

版本控制之道——使用Git

Pragmatic Version Control
Using Git

程序员修炼三部曲 第一部



[美] Travis Swicegood 著
董越 付昭伟 等译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

程序员修炼三部曲 第一部

The Pragmatic Starter Kit-Volume I

版本控制之道——使用 Git

Pragmatic Version Control Using Git

[美] Travis Swicegood 著

董 越 付昭伟 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

《程序员修炼三部曲》丛书包含了三个部分，旨在帮助程序员解决日常工作中遇到的一些具体问题，内容覆盖了对于现代软件开发非常重要的基础知识。这套丛书展现了注重实效的实际技巧及工具使用方面的内容。

《版本控制之道》系列是三部曲中的第一部，它讲述了如何使用版本控制为项目提供安全保障，并提高开发、集成和发布的效率。随着版本控制工具 Git 越来越流行，《版本控制之道》在陆续推出了 CVS 版、Subversion 版后，现在也推出了 Git 版，即本书。

Git 是一个功能强大的工具，这也意味着完全掌握它是件颇费时间的事。而本书的特点是实用：介绍 Git 的精髓和常用的功能，让读者迅速上手，很快就可以在实际项目中使用 Git 并受益。

978-1-934356-15-9: Pragmatic Version Control Using Git.

All rights reserved. Authorized translation from the English language edition published by The Pragmatic Programmers, LLC.

本书简体中文专有翻译版权由 The Pragmatic Programmers, LLC. 授予电子工业出版社。专有版权受法律保护。

版权贸易合同登记号 图字：01-2010-2385

图书在版编目（CIP）数据

版本控制之道：使用 Git/（美）斯威司古德（Swicegood,T.）著，董越等译. —北京：电子工业出版社，2010.5
（程序员修炼三部曲 第一部）

书名原文：Pragmatic Version Control Using Git

ISBN 978-7-121-10719-1

I. ①版… II. ①斯… ②董… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字（2010）第 068450 号

策划编辑：徐定翔

责任编辑：陈元玉

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：12.75 字数：195 千字

印 次：2010 年 5 月第 1 次印刷

定 价：39.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

读者是怎么评价 《版本控制之道——使用 Git》的

What readers are saying about Pragmatic
Version Control Using Git

《版本控制之道——使用 Git》是一名优秀的向导，把你带入 Git 世界。它不仅教会你如何开始使用 Git，而且告诉你如何通过 Git 与团队成员合作，以及如何让历史记录保持整洁。

► **Pieter de Bie**

Author, GitX

如果你正在考虑开始使用 Git，那么我强烈推荐你阅读此书。如果你还没有使用任何版本控制系统，请放下这本书，给自己两下，然后重新拿起这本书，去结账。

► **Jacob Taylor**

Entrepreneur and Cofounder, SugarCRM Inc.

这本书不仅让我确信 Git 比 CVS 或 Subversion 都要好，而且它向我展示了可以自己使用 Git 并受益，同时保持与周围 CVS/Subversion 环境的连接。尽管 Git 是我接触的第一个分布式版本控制系统，期待在阅读本书后，也能尽快成为 Git 的使用者。

► **Chuck Burgess**

2008 PEAR Group Member

把 Git 这样一件复杂的事情用简单的话说明白，Travis 卓越地完成了这项任务。阅读本书，你会发现分布式版本控制系统不再是一件神秘难解的事。

► **Mike Mason**

Author, Pragmatic Version Control Using Subversion

联系博文

您可以通过如下方式与本书的出版方取得联系。

读者信箱: reader@broadview.com.cn

投稿信箱: bvtougao@gmail.com

北京博文视点资讯有限公司 (武汉分部)

湖北省 武汉市 洪山区 吴家湾 邮科院路特 1 号 湖北信息产业科技大厦 1402 室

邮政编码: 430074

电 话: 027-87690813

传 真: 027-87690595

欢迎您访问博文视点官方博客: <http://blog.csdn.net/bvbook>

译序

《版本控制之道——使用 CVS》原著出版于 2003 年秋，中文版出版于 2005 年春。

《版本控制之道——使用 Subversion》原著出版于 2005 年春，中文版（原著第 2 版）出版于 2007 年春。

《版本控制之道——使用 Git》原著出版于 2008 年冬，中文版（即本书）即将出版。

Git 的时代到来了。一些著名的项目如 Linux 内核、Mozilla、Gnome 及 Ruby on Rails 等都已选择使用以 Git 为代表的分布式版本控制系统（Distributed Version Control System, DVCS）。与传统的集中式版本控制系统（如 CVS、Subversion 和 ClearCase）相比，Git 具有若干显著优势，其中最重要的两点是：Git 轻而易举地支持多站点开发和离线工作；Git 的性能优于传统版本控制工具若干数量级。前者使得 Git 适用于更多的应用场景，后者使得 Git 适用于更大的研发项目。当然，即便是普通的中小型项目，也同样可以从 Git 的强大功能和高性能中受益。

