

## 1 概述

### 1.1 目的

本文档主要目的在于规范项目配置管理活动，确保配置项正确地唯一标识并且易于存取，保证基线配置项的更改受控，明确基线状态，在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。

### 1.2 适用范围

本文档适用于不同类别的软件产品和软件项目开发工程的配置管理活动，针对项目不同在流程上作适当的删减。配置管理可采用各种工具及手工办法，本文件以CVS（并行版本系统）配置管理工具为例，规定公司的配置管理办法，使用其他工具时也可对应本文件的要求参照执行。

### 1.3 术语和缩略语

#### 1.3.1 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。

#### 1.3.2 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，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。

每个配置项的主要属性有：名称、标签、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。

#### 1.3.3 基线（Baseline）

在配置管理系统中，基线就是一个配置项或一组配置项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，这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，对其修改要严格地按照变更控制的过程进行。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、标签、版本、日期等。

#### 1.4 权限与职责

##### 1.4.1 研发总经理助理

1) 审核变更请求。

##### 1.4.2 项目经理（Project Manager, PM）

- 1) 审核批准配置管理计划；
- 2) 接收或拒绝小范围的变更申请；
- 3) 召集评估变更；
- 4) 提出配置管理的建议和要求；
- 5) 配合配置管理员的工作。

##### 1.4.3 配置管理员（Configuration Management Officer, CMO）

- 1) 编写配置管理计划；
- 2) 执行版本控制和变更控制方案；
- 3) 制定访问控制策略；
- 4) 负责项目的配置管理工作，包括搭建环境、权限分配、配置库的建立、配置项的控制等；
- 5) 配置管理工具的日常管理与维护；
- 6) 配置库的日常操作和维护；
- 7) 负责配置审核并提交报告；
- 8) 根据配置部署表单编译发布版本，并维护版本；
- 9) 对开发人员进行相关的培训；
- 10) 对配置审核中发现的不符合项，拟订纠正措施，要求相关责任

人进行纠正。

11) 监督项目组成员规范的执行情况。

#### 1.4.4 开发人员（Developer）

- 1) 根据确定的配置管理计划和相关规定，提交配置项和基线；
- 2) 负责项目组内部测试；
- 3) 负责软件集成和版本生成；
- 4) 按照软件配置管理工具的使用模型来完成开发任务。

## 2 实施细则

### 2.1 配置项管理

#### 2.1.1 配置项的范围

软件配置可包括以下几方面：开发文档，代码，第三方控件、插件，参考资料，测试文档，用户文档，项目管理文档，验收文档等。

1 项目文档主要指：立项建议书、可行性分析报告、技术建议书、用户需求说明书、项目计划、项目进度计划、项目阶段性计划、产品需求规格说明书、概要设计报告、详细设计、数据库设计、界面设计、用户操作手册、用户安装手册、培训文档、验收报告以及上述文档的评审记录。

1 代码主要指：源代码等。

1 工具主要指：脚本文件、插件、第三方控件等。

#### 2.1.2 配置项基线管理

结合SPP和ISO9000的相关规定，配置管理员根据配置管理规范及配置管理计划，对配置项进行分阶段管理，每一阶段正式评审通过后纳入受控库，作为该项目的的一个基线。

1 项目启动：配置项包括技术建议书、可行性分析报告、用户需求说明书等立项阶段产生的文档，评审或审批通过后建立发布基线。

1 需求阶段：系统调研后开发人员进行需求分析，并整理产品需求规格说明书。产品需求规格说明书经过客户的确认后，建立需求基线。如需升级版本则必须通过评审或审批并得到客户的确认。

1 项目计划：需求分析完成后即可制定项目的开发计划，包括项目计划和主要下属计划。包括项目进度计划、配置管理计划、质量保证计划、测试计划、项目阶段性计划。项目开发计划评审通过后，建立项目计划基线。

1 设计：系统设计可分为概要设计、详细设计、数据库设计、数据库字典、界面设计。针对用户需求规格说明书进行系统设计，配置时应说明系统设计的版本与需求分析报告版本的对应关系。设计说明书评审或审批通过后，建立设计基线。

1 编码（设计实现）：编码按功能模块分子项目，即每个模块记作一个配置项。代码在提交项目组系统测试时建立Beta版本，系统测试产品正式发布后建立Version版本。

1 测试：单元测试和系统测试。单元测试通过提交《单元测试报告》，项目启动后应提交《系统测试计划》，系统测试完成后应提交《系统测试报告》。配置时应说明测试的版本与编码版本的对应关系。系统测试完成后建立测试基线。

