



## 配置管理的精髓

贝尔实验室先进技术研究院 姜海东

随着国内软件业的崛起和成熟，软件配置管理越来越得到重视。可以说，软件业要想更好的发展，没有软件配置管理的支持是不可能的。手工作坊式的软件开发模式将会成为历史，如何把国外成熟的软件配置管理理论和经验消化吸收，进而应用到国内软件开发中就成为国内软件业迫在眉睫的任务了。

软件配置管理是管理和技术相结合的一门学科。应该说，软件配置管理理论难以理解是其难以实践的原因。本文试从基本概念的角度来探讨这门对软件开发具有重要意义的领域。

### 什么是配置管理

在软件开发中，变更是不可避免的。从某种角度上讲，软件开发过程就是一个变更的过程。有些变更是有益的，是具有创造性的，但是，也有些变更是有害的，导致混乱的。正像 James Bach 总结的那样：

我们为变更所困扰，因为代码中的一个极小的混乱可能带来产品的大的故障，但是，他也能够修复大的故障或启用奇妙的新能力。我们为变更所困扰，因为某个喜欢恶作剧的单个开发者可能破坏掉项目，但是，一些奇妙的思想也源自那些喜欢恶作剧的人员。

因此，如何管理这些变更是一个软件开发能否成功的关键。简言之，软件配置管理就是管理变更的过程，它贯穿着几乎软件的整个生命周期。成功的配置管理系统可以提高产品的质量、项目开发效率，而且最大限度的减少对个别“英雄”式人员的依赖。

尽管配置管理（Configuration Management）这个概念被提出有几十年了，但是，业内还没有一个全面而权威的定义。Configuration 的意思是“使成形”，它来源于拉丁语的 com-（表示“与”或者“一起”）和 figurate（形成）。它还有一个意思是“组成部件或元素的相对排列”。因此，配置管理（Configuration Management）指的是管理组成部件或者元素的相对排列。

配置管理的概念来自于硬件领域，美国国防部最早使用了配置管理的概念。我们知道一架飞机的构成非常复杂，比如机头、机身、机翼和机尾等。不同型号飞机的各个部分是不能随便组装的。因此，我们只有把相匹配的部件组装在一起，才能构成了一个功能完备的飞机整体。随着技术的提高，各个部件可能还要进行功能改善，我们还要使得不同版本的部件能够正确无误组合在一起。

准确地说：

配置管理是对产品进行标识、存储和控制，以维护其完整性、可追溯性以及正确性的学科。

从上面的描述，我们知道，配置管理的基本单位是配置项。软件配置项可以是：

- 与合同、过程、计划和产品有关的文档和数据
- 源代码、目标代码和可执行代码
- 相关产品，包括软件工具、库内的可复用软件、外购软件及用户提供的软件

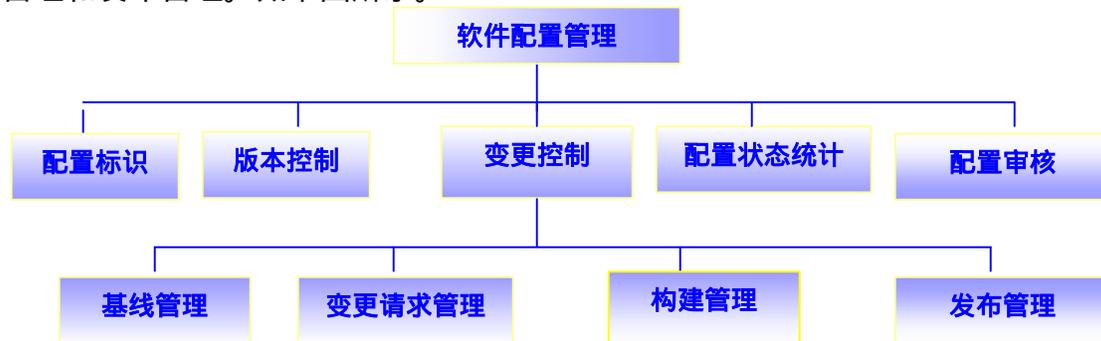
从“哲学”意义上讲，配置管理记录配置项的三个方面：

- 从哪里来？此项可归结为 WWW 的问题，（Who）谁创建的？（When）什么时间创建的？（Why）为什么创建此配置项？
- 当前在哪里？此项纪录配置项当前的存储位置以及状态。
- 将到哪里去？通过配置控制来把配置项“组装”到正确的版本中去。

