

# 《配置管理最佳实践》

## 图书基本信息

书名：《配置管理最佳实践》

13位ISBN编号：9787115321909

10位ISBN编号：7115321906

出版时间：2013-9-1

出版社：人民邮电出版社

作者：[美]Bob Aiello,Leslie Sachs

页数：191

译者：顾刘学,苏建雷,魏朝晖

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

# 《配置管理最佳实践》

## 内容概要

《配置管理最佳实践》贴近实际，旨在指导配置管理从业者如何处理日常工作中需要面对的各种复杂情况。全书详细介绍了配置管理的6个核心职能：源代码管理、构建工程、环境配置、变更控制、发布工程和部署。作者在书中展示了如何实施配置管理，从而可以支持软件和系统的开发，满足SOX、SAS-70等合规准则的要求，提前考虑新兴的IEEE/ISO 12207等标准，同时还可以和最新的ITIL、COBIT和CMMI等框架集成到一起。

《配置管理最佳实践》对于任何与配置管理相关的工作人员来说都是一本必不可少的参考书。从CTO到CIO，再到开发人员、质量保证工程师、项目经理、软件工程师、系统分析员、测试人员和合规专业人士，皆是如此。

# 《配置管理最佳实践》

## 作者简介

Bob Aiello:

CM Crossroads主编和软件流程改进（包括软件配置管理和发布管理）咨询师。在纽约市顶尖的金融服务公司作为技术经理负责分布全球的配置管理工作，在这里他服务了25年。他是IEEE 828标准工作组在配置计划方面的副主席，同时也是IEEE软件与系统工程标准工作组管理层的一员。

Leslie Sachs :

Yellow Spider Inc.的COO和联合创始人，同时也是CM Crossroads的助理编辑。

## 书籍目录

第I部分 配置管理核心实践	1
第1章 源代码管理	3
术语和源代码管理	4
源代码管理的目标	5
源代码管理的原则	5
1.1 为什么源代码管理如此重要	6
1.2 从哪里开始	7
1.3 源代码管理核心概念	8
1.3.1 建立基线和时间机器	8
1.3.2 保留与非保留签出	9
1.3.3 沙箱和工作空间	10
1.3.4 变体管理	10
1.3.5 复制分支与增量分支	11
1.3.6 如何处理缺陷修复	11
1.3.7 流	12
1.3.8 合并	13
1.3.9 变更集	14
1.4 权限和需求跟踪	14
1.5 管理全球分布式开发团队	15
1.6 工具的选择	16
1.6.1 开源软件与商业软件	17
1.6.2 产品成熟度和供应商承诺	18
1.6.3 可扩展性和开放的API	18
1.6.4 不要过度工程化源代码管理	19
1.7 认识质量成本和总拥有成本	19
1.8 培训	20
1.9 建立使用模型	21
1.10 实施时间和风险	22
1.11 建立支持过程	22
1.12 高级特性和授权高级用户	23
结论	23
第2章 构建工程	25
构建工程的目标	26
构建工程的原则	26
2.1 为什么构建工程如此重要	27
2.2 从哪里开始	27
2.3 构建工程的核心概念	28
2.3.1 版本ID和标记可执行文件	28
2.3.2 不可变的版本ID	28
2.3.3 打上版本标记或者标签	28
2.3.4 管理编译依赖	29
2.3.5 独立构建	29
2.4 建立构建职能的注意事项	30
2.4.1 推广独立构建	30
2.4.2 过度工程化构建	30
2.4.3 保持正直和诚实	31
2.4.4 隶属研发部门引起的利益冲突	32

2.4.5 组织结构的选择	32
2.5 构建工具评估和选择	33
2.5.1 Apache Ant进入构建舞台	33
2.5.2 Maven	34
2.5.3 Maven与Ant	34
2.5.4 使用Ant生成复杂构建	34
2.5.5 持续集成	35
2.5.6 持续集成系统	35
2.5.7 集成开发环境	36
2.5.8 静态代码分析	36
2.5.9 构建框架	36
2.5.10 构建工具的选择	36
2.5.11 对比优缺点达成一致	37
2.6 质量和培训成本	37
2.7 把构建做得更好	37
2.7.1 鲍勃的构建秘方	38
2.7.2 测试驱动的构建	38
2.7.3 信任但仍要核查	38
2.7.4 飞机的驾驶舱	38
2.8 构建工程师的角色	39
2.8.1 了解构建的项目	39
2.8.2 与开发人员合作	40
2.8.3 招募新人	40
2.9 架构是构建的基础	40
2.10 建立构建过程	41
2.11 持续集成与每日构建	41
2.12 构建工程的前景	42
结论	42
第3章 环境配置	43
环境配置控制的目标	44
环境配置控制的原则	44
3.1 为什么环境配置如此重要	45
3.2 从哪里着手	45
3.3 支持代码提升	45
3.4 管理配置	46
3.4.1 使用的是哪个数据库	46
3.4.2 那笔交易发生了吗	46
3.4.3 少用几个符号	47
3.4.4 集中分配环境变量	48
3.5 建立配置管理数据库的实际方法	48
3.5.1 识别和控制	48
3.5.2 理解环境配置	49
3.6 依赖于环境配置的变更控制	49
3.7 减少控制	49
3.8 管理环境	50
3.9 环境配置的未来	50
结论	51
第4章 变更控制	53
变更控制的目标	54

