<ATM机模拟项目>

配置管理计划

版本 <1.0>

作 者： 徐珮轩

完成日期： 2022.10.19

签 收 人： 徐珮轩

签收日期： 2022.10.19

修订版历史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <19日/10月/2022年> | <1.0> | <ATM模拟项目配置管理计划定稿> | <徐珮轩> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 简介](#1.                  Introduction)

[1.1 目的](#_1.1__)

[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_)

[1.4 引用](#_1.4__)

[1.5 概述](#_1.5__)

[2. 软件配置管理](#_2.__)

[2.1 组织、职责和接口](#_2.1__)

[2.2 工具、环境和基础设施](#_2.2__)

[3. 配置管理活动](#_3.__)

[3.1 配置标识](#_3.1__)

[3.1.1 标识方法](#_3.1.1__)

[3.1.2 项目基线](#_3.1.2__)

[3.2 配置和变更控制](#_3.2__)

[3.2.1 变更请求的处理和审批](#_3.2.1__)

[3.2.2 变更控制委员会 (CCB)](#_3.2.2__)

[3.3 配置状态统计](#_3.3__)

[3.3.1 项目介质存储和发布进程](#_3.3.1__)

[4. 里程碑](#_4.__)

配置管理计划

# 1.                  简介

## 1.1               目的

本文档目的在于定义那些描述要在ATM机模拟项目软件产品开发中执行配置和变更控制管理 (CM) 方式的步骤和活动，并定义其负责人；建立产品目录结构；说明配置标识方法；确定基线的建立的时间、授权、标识方法等。它详细说明了活动时间表、分配的职责、以及必须的资源，旨在便于对ATM机模拟系统进行软件配置管理，提高软件质量，降低软件开发成本。

## 1.2               范围

配置管理计划的范围应是系统开发过程中的所有工件。

## 1.3               定义、首字母缩写词和缩略语

**软件配置管理：**简称SCM（Software Congfiguration Management），是在项目开发中对标识、控制和软件变更的一种管理。

**基线：**简称BL（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。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。随着工作的进展，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。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。调整基线将把集成工作区中的文件并入开发工作区。

**配置标识**：简称CI（Configuration Identification）对软件项目在开发过程中的资源进行标识，以便识别。

|  |  |
| --- | --- |
| **PLN** | **项目计划** |
| **REQ** | **需求文件** |
| **USC** | **用例** |
| **MOD** | **模型文件** |
| **SRC** | **源代码文件** |
| **INT** | **公共接口** |
| **TST** | **测试脚本和结果** |
| **DOC** | **文档（用户文档、发布说明文档）** |
| **BIN** | **可执行文件** |

## 1.4               引用

课件PPT《3.2 配置管理-配置管理计划》

《研发中心配置管理制度》

《CCB变更控制管理制度》

## 1.5               概述

本项目配置管理工具采用Git，便于无中心服务器的多人协作开发及管理，各开发人员可以使用Github拉取配置经理建立的配置库，在本地仓库按照配置管理计划相关规定进行开发和远程仓库的推送，配置经理应及时对项目进行检查，控制以及管理。web URL如下：