配置项可以是粗粒度的，也可以是小粒度的。如果跟踪个别需求，那么不必要把整个需求规格说明文档定义为一个配置项，可以把每个需求定义为配置项；如果把软件开发工具也放入配置管理系统，那么把配置项定义为文件级就不合适了，只需要跟踪开发工具的版本，即把整个配置工具定义为一个配置项就足够了。

简而言之，配置项可以是文件级粒度的，也可以使文件版本级粒度的。当然，粒度越小管理的成本越高，但是配置的精度也就越高。

一个完整的 SCM 系统要具有三个核心功能：配置标识、版本控制、变更控制、配置状态统计和配置审核。其中变更控制包括基线管理、变更请求管理、构建管理和发布管理。如下图所示。



下面，我们来具体理解这些概念。

## 配置标识

配置标识就是识别产品的结构、产品的构件及其类型，为其分配唯一的标识符，也就是说，每一个配置项要有一个唯一标识。一般说来，标识包括两个方

面：一是文件名，二是版本，可用如下一个二元组来标识：<文件名，版本>。每个项目首先要确定一套命名规则，例如，采用“系统.子系统.模块.文件”的方式，</videoConference/audio/compressing/main.c , 2.1>就是一个唯一标识。

## 版本控制

版本控制就是对在软件开发过程中所创建的配置对象的不同版本进行管理，保证任何时候都能取到正确的版本以及版本的组合。

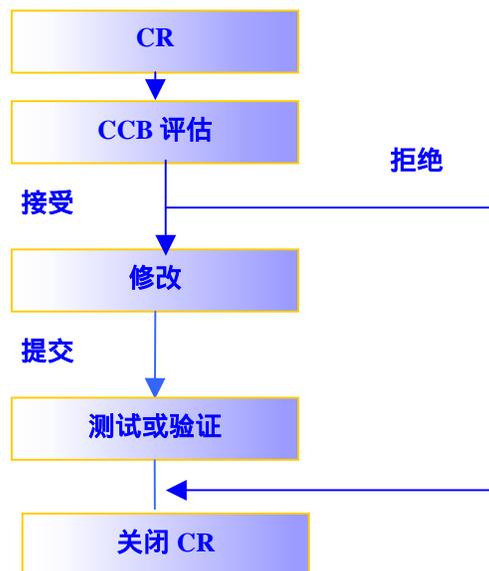
当前，这方面典型的工具有如 VSS 和 CVS。

## 变更控制

在软件开发过程，要产生许多变更，比如，配置项、配置、基线、构建的版本、发布版本等。对于所有的变更，都要有一个控制机制，以保证所有变更都是可控的、可跟踪的、可重现的。

对变更进行控制的机构称为变更控制委员会（Change Control Board，简称 CCB）。变更控制委员会要定期召开会议，对近期所产生的变更请求进行分析、整理，并做出决定。而且要遵循一定的变更机制。

下面是一个典型的变更机制：



## 变更请求管理

变更请求管理就是对变更请求（Change Request，简称 CR）进行分类、追踪和管理的过程来实现的。

变更的起源有两种：功能变更和缺陷修补（Bug-Fix）。功能变更是为了增加或者删除某些功能。缺陷修补则是对已存在的缺陷进行修补。

对变更请求的有效管理可以提高产品管理的透明度，经理可以清楚的知道当前产品的进展情况，比如有多少个新产生的 CR，已经解决了多少 CR 等等，有利于经理做出正确的决策。

### 基线管理

基线是指经过正式评审和批准，可作为下一步工作的基准的一个配置。软件开发过程中，无论是需求分析、设计、测试都需要在完成时建立基线，以作为下一步工作的基础。

通过基线管理可以使用户能够通过对适当版本的选择来组成特定属性（配置）的软件系统，这种灵活的“组装”策略使得配置管理系统象搭积木似的使用已有的积木（版本）组装成各种各样、不同功能的模型。

