BOOKS FOR PROFESSIONALS BY PROFESSIONALS™

# Real World Software Configuration Management

涵盖 Windows .NET 和 Linux

# 软件配置管理

- 提供了数量众多的提示信息和最优方法，既适用于初学者，也适用于经验丰富的软件配置管理员
- 使用真实的 Windows .NET 和 Linux 示例来说明基本的软件配置管理理论
- 详细介绍最新的源代码控制工具，如 SourceSafe 和 CVS 等

（美）Sean Kenefick 著
王海涛 沈火林 译

清华大学出版社

# 软件配置管理

(美) Sean Kenefick 著

王海涛 沈火林 译

清华大学出版社

北 京

# 内 容 简 介

软件配置管理(SCM)对每一个软件项目都非常重要。而大部分公司对 SCM 还没有引起足够的重视，许多人对它还不够了解。

本书分为 3 个部分——“角色”、“工具”和“任务”，介绍了 SCM 角色、源代码控制工具，以及维护源代码数据库和构建产品的方法等。同时还通过 Windows .NET 和 Linux 的示例进行说明。读者可通过这些知识的学习而更深入地了解 SCM。

本书非常适合工作在基础软件工程机构中的人员。

EISBN：1-59059-065-1
Real World Software Configuration Management
Sean Kenefick
Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA.


北京市版权局著作权合同登记号 图字：01-2003-8758



**图书在版编目(CIP)数据**

软件配置管理/(美)凯尼菲克(Kenefick, S.) 著 王海涛，沈火林 译. —北京：清华大学出版社，2004 · 9

书名原文：Real World Software Configuration Management

ISBN 7-302-09093-9

I. 软… II. ①凯… ②王… ③沈… III. 软件—配置—管理 IV. TP31

中国版本图书馆 CIP 数据核字(2004)第 072233 号

**出 版 者**：清华大学出版社 **地 址**：北京清华大学学研大厦
http://www.tup.com.cn **邮 编**：100084
**社 总 机**：010-62770175 **客户服务**：010-62776969

**组稿编辑**：曹 康
**文稿编辑**：徐燕华
**封面设计**：康 博
**版式设计**：康 博
**印 刷 者**：国防工业出版社印刷厂
**装 订 者**：三河市金元装订厂
**发 行 者**：新华书店总店北京发行所
**开 本**：185×260 **印张**：19.75 **字数**：493 千字
**版 次**：2004 年 9 月第 1 版 2004 年 9 月第1 次印刷
**书 号**：ISBN 7-302-09093-9/TP · 6417
**印 数**：1～4000
**定 价**：39.80 元

# 前　言

在 Internet 还处于初期阶段时，组件对象模型(COM)还没有出现之前，我曾负责几个包装图像解决方案的安装软件的开发。也就在那时，我学会了欣赏和尊重软件配置艺术；因为我的工作是最后一道工序，因此我总是落后于时间进度。现在，很清楚，我又落后了。因为软件差不多到我手上总是很迟，这意味着在白板上，我的时间进度只有被压缩才能满足原本已经压缩的包装进度(由全体职员的生产线，填充内容和包装组成)。因此，在我的安装开发进度表开始时，办公室中的开发人员、管理人员和首席技术官(CTO)排成了队，转来转去，为我带来各式美味、水和咖啡，就好像我已经完成了工作一样。

即使是简单的安装程序也非常关键，因为它是客户对软件的第一感受。它将使软件中断，或者使软件成功运行。是的，它很重要，如果您曾经编写发布软件的安装程序，就会知道其中有许多错综复杂的细节，即使是最好的开发人员都会为此而发疯！伴随软件配置的职责有许多压力，并且这种压力会在引进其他的软件管理概念之后逐渐增加，这些概念包括源代码控制、版本控制和自动产品构建等。当技术变得复杂时，这些概念可能会更加复杂。

很高兴，我很幸运地告别了安装的日子，在这个过程中，我们将 COM 组件引入到进展中的图像解决方案中。非常及时！在早期的 COM(或者 OLE Automation 1.0)中，版本控制和部署中的问题令人头痛，几乎难以克服，特别在您不得不处理一些 16 位应用程序遗留的问题并执行 32 位组件的转换时。

