**<ATM>**

**配置管理计划**

**版本 <1.0>**

**修订历史记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018/11/7 | 1.0 |  | 薛豪啸 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

1. 简介 2

1.1 目的：规范配置管理活动，确保配置项正确地唯一标识并易于存取，保证基准配置项的更改受控，明确基线状态，在贯穿整个软件生命周期中建立和维护项目产品的完整性和可追溯性。 2

1.2 定义、首字母缩写词和缩略语： 2

1.3 概述： 3

2. 软件配置管理 3

2.1 组织、职责和接口 3

2.2 工具、环境和基础设施 3

3. 配置管理活动 4

3.1 配置标识 4

3.1.1 标识方法 4

3.1.2 项目基线 5

3.2 配置和变更控制 6

3.2.1 变更请求的处理和审批 6

3.2.2 变更控制委员会 (CCB) 7

3.3 配置状态统计 7

3.3.1 报告和审计 7

4. 里程碑 8

配置管理计划

# 简介

## 目的：规范配置管理活动，确保配置项正确地唯一标识并易于存取，保证基准配置项的更改受控，明确基线状态，在贯穿整个软件生命周期中建立和维护项目产品的完整性和可追溯性。

## 定义、首字母缩写词和缩略语：

**软件配置管理（Software Configuration Management，SCM）**

软件配置管理是对软件修改进行标识、组织和控制的技术，用来协调和控制整个过程。是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程，确保软件开发者在软件生命周期中各个阶段都能得到精确的不同版本的产品配置。

**配置（Configuration）**

配置是在技术文档中明确说明并最终组成软件产品的功能或物理属性。因此配置包括了即将受控的所有产品特性，其内容及相关文档、软件版本、变更文档、软件运行的支持数据，以及其他一切保证软件一致性的组成要素，相对与硬件类配置，软件产品的配置包括更多的内容并具有易变性。

**配置项（Configuration Item，CI）**

凡是纳入配置管理范畴的工作成果统称为配置项（Configuration Item, CI），配置项逻辑上组成软件系统的各组成部分，一般是可以单独进行设计、实施和测试的。一个纯软件的CIs通常也称之为软件配置项（Computer Software Configuration Items，CSCIs）。

配置项主要有两大类：

1)属于产品组成部分的工作成果，例如需求文档、设计文档、源代码、测试用例等；

2)项目管理和机构支撑过程产生的文档。这些文档虽然不是产品的组成部分，但是值得保存。

每个配置项的主要属性有：名称、标识符、文件状态、版本、作者、日期等。所有配置项都被保存在配置库里，确保不会混淆、丢失。配置项及其历史记录反映了软件的演化过程。

**基线（Baseline）**

在配置管理系统中，基线就是一个CI或一组CIs在其生命周期的不同时间点上通过正式评审而进入正式受控的一种状态，些配置项构成了一个相对稳定的逻辑实体，而这个过程被称为“基线化”。每一个基线都是其下一步开发的出发点和参考点。基线确定了元素（配置项）的一个版本，且只确定一个版本。一般情况下，基线一般在指定的里程碑（Milestone）处创建，并与项目中的里程碑保持同步。每个基线都将接受配置管理的严格控制，基线中的配置项被“冻结”了，不能再被任何人随意修改，对其的修改将严格按照变更控制要求的过程进行，在一个软件开发阶段结束时，上一个基线加上增加和修改的基线内容形成下一个基线。

基线的主要属性有：名称、标识符、版本、日期等。通常将交付给客户的基线称为一个“Release”，为内部开发用的基线则称为一个“Build”。

建立基线的好处：

1)重现性：及时返回并重新生成软件系统给定发布版的能力，或者是在项目中的早些时候重新生成开发环境的能力。当认为更新不稳定或不可信时，基线为团队提供一种取消变更的方法。

