

程序员必读之软件架构

作者：Simon Brown

版权信息

书名：程序员必读之软件架构

作者：Simon Brown

译者：邓钢

ISBN：978-7-115-37107-2

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

[版权声明](#)

[献词](#)

[推荐序一：架构师真正要学会的事情](#)

[1. 要学会去看，然后忘掉](#)

[2. 要学会去听，然后忘掉](#)

[3. 要学会去做，然后忘掉](#)

[4. 要学会超越](#)

[推荐序二](#)

[译者序2.0](#)

[初识软件架构](#)

[怎么会翻译这本书](#)

[架构离我们并不遥远](#)

[周爱民老师的序](#)

[谢谢你们](#)

[序](#)

[软件架构的坏名声](#)

[敏捷愿景](#)

[那么你觉得自己是架构师吗](#)

[失意的架构师](#)

[关于本书](#)

[本书写作初衷](#)

[软件开发的新方法](#)

[关于软件架构，每个开发者都应该知道的五件事](#)

[1. 软件架构不是大型预先设计](#)

[2. 每个软件团队都需要考虑软件架构](#)

[3. 软件架构的角色关乎编码、指导和合作](#)

[4. 无需使用UML](#)

[5. 好的软件架构是支持敏捷开发的](#)

[在微博上分享这本书](#)

[软件架构培训](#)

[Part I 什么是软件架构](#)

[第 1 章 什么是架构](#)

[作为名词](#)

[作为动词](#)

[第 2 章 架构的种类](#)

[它们的共同点是什么](#)

[第 3 章 软件架构是什么](#)

[应用程序架构](#)

[系统架构](#)

[软件架构](#)

[企业架构：战略而非代码](#)

[第 4 章 敏捷软件架构是什么](#)

[理解“敏捷”](#)

[好的架构带来敏捷](#)

[你需要有多敏捷](#)

[第 5 章 架构对上设计](#)

[找出区别](#)

[理解意义](#)

[第 6 章 软件架构重要吗](#)

[缺乏软件架构将引发问题](#)

[软件架构的好处](#)

[所有软件项目都需要软件架构吗](#)

[第 7 章 问题](#)

[Part II 软件架构的角色](#)

[第 8 章 软件架构的角色](#)

[1. 架构驱动力](#)

[2. 设计软件](#)

[3. 技术风险](#)

[4. 架构演化](#)

[5. 编写代码](#)

[6. 质量保证](#)

[合作或失败](#)

[技术领导是一个角色而非级别](#)

[提出你自己对这个角色的定义](#)

[第 9 章 软件架构师应该编码吗](#)

[编写代码](#)

[构建原型、框架和基础](#)

[进行代码评审](#)

[实验并与时俱进](#)

[软件架构师和雇主之间的矛盾](#)

[你不必放弃编码](#)

[不要把全部时间都用于编码](#)

[第 10 章 软件架构师应该是建造大师](#)

[联盟的状态](#)

[回顾过去](#)

[建造大师真的会建造吗](#)

[象牙塔](#)

[建造大师角色的差异](#)

[实现角色](#)

[架构师要和团队一起工作](#)

[第 11 章 从开发者到架构师](#)

[经验是一个好的评价标准，但你需要看得更深](#)

[模糊的界线](#)

[跨越界线是我们的责任](#)

[第 12 章 拓展T](#)

[进一步的技术能力](#)

[知识面宽](#)

[软件架构师是通才型专家](#)

[软件架构是技术活](#)

[第 13 章 软技能](#)

[保持积极](#)

[第 14 章 软件架构不是接力运动](#)

[解决方案架构师](#)

[要有人负责大局](#)

[第 15 章 软件架构要引入控制吗](#)

[提供指导，追求一致性](#)

[控制的程度](#)

[控制因文化而不同](#)

[操纵杆，而非按钮](#)

[第 16 章 小心鸿沟](#)

[开发者关注底层细节](#)

[闭门造车的独裁架构师](#)

[拉近距离](#)

[如果你是软件架构师](#)

[如果你是软件开发者](#)

[软件架构的合作方式](#)

[第 17 章 未来的软件架构师在哪里](#)

[指导、辅导和师徒关系](#)

[我们正在失去技术导师](#)

[软件团队需要休息](#)

[第 18 章 每个人都是架构师，除非他们有其他身份](#)

[每个人都是架构师](#)

[除非他们有其他身份](#)

[敏捷需要架构吗](#)

[第 19 章 软件架构咨询师](#)

[领域知识](#)

[权威](#)

[第 20 章 问题](#)

[Part III 设计软件](#)

[第 21 章 架构驱动力](#)

[功能需求](#)

[质量属性](#)

[约束](#)

[原则](#)

