

《软件架构》

图书基本信息

书名：《软件架构》

13位ISBN编号：97871111101666

10位ISBN编号：7111101669

出版时间：2002-8

出版社：第1版 (2002年1月1日)

作者：Dikel

页数：196

译者：张恂

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《软件架构》

内容概要

本书主要描述软件架构与软件组织之间的相互关系，依次介绍了作者根据多年管理经验和研究总结出的软件架构组织的VRAPS 5项原则——构想(Vision)、节奏(Rhythm)、预见(Anticipation)、协作(Partnering)和简化(Simplification)，并通过案例分析、模式和反模式展示了如何运用这一模型。本书的主要读者为软件企业的管理人员、开发人员和软件产品的客户等，也可作为大学计算机及相关专业的本科生、研究生和教师的参考用书。

《软件架构》

作者简介

张恂，1972年生。东南大学计算机软件硕士，国内某自主研发的分布式对象通信中间件的主要开发人员，曾在国内一家著名移动通信设备厂负责一个大型通信系统软件研发项目的管理近两年，后任某民营软件公司的CTO，目前主要从事软件工程和面向对象技术咨询。

书籍目录

译者序

前言

第1章 无形的帮助

1.1 本书主题

1.1.1 软件架构愈显重要

1.1.2 对于某些人“他们是受益人”的消息来得太晚了

1.2 原则揭示本质

1.3 实践组织原则：架构师的新任务

1.3.1 节奏

1.3.2 构想

1.3.3 简化与预见

1.3.4 协作

1.4 VRAPS原则在Web领域中的应用

1.5 小结

第2章 VRAPS参考模型：各组成部分的关系

2.1 概述

2.1.1 模型的重要性

2.1.2 VRAFS模型

2.2 应用环境

2.3 软件架构的组织原则

2.4 概念框架

2.4.1 准则

2.4.2 模式

2.4.3 反模式

2.5 使用VRAPS模型

2.6 VRAPS模型演变过程

2.7 小结

第3章 形成并统一构想

3.1 概述

3.2 构想定义

3.2.1 把价值映射为架构约束

3.2.2 一致性与灵活性

3.3 构想挑战

3.3.1 架构师影响力的局限

3.3.2 高级经理与架构师的合作

3.3.3 产品线加剧了架构师和高级经理面临的挑战

3.3.4 识别构想的瓦解

3.4 形成构想

3.4.1 谁是真正的架构师

3.4.2 构想与领导

3.4.3 缺乏尊重

3.5 将构想原则付诸实践：准则、反模式与模式

3.5.1 准则1：架构师的构想与其发起人、用户和最终客户期望实现的目标保持一致

3.5.2 准则2：实施人员信任并使用架构

3.5.3 准则3：关于架构和构件的潜藏知识对其用户是可见的和可获得的

3.6 小结

3.7 其他可用的模式与反模式

第4章 节奏：保证节拍、过程与进展

4.1 概述

4.2 节奏定义

4.3 动因

4.3.1 节奏帮助移交管理

4.3.2 节奏驱动活动完结

4.4 将节奏原则付诸实践：准则、反模式与模式

4.4.1 准则1：经理们定期地再评估、同步和调整架构

4.4.2 准则2：架构用户对架构发布的进度和内容具有高度的信心

4.4.3 准则3：通过节奏协调明确的活动

4.5 小结

4.6 其他可用的模式与反模式

第5章 预见：预测、验证与调整

5.1 概述

5.2 预见定义

5.2.1 预测

5.2.2 验证

5.2.3 调整

5.3 预见应用

5.3.1 朝多个方向发展架构

5.3.2 架构客户与他们的客户

5.3.3 目标离现实太远

5.3.4 目标离现实太近

5.3.5 平衡现在和未来的需求

5.3.6 掌握平衡

5.4 将预见原则付诸实践：准则、反模式与模式

5.4.1 准则1：不断增强架构的能力以响应预见到的风险和架构客户及其客户的需求；市场驱动的标准和演变的技术；战略性业务方向的改变

5.4.2 准则2：通过快速复审和开发周期，评估技术和业务上的风险与机会

5.4.3 准则3：当发现关键的估计或假设有错时，及时调整功能特性、预算、计划或进度

5.5 小结

5.6 其他可用的模式与反模式

