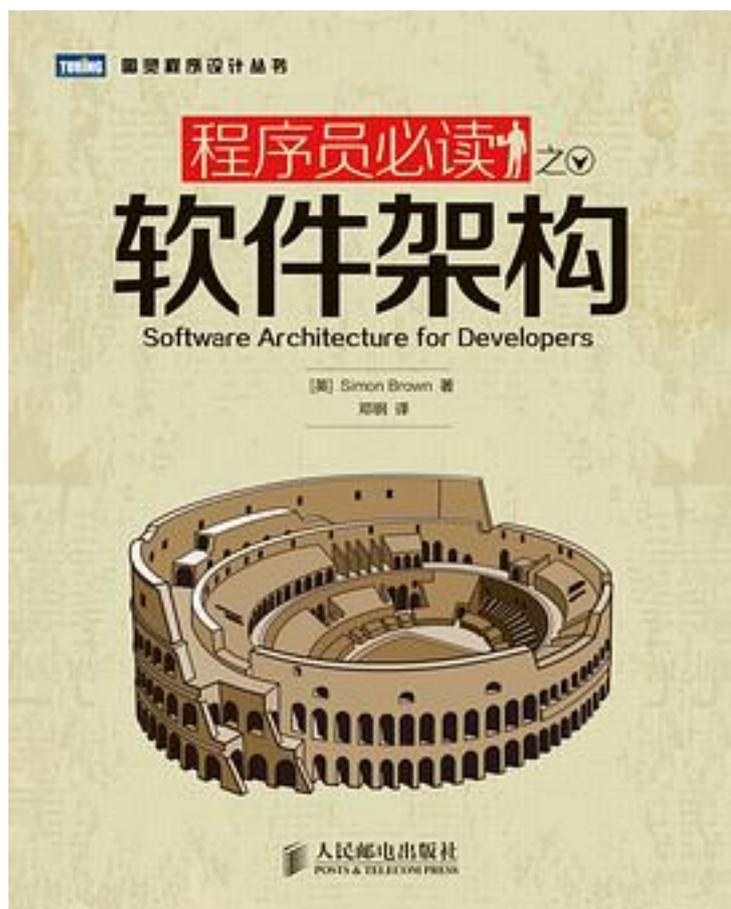


程序员必读之软件架构



[程序员必读之软件架构 下载链接1](#)

著者:[英] Simon Brown

出版者:人民邮电出版社

出版时间:2014-12

装帧:平装

isbn:9787115371072

通常，人们对软件架构师持两种错误的看法。有人认为软件架构师是一种高高在上的职位；有人认为软件架构师完全不懂开发，只是会画条条框框的指挥家。本书将打破这些传统的认知，模糊软件开发和架构在流程中的界限，进而为软件架构正名。本书是一本强调实践、注重实效、轻量级、面向开发者的软件架构指南。

作者介绍:

作者简介:

Simon Brown

全球知名软件架构独立咨询师、讲师，创办了专门讨论软件架构问题的网站“编码架构”（codingthearchitecture.com）。他自称是写代码的软件架构师和明白架构的软件开发者。自2008年以来的7年时间里，Simon在全球28个国家做过有关软件架构、技术领导力及其与敏捷的平衡等主题的百余场演讲，并于2012年8月在中国举办的ArchSummit全球架构师峰会上以“郁闷的架构师”和“如何设计安全的架构”为主题发表演讲，深受与会者好评。Simon已为全球20多个国家的软件团队提供咨询和培训，他的客户既有小型技术初创企业，也不乏全球家喻户晓的品牌公司。

译者简介:

邓钢

误打误撞进入IT行业的80后程序员，爱好Web技术，对前端技术尤其偏爱。曾在盛大创新院担任前端工程师，现在是IBM上海的一名软件用户界面工程师。除了具体的技术，对软件架构、软件工程也很感兴趣，希望把自己在IBM所见所闻分享出来，为前端领域如火如荼的工程化浪潮贡献力量。

目录: 推荐序一：架构师真正要学会的事情	ix
推荐序二	xii
译者序2.0	xiii
序	xvi
关于本书	xix
软件架构培训	xxii
Part I 什么是软件架构	
第1章 什么是架构	2
第2章 架构的种类	4
第3章 软件架构是什么	6
第4章 敏捷软件架构是什么	8
第5章 架构对上设计	11
第6章 软件架构重要吗	13
第7章 问题	15
Part II 软件架构的角色	
第8章 软件架构的角色	18
第9章 软件架构师应该编码吗	22
第10章 软件架构师应该是建造大师	25
第11章 从开发者到架构师	30
第12章 拓展T	32
第13章 软技能	34
第14章 软件架构不是接力运动	36
第15章 软件架构要引入控制吗	38
第16章 小心鸿沟	40
第17章 未来的软件架构师在哪里	42
第18章 每个人都是架构师，除非他们有其他身份	44
第19章 软件架构咨询师	46
第20章 问题	48

