

[illegible]

申峻嵘 黄星琪 王 帅 等译
王千祥 审校

软件工程丛书

软件架构师教程

Software Architect Bootcamp

[美] Raphael Malveau Thomas J. Mowbray 著

申峻嵘 黄星琪 王 帅 等译
王千祥 审校

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书阐述了作为一个软件架构师所需要具备的基本技能,并按照军事训练的一般程序,将这些技能划分为:基础训练、准备战斗、训练学校、伞兵学校、情报操作等阶段,并对每个阶段需要掌握的技术进行了扼要介绍。除了关注软件架构师的技术性能力之外,本书还对软件行业中有关领导素质、心理素质等涉及到人性和心理等社会性的问题进行了探讨,以更好地指导软件架构师的职业生涯。

本书面向专业的软件开发人员和软件架构师,也可作为软件学院的教材或教学参考书。

Authorized Translation from the English language edition, entitled Software Architect Bootcamp, 1st Edition by Raphael Malveau, Thomas J. Mowbray ISBN 0130274070, published by Pearson Education, Inc., publishing as Prentice Hall PTR, copyright © 2001 Prentice Hall PTR.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2003

本书中文简体版专有出版权由 Pearson Education 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号:图字:01-2002-2742

图书在版编目(CIP)数据

软件架构师教程/(美)马维(Malveau, R.)著;申峻嵘等译. —北京:电子工业出版社, 2003.10

(软件工程丛书)

书名原文:Software Architect Bootcamp

ISBN 7-5053-9204-2

I. 软… II. ①马…②申… III. 软件设计—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2003)第 088529 号

责任编辑:张 毅 zhangyi@phei.com.cn

印 刷:北京兴华印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×980 1/16 印张:20.75 字数:304 千字

版 次:2003 年 10 月第 1 版 2003 年 10 月第 1 次印刷

印 数:6 000 册 定价:38.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

本书的英文名是“Software Architect Bootcamp”，直译过来是“软件架构师新兵集训营”，文如其名，两位作者以介绍军事训练的笔调，阐述了成为一名成功的软件架构师所需要掌握的知识、技能和方法，以及所需要具备的素质和态度。

本书的作者 Malveau 和 Mowbray 博士是全球软件架构师协会的创始人，他们也是有着丰富经验的软件架构师和咨询专家，曾著有《反模式 (Anti Pattern)》等多部有影响力的计算机书籍。

作为一本教程，自然要做到“授之以鱼”，现今市面上绝大部分技术书籍即属此类；而要撰写一本“授之以渔”的书籍，则并非易事。本书从内容上看，大致可以分为两部分。一方面介绍了软件架构师所必备的智力工具，即技术、知识、方法、过程等，文中涉及的概念非常丰富，介绍的技术也相当实用，他们是作者赠给读者的“鱼”。技术的发展一日千里，任何一种技术，都不可避免有过时的一天；任何一本技术书籍，从撰写的那一刻起就注定是陈旧的。所幸，本书并没有受繁复技术的羁绊，相反，作者高屋建瓴地抓住了隐藏在技术之后的原理、背景、发展脉络和前景等“不变”的内容，而避开了“变化”迅速的技术细节，娓娓道来，从而给读者提供了一个完整且清晰的技术观。另一方面，作者则在“渔”上下了很大的功夫，他们将一个架构师需要具备的基本技能划分为：基础训练、准备战斗、训练学校、伞兵学校、情报操作等阶段，对每个阶段需要掌握的技术扼要介绍，点到为止，却给出了如何获得相关资料和怎样掌握这些技术的方法。更值得一提的是，除了关注技能，作者还将领导素质、心理素质等涉及到人性和心理等社会性问题引入了这本技术书籍中，并以自己 20 余年的实际经验现身说法，真挚、亲切而诚恳地给读者以启迪和指导，这是极为难能可贵的。