《版本控制之道——使用 Git》秉承了 Pragmatic Bookshelf 系列图书的一贯传统：通俗易懂，精练实用。事实上，这一点对本书原作者是很大的挑战，因为 Git 本身博大精深，能以务实的精神讲清楚 Git，并在内容上取舍得当，实属不易。

本书由 SCMLife 社区 (www.SCMLife.com) 的如下成员翻译：董越（流水先生）、石振勇（Zhenyong）、林伟玲（pink_vivi）、陈靖贤（chenjingxian）、付昭伟（v2_smth）、李晓琴（lxq8081）、文娟（luck_cywj）、任文科（kane）、杨素萍（michelle_0510）、霍林莉（vino）、戴娅利（Daly）、侯颖卓（风信子）、刘贺（He1983）。他们都是富有经验的软件配置管理工作，熟悉包括 Git 在内的各种版本控制工具。他们分工翻译了全书，并进行了交叉互审。最后，由付昭伟和董越分别进行了总审。

本书在翻译过程中添加了少量有关注释，主要有三种情形：

- 方便 Windows 上的读者使用 Git；
- 方便使用 ClearCase 的读者理解 Git；
- 对少量话题再多讲一点，讲得更透彻一点。

本书的翻译工作本身就是用 Git 管理的：每章对应于单独的文本文件；完成翻译后，译者检入并上传；评审人员通过 Git 看到不同的版本，以及每个版本的改动。英汉术语对照表是一个单独的文件，大家偶尔会同时改动它，Git 会负责合并和报告冲突。每个人工作在不同的地点，但 Git 把大家连在了一起。使用 Git，得心应手。

由衷感谢所有翻译人员，感谢他们严肃认真、尽心尽力的工作。在此也向博文视点的白爱萍、徐定翔和陈元玉三位编辑致以特别的感谢。本书的出版因为流程上的原因一波三折，耗费了三位编辑很多心力，是他们坚持不懈的努力使得本书能够在国内出版。

最后，期待所有有志于推动国内配置管理行业发展的朋友加入到我们中间来，加入到 SCMLife (www.SCMLife.com) 的大家庭中来。共同努力推广 Git 等配置管理技术、思想，做一名配置管理专家。

SCMLife 创始人 孙常波（懂你）

2010 年 3 月于北京

致谢

Acknowledgments

是的，我是本书作者。然而，本书能够出版发行，全靠众人相助。请允许我在这里占用少许篇幅，向他们致以诚挚谢意。

感谢 Dave Thomas 和 Andy Hunt，他们把写作本书的机会提供给我这个新作者。

他们所组织的 Pragmatic Bookshelf 团队令人惊叹。感谢本书编辑 Susannah Davidson Pfalzer 所提供的无时无刻不在的帮助。

感谢所有浏览本书并提供意见和建议的人们。特别是技术评审人员：Chuck Burgess、Pieter de Bie、Stuart Halloway、Junio Hamano、Chris Hartjes、Mike Mason、John Mertic、Gary Sherman、Jacob Taylor 和 Tommi “Tv” Virtanen。也感谢 SugarCRM 和 Ning 的开发人员和其他同事，在本书写作过程中，他们给了我很多的支持。

最后，感谢我的朋友和家人，感谢他们的理解和支持。本书写作期间，我的夫人 Meg 不得不承担全部家务，却毫无怨言。没有她及其他朋友和家人的支持，本书无法面世。

序言

Preface

软件开发领域正在发生着全球性的变化。越来越多的软件开发团队放弃了老式的、笨重的**集中式**（*centralized*）版本控制系统（Version Control System, VCS），转而采用一种称为 Git 的新型的、轻量级的分布式版本控制系统（Distributed Version Control System, DVCS）。

下面是关于版本控制系统的简要介绍：版本控制系统就好比银行保险箱。保险箱保管有价值的资产，保证它们的安全。对程序员来讲，源代码就是这样的资产，由版本控制系统来保管。¹版本控制系统把程序员所完成并**提交**（*commit*）的任何修改都记录下来，供日后查询检索。这个功能就好像银行提供的对账单一样。

