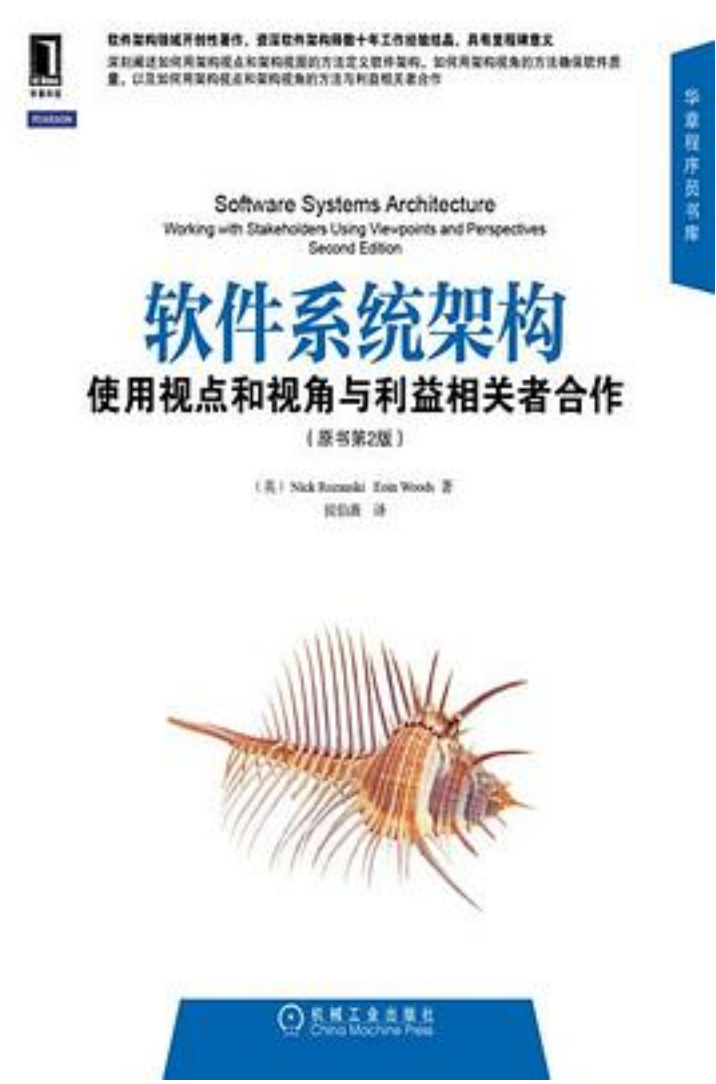


软件系统架构



[软件系统架构_下载链接1](#)

著者:Nick Rozanski

出版者:机械工业出版社

出版时间:2013-5

装帧:

isbn:9787111421863

鲁然斯基等编著的《软件系统架构(使用视点和视角与利益相关者合作原书第2版)》是软件系统架构领域的开创性著作，是两位拥有数十年软件行业工作经验的架构师工作经验的结晶，围绕利益相关者、视点和视角三大主题，创新性地提出了如何用架构视点和架构视图的方法来定义软件架构，如何用架构视角的方法来确保软件质量，以及如何用架构视点和架构视角的方法与利益相关者合作，具有里程碑意义。《软件系统架构(使用视点和视角与利益相关者合作原书第2版)》还展示了一种实用的、经过验证的框架，你可以应用它来处理架构定义过程，并应对创建软件架构工作所带来的挑战。

《软件系统架构(使用视点和视角与利益相关者合作原书第2版)》分为五个部分，共30章。第一部分（第1~5章）阐释利益相关者、架构描述、视点、视图和视角等基本概念，并描述软件架构师的角色；第二部分（第6~14章）描述作为架构师所从事的重要活动，如协商项目的范围、识别并管理利益相关者、使用场景和模式、创建模型以及为架构创建文档并对其加以验证等；第三部分（第15~23章）集合了在创建架构描述时最重要的七种视点：情境、功能、信息、并发、开发、部署和运维视点；第四部分（第24~29章）集合了对于信息系统最重要的视角，包括安全性、性能和可伸缩性、可用性和适应性、演进、位置、开发资源、国际化等；第五部分（第30章）把这些概念融合在一起，并阐释了如何把这些理论应用到实践中。