当然，本书也有值得商榷之处，比如对于作者熟悉的 CORBA 和 RM-ODP 介绍颇多，而对 J2EE、.NET 等新兴技术则少有提及，这也从

另一个角度印证了“技术书籍永远是过时的”的观点。

本书的翻译出版旨在填补国内软件架构师教程的空白，也希望能为国内软件工程领域的研究、教学、管理和技术人员提供一份参考文献和实践指南。

参加本书翻译工作的译者有：黄星琪（第3、6章）、王帅（第2、5章）、罗景（第8、9章）、伊凯（第4章）、刘春风（第7章）、申峻嵘（其余部分，统稿）；王千祥老师审阅并校订了全书。

由于译者自身的知识局限，译稿中难免会出现错漏之处，为此，谨向读者和原书作者致以歉意。

最后，感谢电子工业出版社对本书翻译工作的关注和支持，并给予了最大的方便，使得我们在一种愉快和轻松的气氛中合作。同时，也感谢本书的两位作者，他们在与译者 E-mail 交流时的耐心、谦逊和诚恳令人感动。

读好书，如饮醇酒，回味无穷。

愿与读者共品味之。

译者

启 示

我们知道的与不知道的相比，只是很小的一部分。

——印度谚语

谨以此书献给我们的家人：

Carrie Malveau

和

Kate Mowbray, CPA

序 言

对于计算机专业人员而言，软件体系结构（Software Architecture）是一门新兴学科，同时也是一条令人兴奋的职业道路。无论你是新手还是经验丰富的从业人员，我们都鼓励你阅读本书，以成为一名更好的软件架构师。你也许已经注意到了，软件体系结构在当前多数软件图书中都未详述，因此，在本书中，我们集中讨论了该领域的必要知识，以使你成为最具效率的软件架构师。

我们两个合著者在各自的公司里都经历了从“第一线的技术人员”逐渐升迁为公司最高层软件架构师的职业生涯。我们是设计人员而非管理者，同时也热爱我们工作的技术性质。在彼此的公司中，我们享有同公司高级经理相当的薪金和福利。换句话说，我们都对选择软件体系结构事业而获得的结果心满意足。我们认为许多读者将会从我们的经历中有所收获——于是本书诞生了。

那么，这仅仅是一本讲述软件体系结构的书吗？不！它还是一本实用手册，你可以从中得到训练。我们将通过模拟军队中实战训练的方式来组织全书，因为这种风格体现了学习本门课程所必需的态度。作为一名架构师，你需要多方面的“生存”技能，这些技能包括技术、政治、个人等多个层面。虽然两位作者中无一有过从军经历，我们仍然在许多方面发现软件体系结构类似于战场：

这是一个创意的战场，开发者们争先恐后地推广自己的概念。

这也是一个争夺控制权的战场，经理和开发者们明争暗斗，期望取得对关键设计决策上的控制权。

这还是一个危机四伏的战场，因为相对于绝大多数经理和个体开发者而言，软件架构师承担着更大范围内的技术和流程上的风险。

如果你是一位正在实践的软件架构师，可以想像你会终日忙碌。那么建议你在购买本书之后，细读一下目录以及你觉得新鲜的话题。

首先重点阅读相关章节，然后等你空闲时再通读全书，从而全面掌握本书的内容。

如果你刚刚接触软件体系结构，但又想成为一名软件架构师的话，我们建议你从第 1 章开始，一页一页地仔细阅读本书，同时，认真完成每章后面提供的练习，这将有助于你在学习增加经验。

Raphael Malveau

Thomas J. Mowbray

于美国维吉尼亚州，McLean

致 谢

我们要衷心感谢为本书的写作提供慷慨支持的朋友和做出技术贡献的同事们。在他们之中，我们要特别感谢：Jan Putman, Kirstie Bellman, Liz Zeisler, Thad Scheer, Marc Sewell, Laura Sewell, Hernan Astudillo, Theresa Smith, Roger Hebden, Chip Schwartz, Jack 和 Gillian Hassall, John Eaton, Amjad Farooq, John Holmes, John Weiler, Kevin Tyson, Kendall White, Chibuike Nwaeze, Dave Dikel, David Kane, John Williams, Bhavani Thuraisingham, Jim Baldo, Eric Stein, John Hetrick, Dave Gregory, John Bentley, Nigel Pates, Richard Taylor-Carr, Dan Lam, Garrett Fuller, David Broudy, Mike Baba, Burt Ellis, Matthew Presley, Robert Davis, Peter Lee, Linda Kemby, Georgene Murray, Alfredo Aunon, Jim Gray 和 Woody Lewis。

