有所要求的文档以识别项、版本和更改历史？ |  |
| 更改实施  审核 | 是否全部所要求的更改请求均已结束？ |  |
|  | 是否更改请求标识出全部拟更改的项？ |  |
|  | 更改请求中所标识的全部要更改的项均已更改，被 QC 和在所要求的 QC 后入库？ |  |
|  | 是否可能在项的任何两个版本中间区分更改？ |  |
|  | 项的文档是否足够，能向后追踪更改到相应的更改请求？ |  |
|  | 是否有恰当方法能回到以前的版本？ |  |
| 审核的其  他方面 | 是否对库作了恰当的备份? |  |
|  | 是否已测试过从备份中恢复? |  |
|  | 在群组成员的工作目录中是否有任何未经许可的成分？ |  |
|  | 是否有恰当的保密/批准手续以保证只有经授权的群组成员才能进行入库/出库？ |  |

配置管理员应配合研发中心产品管理部定期对项目进行配置管理的审核。在审核过程

中，提供所需要的配置管理计划及相关资料，在项目开发结束后，需提交所有关于项目的软件配置库。

# 里程碑

|  |  |
| --- | --- |
| 里程碑 | 具体内容 |
| 1.需求分析已确立 | 1. 系统的需求分析全部完成，已形成相应的文档：软件需求规范及其它附属文档。 2. 需求分析说明书已通过评审，需求分析阶段已结束，可以进入设计阶段。 |
| 2.概要设计 | 1. 系统的概要设计全部完成，已形成相应的文档：初始软件项目计划、设计文档及其它附属文档。 2. 系统概要设计评审通过，可以进入详细设计阶段。 |
| 3.详细设计 | 1. 系统的详细设计全部完成已形成相应的初始文档：详细软件项目计划、用户界面原型、配置管理计划、配置库及附属文档。 2. 系统详细设计通过，可以进入软件测试计划阶段。 |
| 4.软件测试计划 | 1. 测试需求已经确定并完成。 2. 已形成相应的初始文档：软件测试计划。 3. 软件测试计划通过，可以进入编码阶段 |
| 5.编码 | 1. 系统的编码全部完成。 2. 系统所有程序已经经过调试并确定可以运行。 3. 编码阶段通过，可以进入测试阶段。 |
| 6.测试设计 | 1. 测试用例已经覆盖所有测试需求。 2. 已形成相应的初始文档：软件测试设计。 3. 测试设计通过，可以进入系统测试阶段。 |
| 7.系统测试 | 1. 系统测试完成，所发现的所有缺陷已得到妥善处理。 2. 符合系统测试退出条件。 3. 已完成相应的文档：测试报告 4. 系统测试通过，可以进入交付阶段 |
| 8.项目交付 | 1. 交付产品。 2. 已完成相应的文档：用户手册。 3. 项目结束。 |

# 5.                  培训和资源

设计阶段：项目经理需对其他所有成员展开培训，描述项目的具体内容、目标以及项目规划，以便其他人员展开工作。

编码阶段：需求分析员与分析员需对开发人员进行培训，让其熟悉系统具体需求以便做出准确的功能开发。

测试阶段：需求分析员、分析员、开发人员需对测试人员进行培训，让其知悉客户与系统的具体需求，以便对开发人员写出的代码进行准确测试，开发人员需使测试人员了解代码组织等，便于测试人员展开测试流程。

交付：其他人员向项目经理交付具体产品。