海报：

作者介绍：

目录: 译者序

前言

第1版前言

第1章 简介 1

1.1 利益相关者、视点和视角 1

1.2 本书结构 4

1.3 谁应该阅读本书 5

1.4 本书约定 5

第一部分 架构的基本原则

第2章 软件架构概念 8

2.1 软件架构 8

2.1.1 系统元素和关系 8

2.1.2 基本系统属性 9

2.1.3 设计和发展的原则 10

2.1.4 系统属性和内部组织形式 10

2.1.5 软件架构的重要性 13

2.2 架构元素 13

2.3 利益相关者 14

2.3.1 个人、团队或组织 14

2.3.2 兴趣和关注点 15

2.3.3 利益相关者的重要性 16

2.4 架构描述 16

2.5 核心概念之间的关系 17

2.6 小结 18

2.7 延伸阅读 19

第3章 视点和视图 20

3.1 架构视图 22

3.2 视点 23

3.3 核心概念之间的关系 24

3.4 使用视点和视图的好处 24

3.5 视点缺陷 25

3.6 视点目录	25
3.7 小结	27
3.8 延伸阅读	28
第4章 架构视角	29
4.1 质量属性	29
4.2 架构视角	30
4.3 向视图应用视角	33
4.4 应用视角的结果	34
4.4.1 深入的观点	34
4.4.2 提升	35
4.4.3 精品内容	35
4.5 核心概念之间的关系	35
4.6 使用视角的好处	36
4.7 视角的缺陷	37
4.8 视角与视点对比	37
4.9 视角种类	38
4.10 小结	39
4.11 延伸阅读	39
第5章 软件架构师的角色	41
5.1 架构定义过程	41
5.1.1 架构定义不仅是设计	42
5.1.2 需求分析和架构定义之间的区别	43
5.1.3 架构定义和设计之间的区别	43
5.2 架构师的角色	44
5.3 核心概念之间的相互关系	46
5.4 架构专门化	47
5.5 组织情境	47
5.5.1 业务分析师	47
5.5.2 项目经理	47
5.5.3 设计主管	48
5.5.4 技术专家	49
5.5.5 开发者	49
5.6 架构师的技能	49
5.7 架构师的责任	50
5.8 小结	51
5.9 延伸阅读	51
第二部分 软件架构过程	
第6章 软件架构过程简介	54
第7章 架构定义过程	55
7.1 指导原则	55
7.2 过程产出物	56
7.3 过程情境	56
7.4 支持活动	57
7.5 架构定义活动	60
7.6 过程完成标准	62
7.7 软件开发生命周期中的架构定义	64
7.7.1 瀑布式方法	64
7.7.2 迭代方法	65
7.7.3 敏捷方法	65
7.8 小结	66
7.9 延伸阅读	67
第8章 关注点、原则和决定	68
8.1 专注于问题的关注点	70
8.1.1 业务策略	70

8.1.2 业务目标和驱动力	70
8.1.3 系统范围和需求	71
8.1.4 业务标准和政策	72
8.2 专注于解决方案的关注点	72
8.2.1 IT策略	72
8.2.2 技术目标和驱动力	72
8.2.3 技术标准和政策	73
8.3 其他现实世界中的约束	73
8.4 什么决定了好的关注点	75
8.5 架构原则	75
8.5.1 什么造就了好的原则	78
8.5.2 定义自己的原则	78
8.6 架构决定	79
8.7 使用原则关联关注点和决定	81
8.8 检查列表	82
8.9 小结	83
8.10 延伸阅读	83
第9章 确定并引入利益相关者	84
9.1 利益相关者的选择	84
9.2 利益相关者的类别	85
9.2.1 出资方	86
9.2.2 评估者	86
9.2.3 沟通者	86
9.2.4 开发人员	87
9.2.5 维护人员	87
9.2.6 生产工程师	87
9.2.7 供应商	87
9.2.8 支持人员	87
9.2.9 系统管理员	88
9.2.10 测试人员	88
9.2.11 用户	88
9.3 示例	88
9.3.1 非专门设计的部署项目	88
9.3.2 软件产品开发项目	89
9.3.3 合作开发	89
9.4 代理利益相关者	90
9.5 利益相关者组	90
9.6 利益相关者的责任	90
9.7 检查列表	91
9.8 小结	91
9.9 延伸阅读	92
第10章 识别并使用场景	93
10.1 场景类型	93
10.2 使用场景	94
10.3 识别场景并排定优先级	95
10.4 捕获场景	96
10.5 什么造就了好场景	98
10.6 应用场景	98
10.6.1 纸质模型	98
10.6.2 走查	99
10.6.3 模拟	100
10.6.4 原型实现的测试	100
10.6.5 完整规模真实测试	100
10.7 有效使用场景	100