目 录

第 1 章 绪论	(1)
1.1 给软件架构师的建议	(2)
劝告的话	(3)
新生知识	(3)
无所适从	(4)
同行相欺	(5)
管理陷阱	(5)
定义软件体系结构	(6)
“体系结构”一词的误用	(6)
体系结构之前	(7)
软件危机	(8)
1.2 作为一门学科的软件体系结构	(9)
体系结构方法	(10)
通用原则	(11)
体系结构的论争	(12)
革新的软件体系结构	(13)
体系结构范型的转变	(14)
体系结构标准	(19)
应用程序和配置文件	(26)
视点的记法	(27)
1.3 设计模式与软件体系结构	(28)
设计模式	(29)
软件设计层模型	(30)
反模式	(39)
1.4 结论	(40)
1.5 练习	(41)

第 2 章 软件体系结构：基础训练	(45)
2.1 软件范型	(46)
面向对象范型	(46)
技术与系统规模	(47)
对象成为商业的基线	(48)
数据库与对象	(50)
主流中的对象	(50)
靠近构件：脚本语言	(50)
构件化软件：面向构件的范型	(51)
“构件”与“对象”	(51)
构件基础设施	(52)
构件的软件模式	(54)
构件化的软件体系结构	(55)
基于构件的软件开发	(56)
2.2 开放系统技术	(57)
2.3 客户机/服务器技术	(61)
2.4 软件应用体验	(72)
2.5 技术和应用体系	(74)
2.6 将标准用于应用系统	(77)
2.7 分布式基础设施	(81)
2.8 结论	(90)
2.9 练习	(91)
第 3 章 软件体系结构：准备战斗	(95)
3.1 软件体系结构范型转变	(95)
传统的系统假设	(96)
分布式假设——对传统的颠覆	(96)
多组织系统	(96)
作出范型转变	(97)
3.2 错误的方法	(98)
旧的软件	(98)
一个例子：错误的方法	(99)

骑士上场：英雄式程序员	(100)
3.3 正确的方法：企业体系结构开发	(101)
以体系结构为中心的过程	(102)
步骤 1：系统构想	(102)
步骤 2：需求分析	(104)
步骤 3：体系结构规划	(105)
计算接口体系结构	(107)
分布式工程体系结构	(107)
技术选择体系结构	(109)
步骤 4：实现模型	(109)
步骤 5：体系结构原型	(110)
步骤 6：项目管理规划	(110)
步骤 7：并行增量开发	(111)
步骤 8：系统转换	(112)
步骤 9：操作和维护	(112)
步骤 10：系统移植	(113)
3.4 底线：时间、人员和费用	(113)
3.5 结论	(114)
3.6 练习	(115)
第 4 章 软件体系结构：训练学校	(117)
4.1 体系结构与编程	(117)
软件的分形模型	(118)
主要的设计要点	(118)
设计要点的规模效应	(118)
软件设计层次	(119)
使用设计层次	(119)
4.2 使用体系结构管理复杂性	(120)
创造复杂性	(120)
选项 1：压缩	(121)
选项 2：隐藏	(122)
选项 3：忽略	(122)

选项 4: 切分	(122)
选项 5: 分块	(123)
4.3 系统集成	(123)
4.4 业务实例	(129)
4.5 从体系结构到软件开发	(133)
4.6 体系结构的软件记法	(139)
4.7 总结	(150)
4.8 练习	(151)
第 5 章 领导素质训练	(153)
5.1 领导素质是一项必要的也是可以学会的能力	(153)
5.2 架构师——团队的建立者	(154)
5.3 坚持优秀产品的理念	(156)
5.4 架构师的走查	(161)
5.5 总结	(164)
5.6 练习	(165)
第 6 章 软件体系结构: 伞兵训练学校	(167)
6.1 过程	(167)
6.2 建立新的过程	(174)
6.3 团队合作	(175)
6.4 结论	(181)
6.5 练习	(182)
第 7 章 沟通的训练	(189)
7.1 沟通的挑战	(189)
7.2 责任驱动的开发	(190)
7.3 沟通的责任	(192)
7.4 处理反馈	(192)
7.5 练习	(193)
第 8 章 软件体系结构: 情报操作	(195)
8.1 体系结构挖掘	(196)
自底向上与自顶向下	(196)
体系结构耕作	(196)