1 版本发布：项目组提交《部署表单》，CMO根据部署表单进行编译，发布测试服务器上，并对版本进行维护。同时将发布的版本上传到文档服务器上备份。

1 交付与验收：在交付前配置审核完成后建立产品基线，产品基线包含程序及有关文档配置项，包括交付文档、代码、工具等。

1 产品部署：部署时应包括操作手册、安装维护手册、维护文档以及必要的业务和技术培训文档。

1 相关资料：相关资料也应作为配置项纳入配置管理，此部分包括：

- 1) 相关法律、法规；必须遵照或项目组约定的技术规范；
- 2) 与客户或项目组内部重要的交互信息记录，如会议记录、会谈记录、e-mail和MSN记录等；

## 2.2 版本控制

## 2.2.1 文档的版本控制

所有文档的管理纳入配置管理库，用版本控制工具进行统一管理。文档的版本控制主要通过文档的名称、文档控制页及版本控制工具的标签来实现，主要分为以下几类：

### 2.2.1.1 版本变化型文档

命名方式：[文档名称]+[子系统名称]（可选）

适用文档：项目计划、配置管理计划、质量保证计划、项目进度计划、用户需求规格说明书、产品需求规格说明书、体系结构设计报告、数据库设计报告、详细设计报告、用户操作维护手册、测试用例等。

示例：项目计划.doc

详细设计\_SP门户.doc

标签结构：[大版本] + [子系统简称] + [版本号] + 日期（标签控制说明版本信息）

1 [大版本]： 可选，表示同一项目为不同用户定制的版本。

1 [子系统简称]： 可选，当一个项目有多个子系统时，为区分不同子系统而设置。

1 [版本号]： 采用[V<sub>s</sub>\_x\_y]的形式。

1 日期： 纳入基线管理的日期，用8位表示，如20071031

说明：

a. 文档发布名称采用[文档名+ V<sub>s</sub>\_x\_y]的形式，文档的版本号应该和版本控制工具中相应标签上的版本号一致。

b. 对文档的修改需要从配置管理库中取到本地进行。

c. 对于文档小的修改，如文字错误，格式调整，变更V<sub>s</sub>\_x\_y中的y来区别（如：V1\_0\_1）。

d. 文档内容没有大的增加和删节，意思表述没有发生重大的变化，版本标识通过版本工具中加上x标签来表示（如：V1\_1\_0），以及在文档内部控制页标注变化来表示。

e. 文档有重大增加和删节，意思表述有重大变化的，版本标识通

过在相应文档加上s标签来表示（如：V2\_0\_0）。

f. 对于纳入基线库的文档的修改需要提交变更申请，经批准才能进行修改，并且修改的内容要经再次评审才能重新纳入基线库，作为后续阶段的参考文档。

#### 2.2.1.2 时间区别型文档

命名方式：[文档名称+撰写时间]

适用文档：文档名称有明确的含义，需要用时间标识的日常性文档。如周例会会议纪要，项目月计划，项目月总结,阶段性计划等等。

示 例：周例会会议纪要20030901.doc

#### 2.2.1.3 时间序号型文档

命名方式：[文档名称+人员姓名（拼音）+撰写时间+序列号]

适用文档：测试报告

示例：单元测试报告\_lixiaohong\_20071112\_01.dco

#### 2.2.1.4 其他文档：

对于不能按照前四种类型进行命名的文档

会议纪要：会议纪要YYYYMMDD( )

示 例：9月9日召开的项目启动会

命名为：会议纪要20030909(项目启动).doc

评审报告：评审报告YYYYMMDD( )