- 变更控制的原则 54
- 4.1 变更控制为何如此重要 54
- 4.2 变更控制从何做起 55
- 4.3 变更控制的七种类型 55
  - 4.3.1 优先级 55
  - 4.3.2 把关控制 56
  - 4.3.3 配置控制 56
  - 4.3.4 变更咨询委员会 57
  - 4.3.5 紧急变更控制 57
  - 4.3.6 过程工程 57
  - 4.3.7 高级管理人员监督 57
- 4.4 建立变更控制 58
- 4.5 变更控制实例 58
  - 4.5.1 29分钟变更控制会议 59
  - 4.5.2 投资银行变更控制 59
  - 4.5.3 贸易公司的变更控制 60
  - 4.5.4 伪造批准 61
- 4.6 时刻不要忘记风险 61
- 4.7 通过变更控制推动配置管理流程 62
- 4.8 进入/退出标准 62
- 4.9 事后审查 63
- 4.10 自我评估 63
- 结论 64
- 第5章 发布管理 65
- 发布管理的目标 66
- 发布管理的原则 66
- 5.1 为什么发布管理如此重要 66
- 5.2 从哪里开始 67
- 5.3 发布管理的概念和实践 67
  - 5.3.1 可行的打包策略 67
  - 5.3.2 发布包版本识别 68
  - 5.3.3 发布版本的材料清单 68
  - 5.3.4 不可变ID意味着什么 68
- 5.4 发布管理人类工程学 68
  - 5.4.1 避免人为错误 69
  - 5.4.2 了解技术 69
  - 5.4.3 构建工程工具 69
  - 5.4.4 避免人为错误 70
  - 5.4.5 三步走 70
  - 5.4.6 太多可变部分 70
- 5.5 发布管理的协调职能 71
  - 5.5.1 沟通发布状态 71
  - 5.5.2 不要忘记发布日程表 71
  - 5.5.3 发布管理和配置控制 71
- 5.6 需求跟踪 71
- 5.7 将发布管理提升到新的层次 72
  - 5.7.1 使用加密技术签名代码 72
  - 5.7.2 操作系统对发布管理的支持 72
  - 5.7.3 改善你的发布管理过程 73

结论	73
第6章 部署	75
部署的目标	76
部署的原则	76
6.1 为什么部署很重要	76
6.2 从哪里开始	77
6.3 实践和实例	77
6.3.1 发布中转区	77
6.3.2 脚本控制发布过程	78
6.3.3 部署框架	78
6.3.4 如果鲍勃犯了个错误怎么办	79
6.3.5 细说存储库	79
6.3.6 审计发行版本	79
6.4 进行配置审计	80
6.5 不要忘记冒烟测试	80
6.6 小失误导致大问题	81
6.7 沟通计划	81
6.8 部署应当授权	82
6.9 信任也要核查	82
6.10 改进部署过程	82
结论	82
第 部分 架构和硬件配置管理	83
第7章 为配置管理设计应用程序架构	85
为配置管理设计应用程序架构的目标	86
7.1 为什么架构很重要	86
7.2 从哪里开始	87
7.3 配置管理如何促进良好的架构	87
7.4 架构师可以从测试人员那里学到什么	87
7.5 配置管理驱动开发	88
7.6 应对不断变化的架构	89
7.7 使用源代码管理促进架构	89
7.8 培训是关键	89
7.9 作为服务的源代码管理	90
7.10 作为服务的构建工程	90
结论	90
第8章 硬件配置管理	91
硬件配置管理的目标	92
8.1 为什么硬件配置管理的重要	92
8.2 从哪里开始	92
8.3 当无法版本控制电路芯片时	93
8.3.1 配置项的任何其他名称	93
8.3.2 设计规范的版本控制	93
8.4 不要忘记接口	93
8.5 了解依赖关系	94
8.6 可追溯性	94
8.7 部署变更到固件	94
8.8 硬件配置管理的未来	94
结论	95
第 部分 配置管理中人的因素	97