https://github.com/crabbby/T18

# 2.                  软件配置管理

## 2.1               组织、职责和接口

配置管理活动总体由配置经理控制、检查、管理，由项目经理审批如配置标识、配置变更控制管理，具体项目工件由对应开发人员进行开发，具体分工如下：

项目经理：初始项目计划的编写与更新、详细项目计划的编写与更新、总体项目配置管理的审查（如变更请求的审批）。

系统分析员：软件需求规约的编写与更新。

设计员：设计文档的编写与更新。

测试经理：软件测试计划的编写与更新、软件测试设计的设计与更新、测试报告的编写与更新。

配置经理：配置管理计划的编写与更新、配置库配置检查以及管理、总体配置活动的负责（如基线的建立、确定标识命名规则等）。

界面设计员：用户界面原型的编写与更新、用户手册的编写与更新。

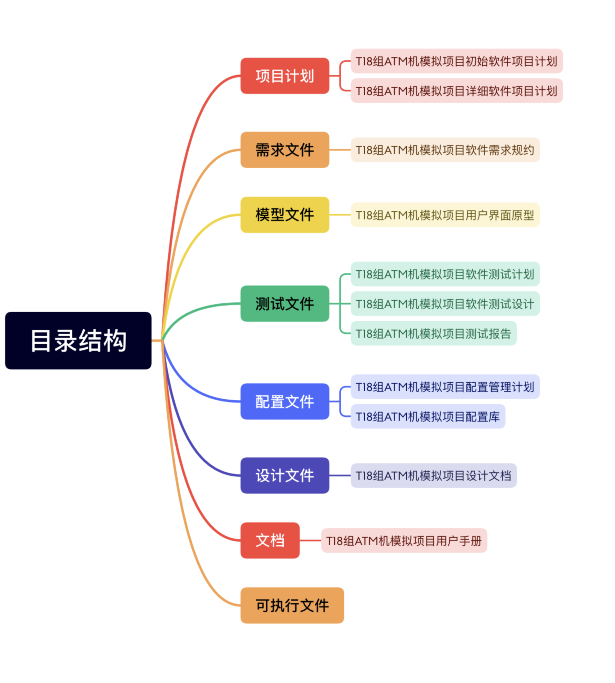
程序员：可执行文件系统的编写、更新以及管理。

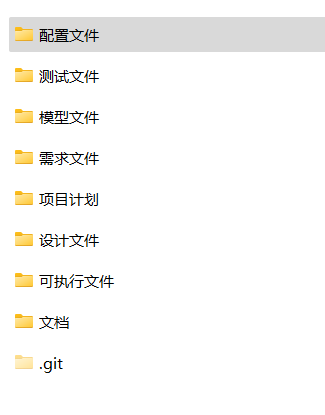
## 2.2               工具、环境和基础设施

**2.2.1 配置环境及工具**

|  |  |
| --- | --- |
| **名称** | **作用** |
| Windows | 操作系统 |
| Git | 配置管理软件 |

**2.2.2 产品目录结构**

****

****

# 3.                  配置管理活动

## 3.1               配置标识

### 3.1.1          标识方法

1. **项目工件的命名：**

项目工件的命名应按照如下格式命名：

**小组名称+项目名称+资料名称：**

|  |  |
| --- | --- |
| **项** | **说明** |
| 小组名称 | T18组 |
| 项目名称 | ATM机模拟项目 |
| 资料名称 | 初始软件项目计划  需求规约文档  用户界面原型  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  …… |

**例如：**

**T18组ATM机模拟项目需求规约文档**

**T18组ATM机模拟项目配置管理计划**

1. **产品版本的标识及命名：**

产品版本的标识应按照如下格式命名：

**小组名\_项目名\_标识字符：**

|  |  |
| --- | --- |
| **项** | **说明** |
| 小组名称 | T18 |
| 项目名称 | ATM |
| 资料名称 | R(发行版本)  B(测试版本)  BL(基线版本)  GUI(GUI子系统)  ……… |

**例如：**

**T18\_ATM\_R1.0(1.0 发布版本)**

**T18\_ATM\_B1.0(1.0 测试版本)**

**T18\_ATM\_GUI.R1.0.BL01（准备以 1.0 版本发布的 GUI 系统的内部发布基线版本第1号）**

### 3.1.2          项目基线

1. 项目基线建立的时机：

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：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）。经过效率和资源的综合考虑，将基线建立的时间确定在四个阶段：

1. 需求分析阶段
2. 详细设计阶段
3. 编码阶段
4. 测试阶段
5. 建立基线的授权：

基线的建立由配置管理员根据项目进展情况（或项目组的要求）和基线计划规定，提出创建构造基线或发行基线的请求，由项目经理授权，用配置管理系统中的配置项创建构造基线或发行基线，把基线中包含的配置项集合形成文件，使现行的基线集可供使用。