Part III	设计软件	
第21章	架构驱动力	50
第22章	质量属性（非功能需求）	52
第23章	处理非功能需求	55
第24章	约束	57
第25章	原则	60
第26章	技术不是实现细节	63
第27章	更多分层等于更高复杂度	66
第28章	协同设计是一把双刃剑	68
第29章	软件架构是对话的平台	70
第30章	SharePoint项目也需要软件架构	72
第31章	问题	74
Part IV	可视化软件	
第32章	沟通障碍	76
第33章	对草图的需要	78
第34章	无效的草图	81
第35章	C4：语境、容器、组件和类	91
第36章	语境图	94
第37章	容器图	98
第38章	组件图	102
第39章	是否包含技术选择	107
第40章	你会那样编码吗	110
第41章	软件架构和编码	112
第42章	你不需要UML工具	117
第43章	有效的草图	120
第44章	C4的常见问题	124
第45章	问题	126
Part V	为软件生成文档	
第46章	代码不会讲述完整的故事	128
第47章	软件文档即指南	131
第48章	语境	136
第49章	功能性概览	137
第50章	质量属性	139
第51章	约束	141
第52章	原则	143
第53章	软件架构	145
第54章	外部接口	147
第55章	代码	149
第56章	数据	151
第57章	基础设施架构	153
第58章	部署	155
第59章	运营和支持	157
第60章	决策日志	159
第61章	问题	161
Part VI	开发生命周期中的软件架构	
第62章	敏捷和架构的冲突：神话还是现实	164
第63章	量化风险	167
第64章	风险风暴	169
第65章	恰如其分的预先设计	173
第66章	初识软件架构	179
第67章	问题	183
Part VII	金融风险系统	
第68章	金融风险系统	186
Part VIII	附录：“技术部落”的软件指南	

• • • • • [\(收起\)](#)

[程序员必读之软件架构_下载链接1](#)

标签

软件架构

架构

软件开发

软件工程

计算机

编程

计算机科学

技术

评论

快速的过了一遍英文版，一般。

开卷有益。。。吧 4C方法还不错，不过整体来说干货不多。。

实用，不啰嗦，恰到好处。特别适合已经有经验的架构师梳理思路。

冠之必读，有些过了，看完这本书，能对架构师的主要工作内容有个大致的了解就足够了。

可能是因为书的前半部分过于说教
而这些说教也都是老生常谈，所以我个人认为这本书的价值主要集中在后半部分。C4
工具还算新颖，可以在具体实践中应用。

内容太水了，像一份提纲

很完整 值得多读几遍

书很好，这是一类书，特点首先是看的快，其次是PPT素材。但是最近不爱看这种书。

整体而言，实用的内容在后面，4C的解构方法和风险分析的模型是亮点……

很好的解答了我一直在问的问题，什么是架构。

非常精准剖析了当前工作困扰。

不够深入 更像是随笔集

作为参考是很不错的书

较浅，点到为止

这种方法论的书看前激动，越看越没兴趣，看完完全忘记！

除了唠叨架构师的概念，其他什么都没有说清楚啊 ...

概览性的小书，内容点很多，但是要融汇贯通不易，需要定期重读。

囫圇吞枣看完，内容稍有点空，也许是经验不够吧

架构入门实践

拼命攒经验

[程序员必读之软件架构_下载链接1](#)

书评

团队里每个人都在做设计，做架构，但从来没人说过什么是架构，该如何架构，这应该是大部分团队的现状。

为什么没人拿架构做为一个明确的主题去讨论，可能的原因是：架构是关于抽象和经验。你说是好的实践，好的架构，最后的落脚点还是交付、性能、可用性上。如果一个的软件满...

这本书的结构大概是这样的： 1.
架构师和程序员是不同的啊！架构师很厉害的啊！架构师也要写代码的啊！ 2.
架构文档要这么写啊朋友！会了没有啊朋友！ 3.
来来来，叔叔手把手教你写架构文档，好好学啊朋友！ 然后，实在没什么意思。
所谓架构师，更多的应该算是成熟期的程序员...

[程序员必读之软件架构_下载链接1](#)