- 10.7.1 识别一系列重点场景 101
- 10.7.2 使用清晰的场景 101
- 10.7.3 尽早使用场景 101
- 10.7.4 包含对系统质量场景的使用 101
- 10.7.5 包含对故障场景的使用 101
- 10.7.6 让利益相关者紧密参与 101
- 10.8 检查列表 102
- 10.9 小结 102
- 10.10 延伸阅读 103
- 第11章 使用样式和模式 104
- 11.1 设计模式介绍 104
- 11.2 样式、模式和惯用法 105
- 11.2.1 架构样式 106
- 11.2.2 软件设计模式 106
- 11.2.3 语言惯用法 106
- 11.2.4 使用样式、模式和惯用法 107
- 11.3 模式和架构策略 107
- 11.4 架构样式的例子 108
- 11.5 使用架构样式的好处 110
- 11.6 样式和架构描述 111
- 11.7 应用设计模式和语言惯用法 111
- 11.8 检查列表 113
- 11.9 小结 113
- 11.10 延伸阅读 113
- 第12章 创建架构模型 115
- 12.1 模型为什么重要 115
- 12.2 模型的类型 117
- 12.2.1 定性模型 117
- 12.2.2 定量模型 118
- 12.2.3 示意图 119
- 12.3 建模语言 119
- 12.3.1 架构描述语言 119
- 12.3.2 统一建模语言 120
- 12.3.3 可执行的领域专用语言 121
- 12.3.4 其他建模语言 121
- 12.4 创建有效模型的准则 121
- 12.4.1 有目的地建模 121
- 12.4.2 应对受众 122
- 12.4.3 仔细、准确地抽象 122
- 12.4.4 根据风险确定工作重点 123
- 12.4.5 选择描述性的名称 123
- 12.4.6 定义你的术语 123
- 12.4.7 以简单为目标 124
- 12.4.8 使用已定义的标记法 124
- 12.4.9 了解暗示的语义 124
- 12.4.10 验证模型 125
- 12.4.11 保持模型的活力 125
- 12.5 和敏捷团队一起建模 125
- 12.6 检查列表 126
- 12.7 小结 127
- 12.8 延伸阅读 127
- 第13章 创建架构描述 128
- 13.1 有效架构描述的属性 129
- 13.1.1 正确 129

13.1.2	充分	129
13.1.3	及时	130
13.1.4	简洁	131
13.1.5	清晰	131
13.1.6	最新	132
13.1.7	精确	133
13.2	词汇表	134
13.3	ISO标准	134
13.4	架构描述的内容	135
13.4.1	文档控制	135
13.4.2	内容表	135
13.4.3	介绍和管理纲要	135
13.4.4	利益相关者	136
13.4.5	通用架构原则	136
13.4.6	架构设计决定	136
13.4.7	视点	136
13.4.8	视图	136
13.4.9	质量属性摘要	137
13.4.10	重要的方案	137
13.4.11	亟待解决的问题	137
13.4.12	附录	138
13.5	展现架构描述	138
13.6	检查列表	139
13.7	小结	140
13.8	延伸阅读	140
第14章	评估架构	141
14.1	为什么要评估架构	141
14.2	评估技术	142
14.2.1	演讲	142
14.2.2	正式评审和结构化的走查	143
14.2.3	通过使用场景来评估	144
14.2.4	原型和概念验证系统	145
14.2.5	骨架系统	146
14.3	基于场景的评估方法	146
14.3.1	以架构为中心的活动	147
14.3.2	以利益相关者为中心的活动	149
14.4	在软件生命周期内评估	150
14.5	验证现存系统的架构	151
14.6	记录评估结果	153
14.7	选择评估方法	154
14.8	检查列表	154
14.9	小结	155
14.10	延伸阅读	155
第三部分	视点类型	
第15章	视点类型简介	158
第16章	情境视点	160
16.1	关注点	161
16.1.1	系统范围和责任	161
16.1.2	外部实体和服务以及所用数据的标识	161
16.1.3	外部实体的本质和特征	162
16.1.4	外部接口的标识和职责	162
16.1.5	外部接口的本质和特征	163
16.1.6	其他外部依赖关系	163
16.1.7	对系统环境的影响	164