2)可追踪性：建立项目工件之间的前后继承关系。目的是确保设计满足要求、代码实施设计以及用正确代码编译可执行文件。

3)版本隔离：基线为开发工件提供了一个定点和快照，新项目可以从基线提供的定点之中建立。作为一个单独分支，新项目将与随后对原始项目（在主要分支上）所进行的变更进行隔离。

## 概述：

包含了软件配置管理、配置管理活动。

# 软件配置管理

## 组织、职责和接口

配置管理员：薛豪啸

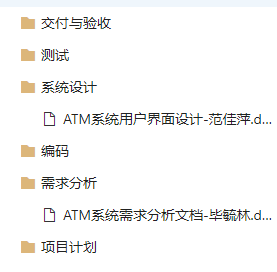
## 工具、环境和基础设施

操作系统：windows 10

软件环境：Visual Studio 2017、Team Foundation Server 2018

Visual Studio 用于编码测试以及连接团队项目，Team Foundation Server 用于版本控制。

**2.3 产品目录结构**



# 配置管理活动

## 配置标识

### 标识方法

命名规则：AAA.YYMMDD.Version

AAA：配置项名称

YYMMDD：YY年份，MM月份，DD日期

Version：版本号

|  |  |  |
| --- | --- | --- |
| 分类 | 配置项名称 | 命名示例 |
| 项目计划 | 软件开发计划 | 软件开发计划.081107.v1.0 |
| 配置管理计划 | 配置管理计划.081107.v1.0 |
| …… |  |
| 需求分析 | 需求分析文档 | 需求分析文档.081107.v1.0 |
| …… |  |
| 系统设计 | 用户界面设计文档 | 用户界面设计文档.081107.v1.0 |
| 用例设计文档 | 用例设计文档.081107.v1.0 |
| …… |  |
| 编码 | 源代码 | 源代码.081107.v1.0 |
| 测试 | 测试计划 | 测试计划.081107.v1.0 |
| 测试报告 | 测试报告.081107.v1.0 |
| …… |  |
| 发布 | 产品版本 | 产品版本.081107.v1.0 |
| …… |  |

### 项目基线

在不同阶段中的各次迭代结束时生成基线。由项目经理授权建立基线。

## 配置和变更控制

### 变更请求的处理和审批

**3.2.1.1变更请求的提出**

3.2.1.1.1　由发起者（客户、最终用户或开发部门）确定变更，填写《变更请求/评审单》，描述变更原因和变更内容，并提交给CMO。

3.2.1.1.2　CMO对填写的申请表是否清晰、明确和完整性进行审查，若CMO发现变更不明确或不完整，应返回申请表给发起者。CMO对通过审查的变更申请分配变更ID，以便跟踪和记录变更信息。

**3.2.1.2　变更评估**

3.2.1.2.1　CMO将《变更请求/评审单》发送给项目经理（或者其他授权人员），由项目经理负责对变更进行评估。

3. 2.1.2.2　变更控制的一个重要环节就是变更评估，变更评估要分析每个变更对系统功能、接口、成本、进度以及约定需求的影响，同时还要分析对软件安全性、可靠性、可维护性、可移植性和性能的影响。

3. 2.1.2.3　变更评估产生的文档应描述若实施变更必须变更的配置项、文档和资源；变更评估文档在完成变更评估后发送给CMO。

3. 2.1.2.4　CMO收到评估后的《变更请求/评审单》后，更新变更记录，并安排CCB会议日程。

**3.2.1.3　变更审核**

3. 2.1.3.1　CCB对提交的变更申请进行审核，并根据变更评估确定变更的影响级别；CCB审核可能的结果有三种：接受变更；拒绝变更；延期变更。CCB也可能需要更多的变更分析的信息。