同”会议纪要”要求一致。

示 例：10月9日召开的项目总体方案评审

命名为：评审报告20030910(总体方案).doc

#### 2.2.2 发行版本表示

发行版本采用标签说明，结构如下：

[大版本]+[版本类型]+[版本号]+[子系统简称(拼音)]+日期+序号

[大版本]： 可选，表示同一项目为不同用户定制的版本。

[子系统简称]： 可选，当一个项目有多个子系统时，为区分不同子系统而设置。

版本类型：分为3种

Beta表示项目组内部测试，标签：B1\_0\_0-20071015-01

Release系统测试，标签：Release1\_0\_0-SPmenhu-20071112-01

Version正式发行版，标签：Version1\_0\_0-SPmenhu-20071112-01

[版本号] 对于Version正式发行版 是必须要注明的，而其它可选。

发行产品基线在版本号前加Version，如

Version\_1, Version\_2, Version\_3....表示分支；

Version\_1\_0, Version\_1\_1, Version\_1\_2... 表示在分支Version\_1上的标签；

Version\_0\_0, Version\_0\_1, Version\_0\_2... 表示在主线上的标签。

## 2.3 配置库管理

### 2.3.1 配置库的分类

配置库统一由配置管理员负责管理，服务器端使用cvsnt2.0.4，客户端主要使用乌龟CVS。配置库目录结构如下：

### 2.3.2 配置库的建立

所有项目应建立配置库，以便管理各配置项，配置管理员组织建立配置库。程序库主要通过设置版本的分支来实现对配置项权限管理：

1)开发库：开发人员相对比较自由的存储空间，开发人员可以在自己的权限范围内任意取出提交。

2)基线库：配置管理员有最高权限，其余相关人员均为读的权限，发生变更时变更人员须提交变更申请后方可修改基线库内的配置项。

Ø 文档评审通过后，文档严格受控。由配置管理员将通过评审后的文档移植到基线库里同时将该配置项从开发库移除。

Ø 代码一般在移交系统测试时纳入基线库受控，可根据项目的具体情况设置基线。

3)产品库：产品库的产品均出自于基线库，产品库存储的产品用于交付和存档。

配置三库统一由配置管理员管理，根据各开发阶段的实际情况定制

相应的版本选取规则，来保证开发活动的正常运作。在变更发生时，应及时做好基线的推进。

### 2.3.3 分配权限

项目开始后配置管理员编写《配置库目录结构表》明确项目组成员以及相关人员的权限。在wincvs里有三种权限，读（r）、写（w）、添加删除（c）权限。在开发库内，文档部分项目组成员有rcw权限，其他相关人员只r权限；代码部分项目组成员有rcw权限，其他相关人员没有任何权限。在基线库内，项目组成员仅有r权限，其他相关人的权限视情况而定。在产品库内，所有人没有任何权限。配置管理员在三库内均拥有最高权限。

## 2.4 配置变更控制

### 2.4.1 变更的分类

软件及其相关文档的变更按照变更的影响范围进行分类：

1)A级：变更会影响系统级的需求、外部接口、产品价格或者交付期；这类变更必须经过配置管理委员会审核并有客户批准和确认。

2)B级：变更会影响配置项间的功能接口、内部功能的设计、组件；这类变更必须由项目经理或配置管理委员会的批准和认可。

3) C级：变更只会影响配置项内部或对BUG问题的处理；这类变更可以由配置项的管理人员负责批准。

Ø 系统测试前变更控制流程：

Ø 系统测试完毕发布release版本后变更控制流程

图2 变更控制流程

### 2.4.2 变更请求的提出

a. 由技术支撑中心汇集顾客意见，影响到需求变更则填写《配置项变更控制报告》，并提交给配置管理员。

b. 配置管理员对申请表是否清晰、明确和完整性进行审查，若发现变更不明确或不完整，应返回申请者。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。



### 2.4.3 评估变更

a. 配置管理员将《配置项变更控制报告》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。

b. 项目经理对变更进行分解，一般的BUG修正不需要审批直接由项目经理决定是否需要变更。新增功能或对整个项目影响重大的变更必须由研发总助审批通过后方可变更。变更评估文档在完成变更评估后发送给配置管理员。

### 2.4.4 变更实施和确认

a. 变更被批准后，项目经理提交变更实施进度计划，开发人员开始实施变更，并详细记录变更的内容；质量部对变更的实施进行跟踪。

b. 对于代码变更，必须进行回归测试，以确保变更没有引入新的Bug。另外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，进行提交。

c. 通过测试后，质保人员需对变更进行审核，审核的范围一般涉及以下方面：测试记录；变更请求；配置项的检入及检出；文件的命名；版本的编号。

a. 审核后，由配置管理员更新到基线库中。

## 2.5 配置状态报告

### 2.5.1 目的

记录和报告整个软件生命周期演化状态。

### 2.5.2 记录内容

配置状态报告记录的内容包括：

- 1) 软件和文档的标识；
- 2) 目前状态；
- 3) 基线演化状态；
- 4) 变更状态；
- 5) 版本交付信息等。

### 2.5.3 生成报告

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。

## 2.6 配置审核

### 2.6.1 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。

### 2.6.2 执行时机

通常选择以下几种情况由质量保证人员负责实施配置审核：

- 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。

### 2.6.3 不符合项处理

对配置审核中发现的不符合现象，配置管理员进行记录，并交由责任部门限期进行纠正，配置管理员负责纠正措施的验证。所有的不符合项报告均关闭后，才能发布新版本。

## 2.7 发行管理

通过配置审核后，经项目经理批准，由配置管理员负责生产新版本。

### 2.7.1.1 交付管理

这里“交付”是指从配置库中提取配置项，交付给客户或项目外的人员。交付出去的配置项必须有据可查，避免发生混乱。流程如下：

- 1)交付人向质量部申请；
- 2)质量部如果不同意交付，则拒绝交付配置项。如果同意交付，配

置管理员应给出详细的交付清单；

3)交付人验收后签字。