

代码管理核心技术与实践



[代码管理核心技术与实践_下载链接1](#)

著者:刘冉

出版者:电子工业出版社

出版时间:2018-1

装帧:平装

isbn:9787121328497

《代码管理核心技术与实践》首先通过系统化的介绍和比较，从整体上讲解了代码管理工具和系统的历史和发展。其次分别从小型团队、中大型团队、分布式大团队、基于微

服务的团队及开源团队的角度总结了代码管理的核心技术及实践经验，其中包括不同类型的团队对代码管理工具和系统的选择，以及代码管理的流程、策略和技巧，还有一些代码管理工具和系统的难点、痛点等，包括如何选择分支策略、如何管理多产品线的代码、代码备份策略，以及如何在大型团队中将代码从Subversion迁移到Git等。《代码管理核心技术及实践》可帮助读者在现实中从团队的大小及代码管理模式是集中式还是分布式、开源还是闭源等各个角度去了解和思考代码管理的核心技术和实践经验，从而帮助团队建立起一套高效的代码管理系统、策略和流程。

《代码管理核心技术及实践》的读者对象主要是每天都需要使用代码管理工具的程序员、代码管理工具和系统的管理人员，以及团队的技术领导人员。

作者介绍:

刘冉，资深软件质量咨询师，拥有超过13年的软件开发和测试工作经验，熟悉自动化测试系统开发及敏捷中的QA，深入理解软件测试及SCM、CI。现在关注软件测试全自动化和敏捷中的QA，以及如何帮助大型团队有效地管理代码和CI，其中包括如何通过有效的代码分支管理、代码提交及CD保证和改进软件的质量。

肖然，精益敏捷专家，在过去15年的从业经历中，先后从事了算法复杂度研究、工业软件开发、全球项目管理，以及大型企业转型等工作。始终把软件开发作为自己的爱好，在各大企业和社区宣扬精益和敏捷的开发思想，践行有高响应力的开发理念。

覃宇，高级软件咨询师，拥有超过10年的移动应用开发经验，为Android技术专家、Git资深用户和狂热爱好者、“主干开发”的坚定拥护者和实践者，曾帮助多个客户团队改进代码管理、依赖管理、分支策略、持续集成等技术实践。

目录: 第1部分 基础与传统

第1章 代码版本管理工具与系统 2

1.1 引言 2

1.2 代码版本管理工具的历史 3

1.2.1 第1代：本地代码管理 3

1.2.2 第2代：中心服务器代码管理 3

1.2.3 第3代：分布式代码管理 4

1.3 常用的代码管理工具 5

1.3.1 Perforce 5

1.3.2 Subversion 6

1.3.3 Git 6

1.3.4 Mercurial 7

1.3.5 Microsoft GVFS (Git Virtual File System) 7

1.4 常用的代码管理系统 8

1.4.1 Virtual SVN Server 9

1.4.2 GitLab Server 9

1.4.3 Gerrit Server 10

1.5 从Subversion迁移到Git的常用工具和方法 11

1.5.1 SubGit 11

1.5.2 git-svn 12

1.5.3 手动 12

1.6 常用云端代码管理系统 13

1.6.1 Sourceforge和Google Code 13

1.6.2 GitHub 14

1.6.3 GitLab和Bitbucket 14

1.6.4 Coding、码云、阿里云Code 15

第2章 独立小型团队	17
2.1 启程：团队与项目	17
2.2 痛点与需求	18
2.2.1 如何选择和搭建Subversion Server	18
2.2.2 定制代码库结构	20
2.2.3 分支策略	22
2.2.4 日常工作模式	24
2.2.5 备份策略	26
2.3 阿里云Code	27
2.3.1 将内网Subversion迁移到阿里云Code	28
2.3.2 权限管理	31
2.3.3 日常工作模式	32
2.3.4 备份方案	33
2.4 小团队代码管理的经典模型	34
第3章 传统中大型团队	36
3.1 传统大型团队的特点	36
3.2 独立大型团队在代码管理上的痛点与需求	38
3.3 大型团队代码管理案例	39
3.3.1 代码模块依赖管理	41
3.3.2 建立相关运作机制	44
3.3.3 建立原子提交的纪律	46
3.3.4 建立持续集成守护机制	47
3.3.5 大型团队代码管理小结	51
3.4 大型团队的代码服务器迁移	51
第2部分 当前与流行	
第4章 分布式中大型团队	58
4.1 分布式中大型团队的特点	58
4.2 分布式中大型团队在代码管理上的痛点与需求	59
4.2.1 离线代码管理	60
4.2.2 在线代码审查	61
4.2.3 对代码进行分布式权限管理	66
4.2.4 对代码进行分布式提交和集成	73
4.3 代码仓库拆分与集成	74
4.3.1 优化单代码仓库	77
4.3.2 代码仓库的拆分	87
4.3.3 代码仓库的集成	91
4.3.4 小结	122
4.4 分支策略	123
4.4.1 主干开发分支策略	124
4.4.2 应对并行开发	132
4.4.3 定制分支策略	147
4.5 代码库热备份	150
4.5.1 服务器端热备份方案	150
4.5.2 客户端热备份方案	151
4.6 案例：Android定制化系统开发	151
4.6.1 项目背景	151
4.6.2 项目及其代码管理介绍	152
4.6.3 分支策略	155
4.7 多产品线	157
4.7.1 多产品线介绍	158
4.7.2 多产品线开发的困境	158
4.7.3 多产品线解决方案	158
4.8 超大型分布式团队	166
第3部分 发展与未来	

第5章 云时代微服务大型分布式团队	172
5.1 云时代和微服务架构	172
5.2 Everything as Code（一切即代码）	173
5.3 代码管理团队自治	175
5.3.1 围绕团队的代码库管理	177
5.3.2 围绕服务的代码库管理	177
5.4 微服务架构下的代码管理挑战	179
5.5 微服务代码管理实例	180
第6章 开源项目与开源社区	184
6.1 开源软件	184
6.1.1 开源软件的特点	185
6.1.2 开源软件和社区	185
6.1.3 开源软件和商业	186
6.1.4 开源软件的代码管理	186
6.2 开源社区中的开源项目	187
6.2.1 简介	187
6.2.2 代码管理模型	187
6.2.3 典型的大型分布式开源项目	189
6.3 企业中的开源项目	193
6.3.1 简介	193
6.3.2 代码管理模型	193
6.4 GitHub中的开源项目实践	195
6.4.1 分支管理	195
6.4.2 分库管理	197
6.4.3 把公开代码库转换成私有代码库	203
6.4.4 GitHub的分支与复刻	205
参考文献	207
名词解释	209
• • • • •	(收起)

[代码管理核心技术及实践_下载链接1](#)

标签

git

计算机

subversion

github

coding

code

SCM

Git

评论

总的来说就是，小团队可以使用git,subversion 来管理代码，大团队可以用git管理代码，这些都是可以在本地自建server的，如果是云端的话，小团队大团队都有微服务式的架构。

太基础。

入门git书籍，讲了一下大概！

[代码管理核心技术及实践 下载链接1](#)

书评

[代码管理核心技术及实践 下载链接1](#)