- 16.1.8 总体完成度、一致性和连贯性 164
- 16.1.9 利益相关者的关注点 165
- 16.2 模型 165
 - 16.2.1 情境模型 165
 - 16.2.2 交互场景 169
- 16.3 问题和缺陷 169
 - 16.3.1 遗漏或者错误的外部实体 169
 - 16.3.2 遗漏隐藏的依赖关系 169
 - 16.3.3 松散或不精确的接口描述 170
 - 16.3.4 详细程度不合适 170
 - 16.3.5 范围蔓延 170
 - 16.3.6 隐藏或假设的情境和范围 171
 - 16.3.7 过于复杂的交互 171
 - 16.3.8 过度使用术语 171
- 16.4 检查列表 172
- 16.5 延伸阅读 172
- 第17章 功能视点 173
 - 17.1 关注点 173
 - 17.1.1 功能能力 173
 - 17.1.2 外部接口 174
 - 17.1.3 内部结构 174
 - 17.1.4 功能设计哲学 174
 - 17.1.5 利益相关者的关注点 175
 - 17.2 模型 176
 - 17.3 问题和缺陷 184
 - 17.3.1 设计很差的接口 184
 - 17.3.2 难以理解的职责 184
 - 17.3.3 基础架构作为功能性元素 184
 - 17.3.4 过载的视图 185
 - 17.3.5 没有元素定义的图 186
 - 17.3.6 难以调节多位利益相关者的需求 186
 - 17.3.7 错误的详细程度 187
 - 17.3.8 “神元素” 187
 - 17.3.9 过多依赖关系 188
 - 17.4 检查列表 188
 - 17.5 延伸阅读 188
- 第18章 信息视点 190
 - 18.1 关注点 191
 - 18.1.1 信息结构和内容 191
 - 18.1.2 信息目的和用途 191
 - 18.1.3 信息所有权 192
 - 18.1.4 企业拥有的信息 193
 - 18.1.5 标识符和映射关系 194
 - 18.1.6 信息语义的易变性 195
 - 18.1.7 信息存储模型 196
 - 18.1.8 信息流 197
 - 18.1.9 信息一致性 198
 - 18.1.10 信息质量 199
 - 18.1.11 及时性、延迟和寿命 200
 - 18.1.12 归档和保留信息 201
 - 18.1.13 利益相关者的关注点 201
 - 18.2 模型 202
 - 18.2.1 静态信息结构模型 202
 - 18.2.2 信息流模型 204

- 18.2.3 信息生命周期模型 206
- 18.2.4 其他类型的信息模型 207
- 18.3 问题和陷阱 209
 - 18.3.1 数据展现不兼容 209
 - 18.3.2 不可避免的多个更新器 210
 - 18.3.3 键值匹配缺陷 211
 - 18.3.4 接口复杂 211
 - 18.3.5 过载的中心数据库 212
 - 18.3.6 不一致的分布式数据库 213
 - 18.3.7 信息质量很差 213
 - 18.3.8 信息延迟过大 213
 - 18.3.9 容量不足 214
- 18.4 检查列表 214
- 18.5 延伸阅读 215
- 第19章 并发视点 216
 - 19.1 关注点 217
 - 19.1.1 任务结构 217
 - 19.1.2 功能元素与任务的映射关系 218
 - 19.1.3 进程间通信 218
 - 19.1.4 状态管理 218
 - 19.1.5 同步和整合 218
 - 19.1.6 支持可伸缩性 219
 - 19.1.7 启动和关闭 219
 - 19.1.8 任务故障 219
 - 19.1.9 重入 219
 - 19.1.10 利益相关者的关注点 220
 - 19.2 模型 220
 - 19.2.1 系统级别的并发模型 220
 - 19.2.2 状态模型 225
 - 19.3 问题和缺陷 228
 - 19.3.1 对错误的并发建模 228
 - 19.3.2 错误地对并发建模 228
 - 19.3.3 过度复杂 229
 - 19.3.4 资源竞争 229
 - 19.3.5 死锁 230
 - 19.3.6 竞争条件 230
 - 19.4 检查列表 230
 - 19.5 延伸阅读 231
 - • • • • (收起)

[软件系统架构 下载链接1](#)

标签

软件架构

架构

计算机

软件工程

软件开发

计算机科学

计算机理论

架构设计

评论

浙江图书馆

内容还算不错，但是自我感觉这本书可读性不强，很刻板的感觉。

一刷。
书中没有想用一个大的模型来解决所有的问题。而是首先引入了利益相关者（为之建立系统）、视点（模型）和视角（质量属性）的概念，接着结合不同的视点和视角来描述系统，需要保证各个视点之间的一致性。一即多，多即一。
PS.书中的概念和定义较多。

内容太干太“硬”，加上期间断断续续，历时半年多，今天终于读完，算的上旷日持久了。

不知道是不是翻译的原因，这本书读起来比较生硬，可读性不是太好；很多关键点如果没有心得的话读了帮助也不大，对于长期思考架构和有所观察的人来说一定程度上能够帮助梳理思路

[软件系统架构_下载链接1](#)

书评

系统架构的视图，视角，利益相关者的概念早已有之，该书对相关理论进行了全面的总结和论述，相信读过其他软件架构方面书籍的人士，对本书中的概念不会陌生。文中提出七个视图，五个视角。视图是对整个系统一个侧面的反映，视角是从某个领域角度观察整个系统，涉及多个...

[软件系统架构_下载链接1](#)