第6章 协作：建立合作型组织

6.1 概述

6.2 协作定义

6.2.1 架构受益人

6.2.2 明确、合作的角色

6.2.3 价值最大化

6.3 产业基础

6.3.1 契约管理

6.3.2 网络化组织

6.3.3 价值链

6.3.4 信任

6.4 将协作原则付诸实践：准则、反模式与模式

6.4.1 准则1：架构师不断地努力了解谁是最关键的受益人，他们如何贡献价值，以他们需要什么

6.4.2 准则2：受益人之间达成明确和强制性的契约

6.4.3 准则3：通过社会行为制度和非正式规范强化合作

6.5 小结

6.6 其他可用的模式与反模式

第7章 简化：澄清与最小化

7.1 概述

7.2 简化定义

7.2.1 Conway定律

7.2.2 澄清

7.2.3 最小化

7.3 将简化原则付诸实践：准则、反模式与模式

7.3.1 准则1：开发人员长期不断地使用架构，减少了总成本和复杂性

7.3.2 准则2：架构小组明确理解关键最小需求并且将其构造成多应用共享的核心元素

7.3.3 准则3：长期预算和行动确保当发生以下情况时把相关元素从核心移走：1)它们没有被共享或者增加了不必要的复杂性；2)有明确的业务理由

7.4 小结

7.5 其他可用的模式与反模式

第8章 原则实践：Allaire公司案例分析

8.1 简介

8.1.1 为什么选择Allaire公司

8.1.2 5项组织原则

8.1.3 我们采用的方法

8.1.4 关于结果

8.2 构想--把好的构想变成现实

8.2.1 定义和描述

8.2.2 形成架构构想并保持其生命力的实践方法

8.2.3 Allaire职员发现的警告信号

8.3 节奏--保持节拍

8.3.1 定义和描述

8.3.2 帮助一个架构组织保持同步的实践方法

8.3.3 Allair职员发现的警告信号

8.4 预见--预测、验证与调整

8.4.1 定义和描述

8.4.2 保持架构"与未来磨合"的实践方法

8.4.3 Allaire职员发现的警告信号

8.5 协作--生命线

8.5.1 定义和描述

8.5.2 支持协作的实践方法

8.5.3 Allaire职员发现的警告信号

8.6 简化--找到要素

8.6.1 定义和描述

8.6.2 支持简化的实践方法

8.6.3 Allaire职员发现的警告信号

8.7 小结

第9章 案例分析：用VRAPS建立和实现基准

9.1 概述

9.2 基准测试提供了一个框架

9.2.1 调查模板

9.2.2 组织背景与环境模板

9.2.3 架构概况与投资回报率模板

9.2.4 原则模板

9.2.5 实践方法模板

9.3 我们如何进行基准测试

9.3.1 获得一个可行的构想

9.3.2 进行访谈

9.4 基准测试结果与体会

9.4.1 原则共鸣

9.4.2 原则关系

9.4.3 几点体会

9.5 小结

附录A 快速参考表：原则、准则、反模式与模式

附录B 反模式和模式一览

参考文献

索引

精彩短评

- 1、组织结构决定软件架构！
- 2、VRAPS关注于实现和发展软件架构的组织方面
- 3、软件架构绝不是纯技术，她同软件组织是密切相关的
- 4、明白到软件架构的设计，其实就是设计一条可以持续适应市场需求的生产线。剩下的，看附录A和附录B的内容就差不多了。作者用模式和反模式的写法并不见得是一个很好的主意。
- 5、从几个方面讨论了注意事项：目标规划，进度管理，团队协同。

章节试读

1、《软件架构》的笔记-第196页

目标规划：多方整合。

客观评价软件架构的质量和稳定性，以及目标的可行性。

整合客户价值、产品构想和技术创新

分享知识，推动技术合作。

进度管理：有节奏的进度有益于团队开发与协作

利用关键特性，进行项目进度安排

与协作特性同步，加速架构技术的推广

过程管理：

技术原型引领，及时复审实施结果

重构或启用替代方案，纠正错误假设。

团队协作：信任，但是要验证

利用价值链，识别受益人。

形成明确的契约，合理利用非正式人际关系。

架构演进：

为其他开发人员建立迁移的途径

监控架构核心的复杂性

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com