## 第9章 合理精简过程 99

合理精简配置管理过程的目标 100

9.1 为什么合理精简配置管理过程很重要 101

9.2 从哪里开始 101

9.3 繁琐的过程只会成为障碍 102

9.4 软件过程改进网络和推广能力成熟度模型 102

9.5 正在消失的烦琐过程 103

9.5.1 敏捷开发过程就是有用 103

9.5.2 开放统一过程 104

9.5.3 变得精益 104

9.5.4 希望能够激励人仔细了解精益软件开发的一个非常简短的描述 104

9.6 过程太少的危险 105

9.7 恰好够用的过程改进 105

9.8 不要过度工程化配置管理 105

9.9 不要忘了技术 106

9.10 测试自己的过程 106

9.11 过程咨询 106

9.12 创建一个可持续发展的结构 107

结论 107

## 第10章 克服变革的阻力 109

克服变革阻力的目的 110

10.1 为什么克服变革阻力很重要 111

10.2 从哪里开始 111

10.3 过程与企业文化相匹配 111

10.4 心理学和计算机程序设计相结合 112

10.5 从内部进行过程改进 113

10.6 选择首先要解决的问题 114

10.7 培养团队协作 114

10.8 为什么优秀的开发人员反对过程改进 115

10.9 程序公正 115

10.10 听取每个人的意见 115

10.11 展现领导能力 116

10.12 实施过程改进的人本身可能会成为问题 116

10.13 过程和技术培训相结合 116

10.14 倾听节奏 117

10.15 过程需要得到测试 118

10.16 婴儿般的步伐和过程改进 119

10.17 推销过程改进 119

10.18 什么是我需要的 119

10.19 作为服务的过程改进 120

10.20 过程改进的游击战术 120

结论 121

## 第11章 个性与配置管理：一位心理学家眼中的工作场所 123

了解个性的目的：对我而言有何用处 124

11.1 配置管理专业人员的个性处理 125

11.2 配置管理专家从个性的角度所要考虑的因素 128

11.2.1 沟通风格 128

11.2.2 男人和女人使用和解释语言或有差异 128

11.2.3 有效的协商 129

- 11.2.4 信息的核实 129
- 11.2.5 信息处理的偏好 130
- 11.2.6 工作中的出生顺序 131
- 11.2.7 作为领导者的长子 131
- 11.2.8 作为妥协者的老二 131
- 11.2.9 作为发起者的老幺 132
- 11.2.10 独生子 132
- 11.2.11 做你自己 133
- 11.3 心理学在工作场所的应用 133
  - 11.3.1 有效的团队协作从家庭开始 133
  - 11.3.2 排球或有效协作 134
  - 11.3.3 把构建工程师和测试人员嵌入开发团队中 134
  - 11.3.4 黑盒、白盒以及灰盒测试的对比 134
  - 11.3.5 破坏性的小组形态 135
  - 11.3.6 适合配置管理和质量检测的位置 135
- 11.4 家庭动态 135
- 11.5 工作场所的文化和个性 136
  - 11.5.1 个性和结构 137
  - 11.5.2 我们已经发明了所有的好点子 137
  - 11.5.3 我行我素，不守规矩 138
  - 11.5.4 在保持列车运行的同时保持有效的监督 138
  - 11.5.5 成功的配方 139
  - 11.5.6 注意事项 139
- 结论 139
- 第12章 从错误中吸取教训 141
  - 从错误中吸取教训的目的 142
    - 12.1 从错误中吸取教训的重要性 142
    - 12.2 从错误中吸取教训的第一步 142
    - 12.3 明白我们的错误 142
    - 12.4 我所犯的 error 143
      - 12.4.1 缺乏大局观 143
      - 12.4.2 编写发布自动化脚本是一项很有挑战性的工作 144
      - 12.4.3 关于良好的进程会自我运行的思考 144
      - 12.4.4 未能取得共识 145
      - 12.4.5 未能在配置管理上展现领导能力 145
      - 12.4.6 成为问题的一部分 145
      - 12.4.7 忘记向他人寻求帮助 146
  - 12.5 把错误变成教训 146
    - 12.5.1 明确知道如何做才能完成工作 146
    - 12.5.2 获得所需要的培训 146
  - 12.6 他人常犯的 error 147
    - 12.6.1 象牙塔 147
    - 12.6.2 没能提高自己的技术和动手能力 147
    - 12.6.3 缺乏诚实和坦然的态度 147
- 结论 148
- 第 部分 合规、行业标准和框架 149
- 第13章 建立IT控制及合规性 151
  - 建立IT控制及合规性的目标 152
    - 13.1 为什么IT控制及合规性很重要 153

