

程序员的三门课：技术精进、架构修炼、管理探秘



[程序员的三门课：技术精进、架构修炼、管理探秘 下载链接1](#)

著者:于君泽

出版者:电子工业出版社

出版时间:2019-10

装帧:平装

isbn:9787121373770

我们都有一个共同的梦想——成为更棒的程序员，但是如何做？如何学习和精进自己的技术？如何做业务分析和架构设计？如何做技术管理？本书就广大程序员都很关注的问题提供一些思路和方法。

《程序员的三门课：技术精进、架构修炼、管理探秘》内容分为三大篇：第1篇介绍程序员的技能成长路径，介绍如何通过学习来加速成长，并讲解非常重要的业务分析和设计知识；第2篇详细介绍了架构修炼中的必备架构思维模式、架构设计过程，还通过真实案例剖析了架构设计的核心要素及关注点，以及如何通过架构设计来把控质量和风险；第3篇总结了做技术管理的诀窍，揭秘如何从准备做技术管理到实现自我管理再到实现团队管理。

《程序员的三门课：技术精进、架构修炼、管理探秘》适用于即将或者正在从事IT相关开发工作，却不知如何提升自己，希望通过适当的学习方法精进技术的初级程序员；也适用于具备一定架构设计经验，想进一步系统化提升架构设计能力的中级程序员；以及有一定技术积累，想从事技术管理却不知如何着手的高级程序员。

作者介绍：

于君泽

某互联网公司技术专家，在分布式系统架构设计、高并发系统设计、领域建模、研发管理、内建质量等方面有较多心得，乐于交流和分享，曾创立技术社群“中生代技术”。

维护技术公众号“技术琐话”。

李伟山

现任某人工智能公司技术总监，江湖人称“山哥”，在分布式系统架构设计、高并发系统设计、系统稳定性保障等方面积累了丰富的实践经验，擅于利用数据分析解决实际问题，对新技术有浓厚的兴趣，乐于分享。

个人技术公众号：技术方舟。

张洪亮

网名Hollis，知名技术博主，个人技术博客(<http://www.hollischuang.com>)阅读量达数百万。90后程序员，2015年毕业于哈尔滨理工大学，现任某互联网公司技术专家，专注于Java研发、技术架构及程序员的成长等方面。

个人技术公众号：Hollis。

彭首長

熊猫优福联合创始人，聚美优品前技术总监。在团队管理、领域设计、分布式架构、高并发设计、IoT架构等方面有丰富的实践经验，擅长行业分析，对新技术高度敏感，热衷于技术实践且乐于分享。

刘朋

高绩效团队教练，教练型领导，中生代技术社区联合创始人，北京大学理学学士、工学硕士。现就职于中国移动（成都）5G产业研究院，曾任诺基亚通信（成都）有限公司研发经理。在敏捷实践、高绩效团队教练技术、技术团队管理等方面积累了丰富的经验，并总结出一套特别适合程序员突破自身瓶颈、快速成长的领导力模式。

目录: 第1篇 技术精进
第1章 程序员技能与成长 2

1.1 如何学习新的编程语言	2
1.1.1 重点学什么	3
1.1.2 学习方法	3
1.2 代码规范与单元测试	6
1.2.1 编码规范	6
1.2.2 单元测试	7
1.2.3 测试驱动设计	9
1.3 使用静态代码分析工具	18
1.3.1 什么是静态代码分析	19
1.3.2 静态代码分析工具	19
1.4 代码审查	24
1.4.1 什么是代码审查	24
1.4.2 代码审查的好处	24
1.4.3 如何做代码审查	25
1.5 清单和模板	29
1.5.1 清单实践	29
1.5.2 产品需求文档模板	31
1.6 程序员的工作法则	33
1.6.1 工具化法则	33
1.6.2 自动化法则	38
1.6.3 关于文档的问题	40
1.6.4 关于YAGNI	40
1.7 程序员的工具箱	41
1.7.1 可帮我们提升工作效率的硬件	42
1.7.2 可帮我们提升工作效率的操作系统	43
1.7.3 可帮我们提升工作效率的软件	43
1.7.4 在线工具	50
第2章 加速成长与学会学习	56
2.1 程序员如何加速成长	56
2.1.1 积极主动	56
2.1.2 空杯心态	57
2.1.3 选择合适的平台	57
2.1.4 别怕犯错	58
2.1.5 注意细节	58
2.1.6 时间管理	58
2.1.7 打破边界	61
2.1.8 写业务代码中的成长机会	61
2.2 学会学习	66
2.2.1 高效学习的方法	67
2.2.2 高效学习的途径	69
2.2.3 舒适区与小步子原则	72
2.2.4 两个著名的学习理论	72
2.2.5 终身成长	74
第3章 业务分析与设计	75
3.1 黄金圈法则	75
3.2 UML建模工具	77
3.2.1 用例图	78
3.2.2 类图	79
3.2.3 对象图	80
3.2.4 状态图	80
3.2.5 活动图	81
3.2.6 序列图	82
3.2.7 协作图	82
3.2.8 构件图	83