体系结构挖掘过程	(197)
挖掘的适用性	(198)
成功挖掘	(199)
水平与垂直	(199)
水平设计元素	(202)
关于可追踪性	(204)
为未来应用而设计	(204)
8.2 体系结构迭代	(205)
软件过程背景	(206)
体系结构过程的角色	(208)
宏观过程：体系结构迭代	(211)
开发人员对体系结构的反应	(213)
寻求情报，迭代设计	(215)
微观过程：子项目体系结构	(216)
混沌构建	(218)
8.3 体系结构评定	(221)
问题求解	(222)
评审与检查	(223)
8.4 结论	(225)
8.5 练习	(225)
第 9 章 软件体系结构：心理战	(229)
9.1 选择性学习	(229)
9.2 内部控制	(230)
9.3 期望管理	(231)
9.4 真理的心理学	(232)
9.5 感知未必真实	(233)
9.6 探讨人性的弱点	(234)
感知参考模型	(236)
生物反应模型	(237)
反应的群体应用	(238)
9.7 例子：参考销售	(239)

9.8	所有权心理学	(241)
9.9	心理合气道	(242)
9.10	智力合气道	(245)
	赢得战争	(247)
	赢得和平	(248)
9.11	结论	(249)
9.12	练习	(250)
附录 A	体系结构实例：测试结果报告系统	(255)
A.1	导言	(255)
A.2	构件互操作的挑战	(255)
A.3	TRRS 的目标体系结构	(256)
A.4	目标企业视点	(256)
A.5	目标信息视点	(257)
A.6	目标技术视点	(259)
A.7	原型实现	(260)
A.8	原型计算视点	(261)
A.9	TRRS 术语	(262)
A.10	用例定义	(263)
	UC001 产品信息检索会话	(263)
	UC002 产品信息登记会话	(264)
	UC003 经验报告登记会话	(265)
A.11	核心 workflow	(266)
	UC004 产品登记 workflow	(266)
	UC005 COTS 确认 workflow	(266)
	UC006 互操作性确认 workflow	(267)
	UC007 经验报告更新	(267)
A.12	信息模型	(268)
	产品的信息	(269)
	标准配置文件的信息	(270)
	组织的信息	(271)
	产品相关服务的信息	(272)

经验报告的信息	(272)
操作环境的信息	(273)
A.13 结论	(273)
附录 B 设计模板及实例	(275)
B.1 概念性设计	(275)
第一部分 目标	(275)
第二部分 概念性综述	(276)
第三部分 责任	(276)
第四部分 体系结构层	(277)
第五部分 类和对象、类的语义以及类之间的关系	(277)
第六部分 特征、数据类型和约束的描述	(278)
第七部分 如何使设计能处理相关的用例及需求	(279)
B.2 关系服务的概念性设计	(279)
第一部分 目标	(279)
第二部分 概念性综述	(279)
第三部分 责任	(281)
第四部分 体系结构的层次	(281)
第五部分 类和对象、类的语义以及类之间的联系	(281)
第六部分 特征、数据类型以及约束的描述	(284)
第七部分 如何使设计能处理相关的用例及需求	(285)
B.3 高层设计	(286)
第一部分 屏幕模型	(286)
第二部分 UML 图	(286)
第三部分 发现界面的使用	(286)
第四部分 构件复用	(287)
第五部分 典型用例	(287)
B.4 关系服务的高层设计	(287)
第一部分 屏幕模型	(287)
第二部分 设计的考虑	(287)
第三部分 构件复用	(288)
第四部分 发现界面的使用	(288)

第五部分	OMG IDL	(288)
第六部分	总体	(291)
第七部分	用于事件跟踪的典型用例	(292)
第八部分	客户端配置文件	(292)
第九部分	UML 类图	(292)
附录 C	软件体系结构术语表	(295)
参考文献		(305)