最后，我抛弃了图像和动画，改为设计文档管理解决方案。此时我遇到了本书的作者，Sean Kenefick，从而知道一个职业的配置管理专家如何将工作做好。我的文档管理解决方案是一个典型的客户机/服务器企业应用程序，用分布式 COM 体系结构(DCOM)进行构造，它支持私有的和第三方数据库，包括工作流的电子邮件服务器配置。Sean 决定要使工作成为一种享受，他开始用一个桌面快捷图标进行自动构建、版本控制和部署过程。

就是这样。

我们每夜的构建将被自动化，这种专门的构建机器将有一个桌面图标，它将从 Visual Source Safe 中检出最新的源代码，用一个适当安排构建次序的脚本构建所有组件，以此来处理组件的相关性，将自动增加的版本号分配给所有的构建输出，用该版本的标记为源代码安全目录贴上标签，并且用生成的 CD 图像创建版本的目录。对此，我很快就深感佩服。我以前的经验告诉我，源代码控制就是每夜向磁带进行备份，并且开发人员要构建和测试他们自己的组件。

从业务的角度看，这种自动化过程非常关键。实施源代码控制以确保公司拥有所有开发人员最后的源代码。该源代码的自动构建过程可确保将正确的源代码用于构建组件，并且使开发人员遵守纪律，避免输入部分完成的工作而中断构建过程。版本控制为我们提供历史足迹，在构建过程中中断功能时可以允许用户查看历史记录，也可以为开发和质量保证(QA)人员记录构

建中所发现的缺陷。这种自动构建过程完全改进了用最新构建更新开发、QA 服务器的能力。

在一个开发环境的后台工作中，核心的任务仅仅是必须具有进程和自动化。您必须使用源代码控制来保护公司的代码资产，并提供版本控制和回退功能。您必须有改进的构建过程，它将根据源代码构建组件并组织分布式组件和其他部署内容。您也必须有值得依赖的自动化部署过程，精确地更新部署的目标、CD 安装、Web 下载，或者是将更新部署到驻留环境的脚本。

10 年来，我们已经走过了很长一段路。我过去常常构建软盘安装程序，它要求每张盘都“填充”接近 1.44MB 的信息，目的是尽量减少所使用的磁盘总数。实际上，即使是利用工具，我也需要手动选取文件才能完成这项工作！而在今天，人们可从 Web 上下载整个 CD 内容。此外，尽管这种传统的 CD 安装程序仍然存在，而且用起来也不错，但是还有更多值得考虑的事情，诸如 Web 部署，通过因特网自动化版本更新，以及应用程序驻留等。

今天，配置管理员、开发团队和管理人员必须协同设计应用程序的部署模型。应用程序版本控制不再像组件标记或源代码控制目录那样简单。因此，新的发布渠道、组件体系结构以及版本更新技术使应用程序开发人员的参与变得非常重要，这包括帮助配置管理员形成正确的软件体系结构，完成部署计划，而不需要对自动更新。同样，在宿主环境中工作时，开发人员和信息技术(IT)人员必须联合配置管理人员以构建高效过程，这将使公司在更新产品场所时仍然能维持服务水平不变。

这些概念都是跨平台的。在作为首席信息官(Chief Information Officer，CIO)负责构建于 J2EE 上的 24×7 的驻留操作期间，我有幸再一次与 Sean 工作。这一次，他的工作是为配置管理构建一个完整的过程，该过程负责整个代码的流水线部署，从开发到质量保证(QA)，再到最终的生产。当您在一个 24×7 的环境中工作并最终负责产品更新的安全部署时，您就能意识到这一切是多么重要，既要能够回滚变更，部署精确的更新，又要有很好的文档处理和配置步骤。如果丢失了一个步骤或部署发生问题，我们将不得不立即重新构建一个产品机。我不敢想象这样的工作除了委托给 Sean 外还能委托给谁。根据我多年的经验，他是我所知道的惟一对此有激情，并真正完美了这种艺术形式的人。

配置管理与具体实现是一个类似的过程。在大型组织中，属于配置管理的许多任务散布在几个人或部门之中。Sean 的这本书关注 SCM 人员的核心任务，但他也将大量的过程和管理实际经验放入到每一章的内容之中。他将教会您如何自动构建脚本，如何在 Web 上配置应用程序，以及如何利用一些容易得到的常用多平台工具构建传统的安装应用程序。更重要的是，他将鼓励您为完善这种艺术而骄傲，这是成功进行软件开发的核心。就此主题，您不可能找到一本比本书更好的信息源，或者不可能找到一位比他更有激情的作者，希望与您共享他丰富的经验！