在 Git 的世界里，可以随身带着有自动取款机的保险箱到处走。这意味着，你既可以完全断开和别人的连接以独立工作，也可以在适当的时候与大家沟通分享工作成果。当然，Git 也具有记录和跟踪代码修改历史这样的基本功能。当初 Linux Torvalds 创造 Git 是为了跟踪 Linux 内核的修改情况。他用几周时间完成了基本的雏形。随着时间的推移，Git 已经从最初简陋的脚本发展成为功能丰富的工具包。对程序员来说，Git 具有如下优势：

- **分布式体系结构**：可以完全断网工作，不受网络连接的限制。
- **分支与合并操作很容易**：创建分支简单、经济、快速，这与其他版本控制系统不一样。Git 把分支上的所有修改合并回父分支，即使多次，也只是一眨眼的功夫。

¹ 写作本书的时候，美国政府正对银行系统实施 7 000 亿美元的救助，所以银行可能不是最合适的比方。不必考虑这方面，只要考虑银行的工作方式就可以了。

- **跟 Subversion 进行交互：**如果你是公司里唯一打算使用 Git 的人，其他人仍然在使用 Subversion，不要担心。Git 可以从 Subversion 的版本库中导入所有的历史，并把你在 Git 中的改动发送回 Subversion 的版本库。

以上是对 Git 的一分钟介绍，本书将在此基础上展开。

谁适合阅读本书

每个人都可以从本书中学到新东西。没有接触过版本控制的人，也可以从头学习本书前几章，了解版本控制和 Git 的基本概念。

能够熟练使用 Subversion 或 CVS 的开发人员，可以在大致翻阅第 1 篇后仔细阅读第 2 篇，以深入细致地学习 Git 命令及其用法。

内容概述

本书分 4 篇：

- 第 1 篇从 Git 的视角来介绍版本控制。

第 1 章“Git 的版本控制之道”（第 3 页）从总体上介绍了版本控制和一些 VCS 基础概念，以及 DVCS 的不同之处。

第 2 章“Git 安装与设置”（第 15 页）详细介绍了 Git 的安装和配置。

第 3 章“创建第一个项目”（第 25 页）在一个简单的 HTML 项目²中演示 Git 的使用，以展示 Git 的丰富功能。

第 2 章和第 3 章须要上机操作，所以请准备好计算机。

- 第 2 篇开始展开介绍。第 4 章“添加与提交：Git 基础”（第 41 页）介绍了最基本的命令，包括准备好版本库、提交修改等。

接下来是关于分支的介绍，第 5 章“理解和使用分支”（第 55 页）。在 Git 里，分支是非常重要的操作。这里将用一整章来解释什么是分支，以及怎样使用它们。

² 即使不了解 HTML，也不用担心——我们的例子很简单。

学习了以上两章之后，将开始讲解如何查看历史记录：

第 6 章“查询 Git 历史记录”（第 71 页）。

第 7 章“与远程版本库协作”（第 91 页）介绍了使用远程版本库共享工作的方法。对任何版本管理工具来说，支持协作都是其关键功能，Git 也是这样。

当学习完如何使用 Git，以及如何与其他开发人员的版本库进行交互后，你还要了解一些组织管理版本库的技巧。第 8 章“管理本地版本库”（第 101 页）介绍了如何让本地版本库保持整洁。

第 9 章“高阶功能”（第 115 页）是第 2 篇的结尾，本章介绍了一些在特定情况下非常有用的命令。

这些章节均包含了丰富的实例，但请注意，这些仅仅是起点。熟悉了这些实例以后，要进行更多的探索和实践，以便在真实项目中运用自如。

- 第 3 篇介绍系统管理。如果你所在的团队或你的公司已经有专人负责系统管理，则可以略过这部分。第 10 章“迁移到 Git”（第 131 页）介绍了如何从其他常见的版本控制系统迁移到 Git。第 11 章“使用 Gitis 管理 Git 服务器”（第 143 页）介绍了如何利用 Gitis 管理公共版本库。
- 本书最后是附录。附录 A（第 155 页）是命令索引，可用于快速查阅常用命令及其使用方法。

附录 B（第 165 页）介绍了 Git 配套工具，以及其网上资源的链接。其中一些工具是 Git 安装包自带的，另一些链接到在线资源，须要另行安装。

最后，附录 C（第 173 页）是参考书目列表。

排版约定

本书使用如下排版约定：

Git	首字母大写表示 Git 工具本身。
git	表示命令行中的命令。
黑体	表示新概念。
files 和 directories	文件和目录用这种字体显示。
prompt>	表示须要输入的命令。较长的命令将在每行行尾用“\”截为几行。当然，它们可以在一行中输入完。但为了便于排版，本书中的长命令通常分成几行。 ³