3.2.9 部署图	84
3.3 业务分析与设计的方法	85
3.4 系统分析与设计的三个发展阶段	87
3.4.1 面向数据驱动分析与设计	87
3.4.2 面向对象和服务分析与设计	88
3.4.3 面向问题域分析与设计	89
3.5 面向对象分析与设计	90
3.5.1 什么是面向对象	90
3.5.2 面向对象的特征	91
3.5.3 面向对象设计的原则	91
3.6 面向服务分析与设计	93
3.6.1 Web Service模式	93
3.6.2 ESB模式	94
3.6.3 微服务架构	95
3.7 领域驱动设计	97
3.8 领域模型实践	102
3.8.1 什么是领域模型	102
3.8.2 领域模型的作用	102
3.8.3 如何进行领域建模	104
第2篇 架构修炼	
第4章 架构思维	110
4.1 分解	110
4.1.1 分解的作用	111
4.1.2 分解的原则	111
4.1.3 分解的时机	112
4.2 集成	112
4.2.1 常见的集成方式	112
4.2.2 集成的难点	115
4.3 动静分离	120
4.4 复用	121
4.4.1 系统复用的分类	122
4.4.2 可复用性和可维护性的关系	123
4.4.3 可维护性地复用的设计原则	123
4.5 分层	126
4.6 模式	127
4.6.1 分层模式	128
4.6.2 客户端-服务器模式	131
4.6.3 主从设备模式	132
4.6.4 管道-过滤器模式	133
4.6.5 代理模式	133
4.6.6 P2P模式	134
4.6.7 事件总线模式	135
4.6.8 MVC模式	136
4.6.9 黑板模式	137
4.6.10 解释器模式	138
4.7 抽象	139
4.8 结构化	139
4.8.1 结构化的原则	140
4.8.2 结构化分析工具	140
4.8.3 结构化思维的7个环节	141
4.9 迭代	141
4.10 勿做过度设计	141
第5章 架构设计	143
5.1 架构设计概要	143

5.1.1 业务架构	143
5.1.2 应用架构	145
5.1.3 技术架构	147
5.1.4 数据架构	149
5.2 架构设计的流程	151
5.3 架构设计的核心要素	153
5.3.1 性能	153
5.3.2 可用性	153
5.3.3 伸缩性	154
5.3.4 扩展性	154
5.3.5 安全性	154
5.4 高性能设计	154
5.4.1 主要的性能测试指标	155
5.4.2 性能测试方法	156
5.4.3 性能优化策略	157
5.5 高可用设计	159
5.5.1 高可用性的度量与考核	159
5.5.2 高可用的架构	160
5.5.3 高可用质量保证	164
5.5.4 系统运行监控	165
5.6 可伸缩设计	166
5.6.1 架构的伸缩性设计	166
5.6.2 应用服务的伸缩性设计	167
5.6.3 分布式缓存的伸缩性设计	169
5.6.4 数据存储的伸缩性设计	169
5.7 可扩展性设计	169
5.7.1 通过消息降低系统的耦合性	170
5.7.2 通过分布式服务提供可复用的业务	170
5.7.3 建立开放平台	170
5.8 架构设计的典型案例——订单系统	171
5.8.1 订单的基本概念	172
5.8.2 订单流程	174
5.8.3 订单系统设计中的挑战和实践	178
5.8.4 数据库优化	182
5.8.5 小结	185
第6章 架构的保障：质量与风险	186
6.1 内建质量体系	186
6.2 从黑天鹅事件到墨菲定律	189
6.2.1 黑天鹅事件	189
6.2.2 蝴蝶效应	191
6.2.3 墨菲定律	191
6.3 软件质量稳定性之殇	192
6.3.1 业务高速发展带来的变化	193
6.3.2 问题域的复杂性	193
6.3.3 系统的复杂性	194
6.3.4 技术债问题	194
6.3.5 人、流程、文档的博弈	195
6.3.6 采用不能掌控的工具和框架	196
6.3.7 质量意识	196
6.4 从康威定律和技术债看研发之痛	197
6.4.1 康威定律	197
6.4.2 通过3个案例来看研发之痛	197
6.4.3 架构方案和业务形态息息相关	204
6.5 求解质量熵	206