- 13.2 建立IT控制及合规性的第一步 153
- 13.3 理解IT控制及合规性 154
  - 13.3.1 2002年发布的“萨班斯-奥克斯利法案” 154
  - 13.3.2 内部控制的管理评估 154
  - 13.3.3 发起机构委员会 155
  - 13.3.4 用于IT控制框架的COBIT 155
  - 13.3.5 核实并汇报管理层所做出的评估 155
  - 13.3.6 1996年发布的健康保险隐私及责任法案 156
  - 13.3.7 当美国审计署来敲你门的时候 156
  - 13.3.8 审计结果 157
  - 13.3.9 美国审计署关于国家档案记录管理局的配置管理实践的报告 158
  - 13.3.10 美国电子文件档案馆的配置管理规划 158
  - 13.3.11 有待改善的领域 159
  - 13.3.12 了解审计结果 159
  - 13.3.13 美国金融管理局 159
- 13.4 必不可少的合规性要求 160
  - 13.4.1 为版本发布提供可追溯的需求 160
  - 13.4.2 控制生产分离 160
- 13.5 支持配置管理最佳实践的道德观点 161
- 13.6 通过合规性来提高工作质量和效率 161
- 13.7 进行配置管理评估 162
  - 13.7.1 评估的第一步 162
  - 13.7.2 无论出现多么糟糕的情况也要先留心去听 163
- 结论 164
- 第14章 行业标准和框架 165
  - 使用行业标准和框架的目标 166
    - 14.1 为什么标准和框架很重要 166
    - 14.2 以IT控制及合规性为最佳实践的第一步 166
    - 14.3 必知的专业术语 167
      - 14.3.1 配置项 167
      - 14.3.2 配置标识 167
      - 14.3.3 配置控制 168
      - 14.3.4 接口控制 168
      - 14.3.5 配置状态统计 168
      - 14.3.6 配置审计 169
      - 14.3.7 分包商/供应商的管理手段 169
      - 14.3.8 符合规范与违规 170
    - 14.4 将这些条款应用在标准和框架里 170
    - 14.5 行业标准 170
      - 14.5.1 IEEE 828——标准软件配置管理方案 171
      - 14.5.2 ISO 10007质量管理体系——配置管理的指导方针 172
      - 14.5.3 ANSI/ITAA EIA-649-A——配置管理的国家统一标准 172
      - 14.5.4 ISO/IEC/IEEE 12207和15288标准 173
    - 14.6 行业框架 173
      - 14.6.1 ISACA COBIT 173
      - 14.6.2 能力成熟度模型/能力成熟度模型集成 182
      - 14.6.3 itSMF的ITIL框架 183
      - 14.6.4 软件工程知识体系 189
      - 14.6.5 开放统一过程(OpenUP) 190

14.6.6 敏捷/SCRUM 190  
结论 191

# 《配置管理最佳实践》

## 精彩短评

- 1、只能用垃圾来形容了，千万不要浪费时间读。
- 2、#配置管理#
- 3、配置管理方面很不错的书。
- 4、理论太多，简单扫读即可
- 5、在不到200页的内容里，即讲了配置管理涉及的六大主题，也讲了过程改进、克服组织变革阻力等“高阶”话题，最后还给出了业界标准框架的概述，内容相当丰富，但很多内容不免浅尝辄止，不够深入。对于非配置管理领域的人来说，是一本快速熟悉这个领域的“地图册”。

# 《配置管理最佳实践》

## 精彩书评

1、和传统的配置管理不同的是，这本书中介绍的东西偏实用性。其中包括：1.源代码管理。源代码管理注意分支和主干开发策略。2.构建工程。构建可以熟悉Maven的使用。3.环境配置。这块可以借鉴避免硬编码的实践。现在很多研发团队都在使用硬编码，造成了很多配置的遗漏。4..发布管理。这里面说的发布是指构建成功后，打包以致部署至目标环境的过程。

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu111.com](http://www.tushu111.com)