在线资源

在第 2 篇中，每章开头给出了一个版本库，其内容与你按顺序完成本书此前各章的例子后所得到的版本库的内容相同。如果在学习时跳过了前面的一些章节，也可以从 GitHub 克隆这个版本库。在我的 GitHub 主页上可以找到这些版本库：
<http://github.com/tswicegood>

每章的开头都给出了克隆相应版本库的命令。

本书在 pragprog.com 网站上有个主页⁴，可以浏览该页面，获得本书的相关信息。你可以在论坛里给我发消息，也可以在勘误表页面中提出建议或修正。

读到这里，现在脑海里是不是有很多关于 Git 的问题，想知道答案？不要再耽搁了，让我们马上开始学习 Git！

³ Windows 命令行不支持“\”这种使用方式。

⁴ <http://www.pragprog.com/titles/tsgit>

目录

Contents

致谢	I
序言	III
谁适合阅读本书	IV
内容概述	IV
排版约定	VI
在线资源	VI
第 1 篇 欢迎来到分布式世界.....	1
第 1 章 Git 的版本控制之道	3
1.1 版本库	4
1.2 版本库中存储什么	5
1.3 工作目录树	6
1.4 代码修改与文件同步	6
1.5 跟踪项目、目录和文件	7
1.6 使用标签跟踪里程碑	8
1.7 使用分支来跟踪并行演进	9
1.8 合并	10
1.9 锁机制	12
1.10 下一步	13
第 2 章 Git 安装与设置.....	15
2.1 安装 Git	15
2.2 设置 Git	20
2.3 使用 Git 图形界面（GUI）	22
2.4 获取 Git 内置帮助信息	23

第3章 创建第一个项目	25
3.1 创建版本库	26
3.2 代码修改	26
3.3 在项目中工作	29
3.4 理解并使用分支	32
3.5 处理发布	33
3.6 克隆远程版本库	37
第2篇 Git 日常用法	39
第4章 添加与提交: Git 基础	41
4.1 添加文件到暂存区	42
4.2 提交修改	45
4.3 查看修改内容	48
4.4 管理文件	51
第5章 理解和使用分支	55
5.1 什么叫分支	56
5.2 创建新分支	57
5.3 合并分支间的修改	59
5.4 冲突处理	64
5.5 删除分支	67
5.6 分支重命名	68
第6章 查询 Git 历史记录	71
6.1 查看 Git 日志	72
6.2 指定查找范围	73
6.3 查看版本之间的差异	76
6.4 查明该向谁问责	77
6.5 跟踪内容	79
6.6 撤销修改	82
6.7 重新改写历史记录	85
第7章 与远程版本库协作	91
7.1 网络协议	91
7.2 克隆远程版本库	94
7.3 版本库同步	95

7.4	推入改动	96
7.5	添加新的远程版本库	97
第 8 章	管理本地版本库	101
8.1	使用标签标记里程碑	102
8.2	发布分支的处理	104
8.3	标签与分支的有效名称	106
8.4	记录和跟踪多个项目	107
8.5	使用 Git 子模块跟踪外部版本库	108
第 9 章	高阶功能.....	115
9.1	压缩版本库	116
9.2	导出版本库	117
9.3	分支变基	118
9.4	重现隐藏的历史	121
9.5	二分查找	124
第 3 篇	系统管理.....	129
第 10 章	迁移到 Git.....	131
10.1	与 SVN 的通信	131
10.2	确保 git-svn 是可用的	135
10.3	导入 Subversion 版本库	136
10.4	与 Subversion 版本库保持同步更新.....	138
10.5	将修改推入 SVN	140
10.6	从 CVS 导入.....	141
第 11 章	使用 Gitis 管理 Git 服务器	143
11.1	确定 Gitis 所依赖的程序已经安装	144
11.2	安装 Gitis	145
11.3	创建管理员 SSH 证书	145
11.4	配置 Gitis 服务器	146
11.5	初始化 Gitis	147
11.6	配置 Gitis	147
11.7	添加新版本库.....	148
11.8	设置公共版本库.....	150
11.9	结束语.....	151

第 4 篇 附录	153
附录 A Git 命令快速参考	155
A.1 安装和初始化.....	155
A.2 日常操作.....	156
A.3 分支.....	157
A.4 历史.....	159
A.5 远程版本库.....	161
A.6 连接 Git 和 SVN	162
附录 B 其他资源和工具	165
B.1 Git 附带工具.....	165
B.2 第三方工具.....	166
B.3 Git 版本库托管服务.....	169
B.4 在线资源.....	170
附录 C 参考书目	173
索引	175