Michèle Leroux Bustamante
Associate, IDesign
http://www.idesign.net

# 序　言

在一个竞争日趋激烈、要求日益苛刻的市场上，将技术人才和工具聚集起来构成解决方案时，您会发现时间越来越紧，资金逐渐不够用，解决方案设计团队经常会想用并不正规的程序来尽快交付解决方案。其实这样做并不那么容易，速度未必更快，花费也未必更少——而且因为对基础工作缺乏足够的关注，可能会使一个非常酷的解决方案立即变得一无是处。

不只是为了遵循笨拙的程序和冗长乏味的 ISO 标准，但是知道一些基础构造过程并能做到深谋远虑，这一点是必不可少的，Sean Kenefick 的方法就不错。该方法具有服务于绝大多数开发团队的平衡能力和灵活性。如果有足够时间来等待其他的团队成员修复无法正常运行的软件，或者细心地对新上手的程序员进行有关源程序控制规则的培训，那么将不需要自动化的工具或者程序来保护彼此不受侵害。

设想一组世界级的跑步运动员在划有界限的跑道上进行冲刺。为了赢得胜利，他们必须集中注意力不超出自己的跑道，又要使出浑身的气力保持最高的速度。他们知道：超出自己的跑道将会被取消比赛成绩，因此他们必须始终只能在自己的跑道上前进。可能是太过于注意安全——其中有两个人彼此绊了一跤，其中一人以差不多每小时 20 英里的速度摔了出去，此时身体受伤已经不可避免。必须既要保证运动员向前奔跑，同时又要保证他们的人身安全——避免他们彼此之间受到伤害。

软件配置管理(SCM)及其工具和程序可帮助开发人员集中注意力，并使他们的处理器发挥最大的效率。如果要保证运动员在奔跑时不会抢跑道，需要许多人来监视，而一个良好的 SCM 实际上就可以解决所有问题——但是必须通过团队合作。如果一个团队一开始就视 SCM 为多余，那么该团队就可能会因为短期的成功而失去美好的未来。一个“跨线”的开发人员，都可能会让其他人摔一跤。如果这种情况继续下去，这个开发团队就不可能赢得竞赛。

软件开发与技术过程很类似。记得曾经有一位开发人员走进我的办公室，将门关上，然后很严肃地建议将某个项目的整个测试团队解散。“他们总是与我过不去，”他说，“我做的一切在他们看来都不对。”有经验的开发人员对此都会置之一笑。我们不应该自以为是，正确的做法是赶紧修正错误。

还有一件事，有 3 个开发人员以绝对惊人的速度推出了许多强大的功能。这 3 个人理解软件开发的过程，互相测试了彼此的工作，并且通过纯粹的意志力维持原则，保持专注，通力合作并以最快的速度解决争端。他们解决冲突的策略是呆在同一房间中，协调每一件悬而未决的事——对他们来说，这条原则还比较管用。因为作为一个长期合作的团队，他们已经协调一致，完全可以完成彼此的语句。但这种情况非常少见——以致需要长期存在的 SCM 来减少诸多不足，如彼此缺乏协作、难以控制项目开发的步调、开发团队的规模以及性格的冲突等，或者难以满足长期存在的，在某一时间段需要增加、减少或更换技术专家的需要。

过程的一致性同时也能使您的开发人员从处理软件的许多复杂工作中解脱出来，如软件版本控制、构建、部署和软件交付过程中出现的大量管理问题等。曾经有一位开发人员对我说，他感到使用软件版本控制有很大的限制。如果除了他之外再没有其他人为此工作，为什么需要将许多内容检入中央仓库呢？管理专家理解这样做的必要性，但是技术人员—— 特别是缺乏经验者—— 却无法理解。如果整合了一个软件开发团队，并且没有适当的基础结构，您将会很奇怪：居然有这么多开发人员希望加入您的团队。然后，更让您惊诧的是，他们几乎不能做什么事。