6.5.1 运用敏捷思想	207
6.5.2 运用系统化思想	208
6.5.3 技术债偿还计划	209
6.5.4 抓住合适的时机做架构升级	209
6.5.5 内建质量	210
6.5.6 不迷信，不唯新、不唯上，实践是检验真理的标准	210
6.5.7 复杂的问题域：专项突破	210
6.5.8 领导者的意识	211
6.5.9 创新解决方案	211
6.6 踩过的坑和经验总结	212
6.6.1 简单的数字改动导致电商网站搜索功能失灵	212
6.6.2 应用竟然被外部系统拖垮了	212
6.6.3 HashMap与并发	213
6.7 故障复盘流程及模板	215
6.7.1 什么是故障复盘	215
6.7.2 为什么要做故障复盘	215
6.7.3 如何做故障复盘	216
6.7.4 故障复盘模板	219
6.8 监控与告警	221
6.8.1 监控的重要性	221
6.8.2 监控哪些内容	221
6.8.3 告警	224
6.8.4 监控的架构	224
6.9 应急处置	226
第3篇 管理探秘	
第7章 为成为技术主管做准备	232
7.1 构建自我阶段性目标	232
7.2 体验自己的目标身份	234
7.3 勇于抓住机会	235
第8章 从自我管理转为管理团队	236
8.1 什么是领导力	236
8.2 如何构建领导力	237
8.2.1 维度1：提供清晰的领导力风格，并以信任感作为基石	238
8.2.2 维度2：了解业务，并带领团队达到高绩效	240
8.2.3 维度3：发展自己和团队成员	240
8.2.4 维度4：塑造未来	241
8.3 让自己成为T型人才	242
8.4 高效时间管理	243
8.4.1 确定在做的事情符合自己的目标	244
8.4.2 随时应用80/20法则	246
8.4.3 创造大块时间	247
8.5 遇到“不服管”的员工怎么办	249
8.5.1 每个人都是不错的	250
8.5.2 建立亲和与信任感	251
8.5.3 从绩效评估到建设性反馈	255
8.6 如何处理冲突	261
8.6.1 了解产生冲突的原因	261
8.6.2 正确看待冲突	262
8.6.3 处理冲突	263
8.7 引导员工主动工作	264
8.8 如何从带10个人到带100个人	266
8.8.1 在管理上面对的挑战	267
8.8.2 从管理1.0到管理2.0	269
8.9 如何对上管理	276

8.9.1 做好对上的预期管理 277
8.9.2 及时汇报 277
· · · · · (收起)

[程序员的三门课：技术精进、架构修炼、管理探秘](#) [下载链接1](#)

标签

技术

管理

架构

进阶

计算机

软件开发

编程

生动

评论

@微信读书

19年等77本书，就是大型目录，说的都很空泛。不如直接看相关的名著。如敏捷相关，高效程序员的45个习惯等。内容太多，蜻蜓点水，跟名词解释无区别。如果想阔知识面，不如看

李知慧的大型互联网站，还有就是淘宝味有点重。内容还提到阿里规范和在阿里的历程，感觉对内容没有什么关系。何必强调自己的阿里身份。书中谈到太多，设计原则，面向对象，tdd，ddd。技术管理，还有不服气的员工怎办。2.5星吧。可以翻，没有太大价值。

只看了技术和管理的部分 感觉有点点到为止

过去知识和注意点的梳理

程序员不仅需要写好代码，还需要合理规划自己的职业发展路径。本书以作者多年实践经验总结为基础，从程序员的技能、成长、学习、进阶，到架构和管理等的成长周期做了深入分析，不仅讲到如何写出具有专业水准的代码，还讲到如何快速成长，以及如何转型架构和技术管理。

泛泛而谈，随手翻翻

一些通识的框架、管理方法的理论介绍。还可以吧，快速翻翻～

技术精进和管理探秘部分不错，但架构修炼部分感觉太官方化，有种直接从教材里面复制黏贴的感觉，领域驱动设计过于缺乏可操作性

简直就是垃圾，泛泛而谈，引用各种概念，毫无营养，70+买的，10元转卖不包邮

程序员都应该认真阅读的一本书

书写的很扎实，由浅入深。

提升的部分是不错，但是宣扬了一种加班的精神，12个小时的上班时间，
16小时才算加班，路上通勤的时间，
作者你不需要睡觉的吗？我们是人需要睡觉的，不是你家马爸爸生产的机器。

从技术到架构，从事到人，从业务到管理，本书为技术人员，提供了一条可复制的进阶之路。

这本书非常不错，涉及了不少的知识点，值得大家学习。一本好书，往往让你看破一些道理，甚至会觉得，原来这些其实就是这样回事啊。

越是看起来简单的道理，越是需要花心思去揣摩才能体会。就像许多看似简单的技术实现，没有亲身经历的，都是“二手知识”。网上有许多的学习资料，尤其是当我们打开某些信息流app时，其实那是你的“信息茧房”，每天阅读这些内容，你会以为技术的获得是多么的轻易，但只有手把手写过一波，才能真正获得价值。而阿里巴巴和众多互联网公司们，花了十年时间做了那么多东西，一定是走了弯路，踩了许多坑才最终获得了最佳实践，而这些在这本书里都有所介绍。

- 三大热点问题大讨论，想要讨好三拨人

- 技术人写书还是太轻松了，把平时是博文和搜集的资料理一理；技术书籍编辑的活也轻松，谁热谁牌不错就让谁上桌

[程序员的三门课：技术精进、架构修炼、管理探秘 下载链接1](#)

书评

[程序员的三门课：技术精进、架构修炼、管理探秘 下载链接1](#)