基线的变更需要一个严格的流程，需要提出申请，经过审批，然后才能进行。

### 构建管理

在做构建时，我们需要首先取出正确的配置，然后再做构建。我们可以利用基线，可以取出某个基线的所有配置项，也可以利用配置管理系统的构建功能直接在工作空间内做构建。

构建管理需要配置管理工具的支持。

### 发布管理

软件产品的每个版本都是一组配置项（源代码、文档、数据）的集合。举个例子来说，我们要发布软件的 3.2.6 版本，那么我们就要把源代码、文档、数据中所有应该包含到这个版本中的正确配置项检出。

所以如何管理每个版本中包含哪些配置项是非常重要的。

### 状态报告

状态报告要回答所谓 4W 的问题：

What：发生了什么事？

Who：谁做的此事？

When：此事是什么时候发生的？

Why：为什么做此事？

状态报告要能够报告所有配置项以及变更请求的状态，通过量化的数据和报表反映项目开发进度的状态。

### 配置审核



配置审核要审查整个配置管理过程是否符合规范，配置项是否与需求一致，记录正确，配置的组成是否具有一致性等等。比如，需求分析文档提交后，需要由一个由相关人组成的小组进行正式评审，只有通过了评审才能基线化。对于源代码也一样，一般说来，每行代码都要进行评审（Review），只有通过评审才能交由测试人员进行测试。

## 实施配置管理的好处

我们知道软件有三个要素：时间、预算和质量。一个成功的软件就是要在限定的时间内，不超过预算，交付符合质量要求的产品。真正实施配置管理后，我们会对产品的开发过程进行有效的控制，可以加快开发进度，降低开发成本，保证产品的质量。

### 产品经理可以得到什么好处呢？

- 准确掌握项目的开发进度。配置管理系统可以提供详尽的状态报告，例如当前系统有多少个 Bug，所有 Bug 的状态如何？已经解决了多少 Bug？
- 了解项目组成员的工作负荷、工作效率以及工作质量。例如，我们可以知道当前分配给每个成员的工作量，每个成员已完成的工作量，每个成员未通过正式评审的工作比例等等。
- 减少人员流动所带来的影响。每个成员的所有变更，包括文档、代码的增删都是可追踪的，而且对于变更的原因、描述也都有记录。这样，一旦成员离开，其它成员就可以在最短的时间里接手。
- 有效提高过程管理，配置管理产生的许多数据可作为管理者度量项目的依据。

### 开发人员和测试人员可以得到什么好处呢？

- 提交的代码被有效保存，开发人员再也不用花费精力去保存各个版本了。
- 提高团队的协作效率。开发人员之间以及开发人员和测试人员之间可以有有效的沟通，大家都互相知道其它人的工作状态。
- 提高修复缺陷的效率。可以依据 Bug 发现的版本，迅速重建环境，重现 Bug，快速定位代码，找出根源。
- 职责清楚，任务明确。每一步的工作都是基于某一基线的，比如，设计文档是依据基线化了的需求分析文档，这样一旦出现问题，就可以找出问题出在什么地方。

当然，实施配置管理的好处远不止这些。软件配置管理作为软件开发的基石，它提供了一个协作开发的环境，只有大家共同遵守配置管理规范，互相协作才能保证项目的成功。

## 结束语



配置管理本身无论从理论和实践都在不断丰富和发展。例如，配置管理应用于“知识库”的管理就产生了“内容管理”这一新的领域。配置管理提供的状态报告和数据统计也为软件度量提供了决策依据。配置管理为项目管理提供了各种监控项目进展的视角，为项目经理确切掌握项目进程提供了保证。配置管理也为开发人员提供了一个协作的平台，在此平台上，大家能够更有效率的交流和协作。可以说，配置管理是软件开发的基石！

配置管理近年来在中国得到了极大的认可，可以毫不夸张的说，没有配置管理，就谈不上软件开发，就谈不上软件质量，就谈不上软件业的发展。随着软件业规模的扩大，配置管理的实施不是要不要的问题，而是什么时候、如何实施的问题了。

参考文献：

Babich, W.A., Software Configuration Management, Addison-Wesley, 1986.

Peter H. Feiler, Configuration Management Models in Commercial Environment, CMU/SEI-91-TR-7, 1991.4

作者简介：2000年在北方交通大学获工学硕士学位。现供职于朗讯科技（中国）有限公司贝尔实验室先进技术研究院，从事配置管理系统的研究和开发，有四年多的配置管理领域开发、咨询、培训经验。