SCM 也能避免团队成员中的游手好闲行为。“积极的游手好闲”容易发现和控制，因为表现很明显，而且凡事都停留在口头上。“消极的游手好闲”在开发队伍中可能相当多，当进度吃紧时，他们就会牺牲产品的质量。对待这种情况，您需要的是一个合适的过程，它可以使您的开发团队不会因为受到快速交付的影响而忽略隐含的错误——这些错误最终会随着时间的推移而变成难以修复的问题——或者在交付后，用户也会遇到该问题。例如，在本地机器的数据库中增加一个新的表，但从来没有将该表送到中央服务器中？为了解决某个问题，您升级到 Widgets 4.0，但是其他的任何人仍然使用 Widgets 3.0？在长长的周末度假前，您检入了一堆软件，但是没有花费时间去查看它是否完全编译好？

在绝大多数失败的软件项目中——至少在资金和人员配备充足的项目中——失败归咎于无法保证所有的开发人员尽职守责，以及彼此不能相互配合。SCM 担负设计基础结构的重大责任，但是这需要项目领导人员的大力支持。

关于如何处理棘手的管理问题以及如何帮助人们循规蹈矩，Sean 进行了极好的归纳。理想情况下，SCM 可以是一个管理者的角色、技术员的角色、鼓舞人心的领导者角色、外交官的角色和管理办公室的角色，并且必须永远都有自律能力且不乏幽默感。如果您的 SCM 态度粗暴、反应迟钝、言语刻薄，或者存在任何不能与各层的人员积极热情沟通的障碍，您将面临问题的严重性将超出没有 SCM 时所产生的问题。Sean 描述了 SCM 需要有的灵活性和亲和力，而且在即将读到的内容中，您将发现 SCM 的个性呼之欲出。

不管是一位管理人员、业务分析专家、技术专家、测试人员，还是从事软件开发项目中的任何一种职业，您都将看到一些有关控制简单过程和基础结构能力的至理名言，扬起您的风帆并将一些有益的内容应用到您的项目开发周期中去。对此，谁不想拥有更多呢？

David Birmingham

CEO, Virtual Machine Intelligence

# 引 言

除非为吝啬鬼工作，否则当公司取得成功时，每个员工都会是赢家。此时，公司股票将突然升值，员工的工资也以看得见的速度上升。公司也能为员工提供更好的福利。很明显，希望公司成功是您最大的期望所在。

正因为如此，构建产品并将它们迅速推向市场是在公司中最重要的工作。但是，对于发布一个产品，有许多重要的工作远比拿出艺术性的包装重要。必须有人确保软件的正确版本被复制到光盘上，或者被部署到 Web 上；然后必须确保客户不会感染上 Melissa 病毒。

但是更重要的是，在消费者看到包装盒之前，这个人同样必须查看产品的构建代码块，确保软件在放置到最终的光盘之前没有任何错误。这个人是团队中的守门员：在每一个角落都可能隐藏影响公司成功的隐患，他必须阻止无意中的错误。

肩负如此重要责任的人是谁？他们就是合格的软件配置管理员(SCM)。

## 软件配置管理员

很奇怪的是，即使是最高级的程序员也可能会热心于这个职位。怀着最美好的意愿，时常从该工作中学习，这些学徒必须确保程序员的输出永远可用，并且要确保可以正确构建和安装产品。

令人不解的是，许多公司竟然将这种极度重要的工作交给最没有经验的人去做。但在软件工程快速发展的世界中，这些管理人员必须满足许多人认为是不可能的交货期，而他们监管的却是一个非常自负的开发队伍——试图省时省力，他们一般可以被原谅。尤其是现在，我们已经进入了一个新的世纪，并且软件工程世界的所有范例都已经发生了改变。

许多公司都不再盲目花费金钱和精力。在这些公司中，那时管理员常常要花费数月的时间寻找合格的候选人，结果却徒劳无功，甚至当他们偶尔遇见一位合格人选时愿意为他设立合适的职位，而现在，一般会让公司内部人员来从事该项工作。抱有“要多少就给多少”的软件开发理论并且愚昧花钱的公司已经很难找到。在今天的市场上，只有精打细算的公司才能生存。

了解内情的公司比较钟情于高级SCM(也称为配置经理)，尽管面对的可能是最不景气的求职市场，但是优秀的SCM依然可能每月有两到三次接到猎头公司的聘用电话。

然而，还有一些公司似乎对此并不了解。在这些公司中，低级的雇员被委以确保所有同事尽职工作的重任。

## SCM 的当前职责

在软件工程界开始工作时，我为一家规模不大的生产视频卡驱动程序的公司工作，该公司位于加利福尼亚的 San Rafael。作为最低级雇员，我立即被编排到编程队伍中，SCM 基本的工作摆在了我的面前：管理源代码数据库、构建产品并创建配置安装程序。