3. 2.1.3.2 CCB批准的变更，由CMO将变更项目发送到指定的开发人员（Assigner）进行下一步的实施变更工作；对于拒绝的变更，由CMO将CCB拒绝变更的原因发送给发起者，并保存《变更请求/评审单》，更新变更记录，关闭变更活动；需要进一步分析的，由CMO将变更项目随同CCB的Question Sheet发送给评估分析人员；对于延期的变更，由CMO对变更的相关文档进行归档，以便在适当时机提交CCB审核。

3. 2.1.3.3　CMO负责整理CCB会议记录，填写《变更请求/评审单》中相应审核项；更新变更记录，如果是接受变更，还需将要变更的CIs状态改为“修改中”；将变更文档分发给相关人员。

**3.2.1.4　变更实施**

3.2.1.4.1　变更被批准后，PL负责将要变更的CIs以及相关文档迁移至变更负责人的Private Branch上，由变更负责人开始实施变更，并详细记录变更的内容；项目管理部门要对变更的实施进行跟踪。

3.2.1.4.2　对于代码变更，必须修改设计、代码、测试以及变更正确性的验证。而且与变更相关的文档必须修订，以反映变更。当变更以及测试完成后，进行Merge。

3.2.1.4.3　对于开发计划、配置管理计划发生变更的，项目组人员要按照变更过的开发计划、配置管理计划提交配置项。

**3.2.1.5　变更确认**

3.2.1.5.1　变更后的程序Merge后必需经过测试组进行回归测试，以确保变更没有引入新的Bug。不会引起程序变更的文档（如计划文档）的变更不需经过测试。

3.2.1.5.2　通过Merge后测试后，SQA需对变更进行审核，审核的范围一般涉及以下方面：

1)测试记录；

2)变更请求；

3)配置项的检入及检出；

5) 文件的命名；

6) 版本的编号。

3.2.1.5.3　SQA审核后，开发人员才能生成新的版本，由PL更新到基线库中。

3.2.1.5.4　PL应重新标识所有被影响的配置项及版本。

3.2.1.5.5　A级和B级的变更项也可能直到下次系统发行版本时才生成。

3.2.1.5.6　生成新版本后，CMO负责收集所有变更信息归档，修改变更CIs状态为“正式发布”，关闭变更，并将变更报告发送给发起者。

### 变更控制委员会 (CCB)

**3.2.2.1　CCB成员组成**

CCB成员人数一般为奇数，人数在3～7人范围内。CCB成员一般包括：

1)项目经理PM；

2) 配置管理员CMO；

3) SQA；

4) 测试人员Tester；

5)顾客代表；

6)主要开发人员等。

**3.2.2.3　CCB的决策机制**

寻求CCB成员的一致意见。若不能达成一致，可采取由顾客代表做出决策；或采取少数服从多数的原则，由CCB成员投票确定，投票超过半数即为通过。

## 配置状态统计

### 报告和审计

**3.3.1.1　配置状态报告的目的**

记录和报告整个软件生命周期演化状态。

**3.3.1.2　配置状态报告记录的内容**

配置状态报告记录的内容包括：

2)软件和文档的标识；

3) 目前状态；

4) 基线演化状态；

5)变更状态；

6)版本交付信息等。

**3.3.1.3　配置状态报告的生成**

配置管理报告自第一个基线创建时建立，由配置管理系统生成，及时反映当前配置状态。

**3.3.2　配置审核**

**3.3.2.1　配置审核的类别**

配置审核分为：

1)功能配置审核（Functional Configuration Audit，FCA）：审核软件功能是否与需求一致，并符合基线文档要求；通常要审查测试方法、流程、报告和设计文档等。

2) 物理配置审核（Physical Configuration Audit，PCA）：审核要交付的组成项是否存在，是否包含所有必需的项目，如正确版本的源代码、资源、文档、安装说明等等。

**3.3.2.2　配置审核执行的时机**

通常选择以下几种情况由SQA负责实施配置审核：

1)软件产品交付或是软件产品正式发行前；

2)软件开发的阶段工作结束后；

3)在产品维护工作中，定期地进行。

# 里程碑

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)