[理解影响](#)

[第 22 章 质量属性（非功能需求）](#)

[哪些对你重要](#)

[第 23 章 处理非功能需求](#)

[捕捉](#)

[提炼](#)

[挑战](#)

[第 24 章 约束](#)

[时间和预算的约束](#)

[技术约束](#)

[人员约束](#)

[组织约束](#)

[约束都是不好的吗](#)

[约束可以划分优先级](#)

[倾听约束](#)

[第 25 章 原则](#)

[开发原则](#)

[架构原则](#)

[谨防最佳实践](#)

[第 26 章 技术不是实现细节](#)

[你有复杂的非功能需求吗](#)

[你有约束吗](#)

[你有一致性吗](#)

[推迟与解耦](#)

[每个决策都是权衡](#)

[第 27 章 更多分层等于更高复杂度](#)

[非功能需求](#)

[时间和预算：没有什么是免费的](#)

[第 28 章 协同设计是一把双刃剑](#)

[经验影响软件设计](#)

[第 29 章 软件架构是对话的平台](#)

[软件开发不只是交付特性](#)

[第 30 章 SharePoint项目也需要软件架构](#)

[很多SharePoint实现都不只是SharePoint](#)

[非功能性需求仍然适用于SharePoint解决方案](#)

[SharePoint项目很复杂，也需要技术领导力](#)

[SharePoint解决方案仍然需要编写文档](#)

[强大的领导力和纪律不只是针对软件开发项目](#)

[第 31 章 问题](#)

[Part IV 可视化软件](#)

[第 32 章 沟通障碍](#)

[抛弃UML](#)

[敏捷需要良好的沟通](#)

[第 33 章 对草图的需要](#)

[测试驱动开发与图表](#)

[为什么人们应该学习如何画草图](#)

[画草图不是艺术](#)

[画草图不是综合模型](#)

[画草图可以是协作活动](#)

[第 34 章 无效的草图](#)

[采购清单](#)

[只有框没有线](#)

[“功能视图”](#)

[航线图](#)

[一般正确](#)

[推迟技术](#)

[部署和执行上下文](#)

[太多假设](#)

[无家可归的C#对象 \(HOCO\)](#)

[选择你自己的冒险](#)

[应该用白板](#)

[创建有效的草图](#)

[第 35 章 C4: 语境、容器、组件和类](#)

[通用的抽象集合](#)

[总结软件的静态视图](#)

[通用标记法的通用抽象](#)

[图应该简单且脚踏实地](#)

[第 36 章 语境图](#)

[意图](#)

[结构](#)

[用户、演员、角色、人物等](#)

[IT系统](#)

[交互](#)

[动机](#)

[受众](#)

[示例](#)

[第 37 章 容器图](#)

[意图](#)

[结构](#)

[容器](#)

[交互](#)

[系统边界](#)

[动机](#)

[受众](#)

[示例](#)

[第 38 章 组件图](#)

[意图](#)

[结构](#)

[组件](#)

[交互](#)

[动机](#)

[受众](#)

[示例](#)

[第 39 章 是否包含技术选择](#)

[在设计过程中绘图](#)

[回顾性绘图](#)

[架构图应该概念化](#)

[明确技术选择](#)

[第 40 章 你会那样编码吗](#)

[共享组件](#)

[分层策略](#)

[图应该反映现实](#)

[第 41 章 软件架构和编码](#)

[职责驱动设计和组件分解](#)

[我们谈论组件但编写类](#)

[用层封装代码](#)

[用特性封装](#)

[用组件封装](#)

[对齐软件架构和代码](#)

[第 42 章 你不需要UML工具](#)

[有很多类型的UML工具](#)

[既有效又简单](#)

[UML的用途](#)

[没有银弹](#)

[第 43 章 有效的草图](#)

[标题](#)

[标签](#)

[形状](#)

[职责](#)

[线条](#)

[颜色](#)

[边框](#)

[布局](#)

[方向](#)

[要点](#)

[图表的评审清单](#)

[倾听问题](#)

[第 44 章 C4的常见问题](#)

[语境图上的系统名称](#)

[混合的抽象层次](#)

[共享组件](#)

[工具组件](#)

[从IT的角度勾画企业语境](#)

[第 45 章 问题](#)

[Part V 为软件生成文档](#)

[第 46 章 代码不会讲述完整的故事](#)

[代码未描绘的设计意图](#)

[辅助信息](#)

[第 47 章 软件文档即指南](#)

[1. 地图](#)

[2. 景色](#)

[3. 历史和文化](#)

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《程序员必读之软件架构》Simon Brown 著.epub

请登录 <https://shgis.cn/post/316.html> 下载完整文档。

手机端请扫码查看：