一年以后，我在圣地亚哥谋到一份新的职位：担任一家产品开发机构的高级 SCM。他们选中我的原因更多是看重我的编程背景，而不是我的 SCM 能力。我对接受这份工作很是小心——毕竟在许多公司中，配置管理员(与质量保证和技术支持人员一起)被看作是二等公民。同时，我也担心此职位对我没有挑战性。

那就是我在圣地亚哥的工作，我接受了。

但是我决不后悔。我想在这家给了我机会的公司中，我所赢得的宝贵经验是继续从事主流编程工作永远也无法得到的。我发现自己所处的职位并不是那么容易胜任。此外，我发现自己常常要对技术专家进行挑战，如果我继续呆在 C++程序员位置上，那就根本不可能有这种机会。

在这个市场上，至少在加利福尼亚这儿，许多程序员徘徊在人行道上，寻求有限的职位。但是 SCM 仍然供不应求。最近，一位招聘人员给我打电话，让我考虑一下他手头的两个空闲职位。因为我对现在的职位很满意，于是委婉地回绝了他。“那么，您能否为我们推荐一位与您有同样经验的朋友呢？我想您一定有这样的朋友。”他说。很不幸，我帮不了他。实际上，当我是一位经理时，为了寻找一位胜任的 SCM 候选人，我花了 8 个月的时间进行面试——最终发现仅有一人完全胜任该项工作。然而，他拒绝了我的加盟要求，原因是他已经被一家有竞争力的公司以更优厚的待遇聘用了。

二等公民？根本不是这回事！聪明的公司理解强有力的 SCM 的必要性。而且，这种需要将永远不会过时。

## 写作本书的原因

我决定写一本书，它能帮助一些像我这样勇于投身到完全陌生的角色中的人们。他们当中的许多人可能还没有正规的编程经历，而且更糟糕的是，没有人可以对他们进行引导或者启发。他们的所有能力差不多都来自于从这项工作中不断学习。我希望这本书能对他们的工作有所帮助。

### 本书读者对象

在开始担任 SCM 角色时，我搜寻了一些书，希望它们能有助于我理解新的职位；然而，我却发现许多书中都只是覆盖该工作的极小一部分内容，而且没有一本书能涉及我将必须掌握的基本任务和工具。时间在不断地流逝，有关工作中新兴的 SCM 的基本资料依旧缺乏。

因为SCM的职位可能不仅仅是强加给程序员，而且也可能是质量保证或技术支持人员。这本书的设计理念是“从零开始”。它描述了该职位各方面的基本知识，以及可以帮助您避免典型的新手失误的策略。比较有经验的SCM可能发现本书将围绕他们的许多技能展开，并且详述他们可能需要结合到过程之中的特定任务。

本书非常适合在基础软件工程机构中工作的人员。然而，可能您正在构建关键的医疗和工程软件，也可能正在编写过山车的功能或者正将软件集成到一个飞机的驾驶员座舱中。如果是

这样，本书将给您一个适应工作的极好基础——但是也有一些您必须遵循的规则和原则不在本书范围之内。对此，您需要查找适合特定应用的参考材料。

## 本书的结构

本书分为 3 个部分，每一部分又由几章组成。

第 I 部分——角色，这是关于 SCM 角色的基本介绍。在这一部分中，您将找到对角色本身的描述，以及有关源代码控制和用于实现它的工具的基本信息：

- 第 1 章：初步了解 SCM 角色
- 第 2 章：SCM 和软件开发过程
- 第 3 章：源代码

第 II 部分——工具，包括许多源代码控制工具概述，并且详细描述了常见的免费软件工具：

- 第 4 章：源代码控制工具
- 第 5 章：CVS
- 第 6 章：SourceSafe

第III部分——任务，简述 SCM 工作的部分职责。这些章节将介绍维护源代码数据库以及构建产品的方法。此外，这部分还讨论了针对不同操作系统和编译工具的特殊策略。从中，您将会找到有关 Web 部署和产品发布的信息：

- 第 7 章：SCM 实验室
- 第 8 章：基本的构建
- 第 9 章：Windows .NET 的构建
- 第 10 章：安装
- 第 11 章：部署和构建的再思考

# 目 录

## 第Ⅰ部分 角 色

# 第Ⅱ部分 工 具

## 第III部分 任 务