1. 基线的标识：

基线的标识应按照如下格式命名：

基线+版本号

|  |  |
| --- | --- |
| **项** | **说明** |
| 基线 | BL |
| 版本号 | 01(先启阶段)  02(精化阶段)  03(构建阶段)  04(产品化阶段) |

例如:

BL01

## 3.2               配置和变更控制

### 3.2.1          变更请求的处理和审批

1.确认变更请求

2.描述变更请求

3.将变更请求带回本部分析

（1）对时间的影响

（2）对成本、质量、干系人等综合影响

4.变更是否有必要，判断对项目有利/有害

5.反馈

6.正式变更控制程序

\*注意：必须走流程，最好有正式文件

7.更新计划，基准

8.实施

### 3.2.2          变更控制委员会 (CCB)

变更请求分为四类：1.纠正措施：实际绩效跑偏之后拉回正常(计划).2.预防措施：降低负面后果提前做的工作.3.缺陷补救：可交付物已出现问题进行补救.4.更新：变动计划（增加需求/修改原计划等）时提出的变更。

变更请求提出之后，需要统一由“实施整体变更控制”这个过程来批准是否执行。CCB或相关人员会在这个过程中对变更进行审批，决定批准或者否定这个“变更请求”，继而形成一份“批准的变更请求”（或者“否定的变更请求”）。批准与否的状态，需要记录到一个叫做“变更日志”的项目文件中。

## 3.3               配置状态统计

### 3.3.1          项目介质存储和发布进程

在项目开发实施过程的各个阶段，配置管理员应定期做好软件配置库的备份，以防造成劳动成果的丢失而给整个项目带来的严重损失。

备份可按照要求定期（按周或月）进行。在每个阶段或里程碑处在做完基线工作后应进行备份。备份文件应存放在不同的地方。

本项目的备份按如下方式进行：

定期备份时间为每个月备份一次，备份时间定于每个月的最后一个星期日；

当在月末（大于当月20日）达到一个里程碑时，对配置库进行一次备份，取消当月月备份；

当在月中（大于当月10日，小于等于当月20日）达到一个里程碑时，对配置库进行一次备份，当月月备份不变。

当在月初（小于当月10日）达到一个里程碑时，不需要对配置库再进行一次备份，当月月备份不变；

备份的文件要明确标明备份日期，应保存在可靠的计算机中。

# 4.                  里程碑

|  |  |
| --- | --- |
| 1. 需求分析已确立 | 系统（或所有已确定子系统）的需求分析全部完成  已形成相应的需求分析说明书及其它附属文档  需求分析说明书已通过公司评审或与客户一致认为需求分析阶段已结束，可以进入设计阶段 |
| 1. 概要设计完成 | 系统（或所有已确定子系统）的概要设计全部完成  已形成相应的概要设计说明书及其它附属文档  概要设计说明书已通过公司评审或与客户一致认为概要设计阶段已结束，可以进入详细设计阶段 |
| 1. 详细设计完成 | 系统（或所有已确定子系统）的详细设计全部完成  已形成相应的详细设计说明书及其它附属文档  详细设计说明书已通过公司评审或与客户一致认为详细设计阶段已结束，可以进入编码阶段 |
| 1. 编码完成 | 系统（或所有已确定子系统）的编码全部完成  系统所有程序已经经过调试并确定可以运行  已通过公司评审或与客户一致认为编码阶段已结束，可以进入系统测试阶段 |
| 1. 测试计划完成 | 测试需求已经确定并完成；  已形成相应的测试计划说明书及其它附属文档 |
| 1. 测试设计完成 | 测试用例已经覆盖所有测试需求  已形成相应的测试用例说明书及其它附属文档 |
| 1. 系统测试完成 | 系统测试完成，所发现的所有缺陷已得到妥善处理  符合系统测试退出条件  已完成测试分析报告 |
| 1. 项目结束 | 上线成功 |

其中，应该根据项目产出及时更新项目配置库以及按需更新管理计划。