

十年磨剑——CSDN原软件工程大版主心血凝聚之作
新版再现——理论与实践的密切结合

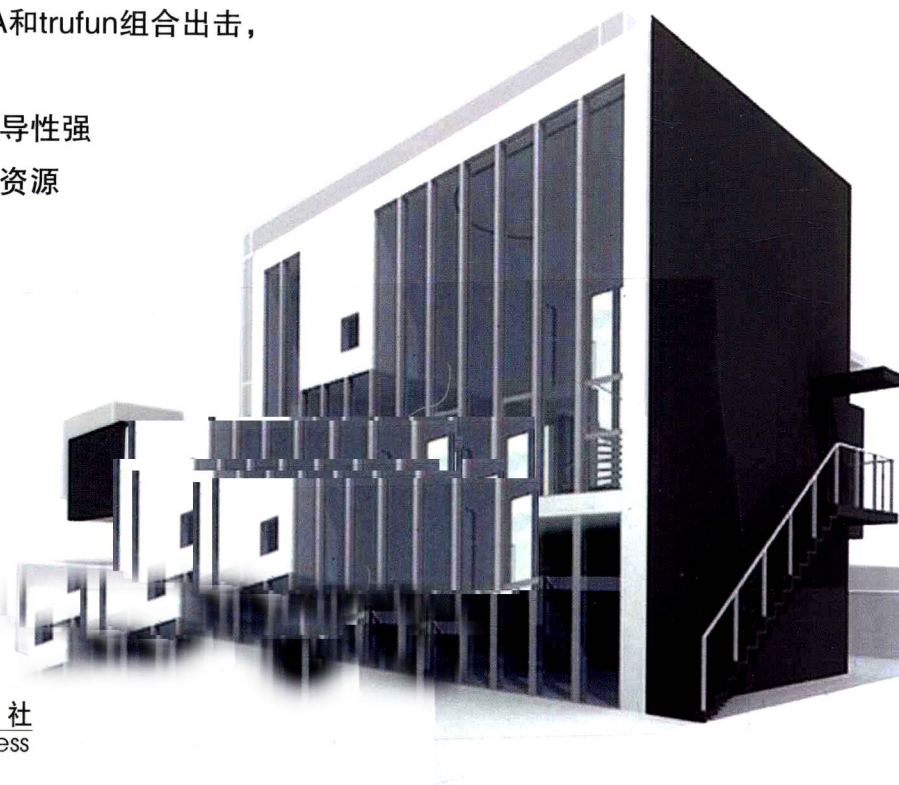
软件工程 之全程建模实现

青 润 著

- 延续实用风格，扩展最新研究成果
- 直击建模过程中的用户问题，有的放矢
- 建模工具rose、EA和trufun组合出击，
突出重点
- 配有讲座视频，指导性强
- 随书附赠丰富学习资源



机械工业出版社
China Machine Press



软件工程 之全程建模实现

青 润 著



机械工业出版社
China Machine Press

本书采用 UML 建模实现了软件工程的主要过程：需求、分析、设计、代码导出、设计模型维护等。本书采用了大量国内实际工程软件过程中的截图，通过图形和示例来描述工程实际中的问题和过程。这在国内原创的软件工程类书籍中是很少见的。本书对软件工程领域进行了重新划分，其中涉及软件工程方法、软件工程过程，以及软件工程理论的发展和现状，还包括全新的软件组织方法论：交换编程、可度量绩效管理模型、实用知识库体系构建，同时对软件开发中心态对抗问题进行了初步研究。

本书适合于对 UML 基础知识有一定了解的、同时参加过一些实际工程项目开发的人员阅读，也适合对全程建模过程实现和人性化软件开发感兴趣的读者参考。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

软件工程之全程建模实现/青润著. —北京：机械工业出版社，2010.8

ISBN 978-7-111-30790-7

I. 软… II. 青… III. 软件工程 - 系统建模 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 128901 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑：秦健

北京市荣盛彩色印刷有限公司印刷

2010 年 10 月第 1 版第 1 次印刷

186mm × 240mm · 22.25 印张

标准书号：ISBN 978-7-111-30790-7

ISBN 978-7-89451-533-9 (光盘)

定价：49.00 元 (附光盘)

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com



Preface 前言

写作经过

本书的撰写过程整整耗费了8年多的时间。回想起来，初次产生写本书的想法是在2002年的4月，当时笔者正在上海做一个企业ERP项目。这个项目完成后，笔者就觉得实验没有完全成功，因为笔者只是亲自验证了需求部分的实践，而没有时间来完整地验证分析、设计和编码的后续过程，与其他人配合开发的过程，以及将一个模型任务分配给其他程序员的实现过程。直到2002年的7月，笔者有机会到南京参加了一个大型信息系统实用性原型的开发，从而顺利地完成了分析、设计和编码过程的实践。同时，还完成了一本针对高级用户进行培训的开发过程教材。

经过笔者一段时间对教程资料的整理，以及在2001年中完成的几个项目的经历，笔者最终完成了本书现有的第3章（分析设计）和第4章（代码模型一致性）内容的第一次成稿。当时命名为《基于Rose的全程建模实例》，并将前几部分发表于《程序员》杂志2002年第10期~第12期。后来因为全文较长，CSDN^①就将全文直接合发于《程序员》杂志2002年合订本。

该文发表后，笔者来到北京进行一个大型项目的需求调研工作。期间，笔者与Wonder、Ozzzzzz和Lynx几位同仁从上午10点开始一直到晚上7点多进行了一次长谈。除了讨论软件工程过程的相关内容外，笔者还和Lynx专门针对全程建模过程中从设计阶段到编码阶段可能遇到的问题进行了相当长时间的讨论，我们的观点略有不同。笔者认为：设计可以有效地转换成代码，实现软件工程的模式，实现设计与编码的分离；Lynx兄则认为：在目前条件下，设计与编码不可能分开。不过，他认

① 中国软件开发网，英文全称为China Software Developer Net（CSDN）。国内一个著名的开发者论坛网站。

可了在设计人员水平足够高的情况下,是可以实现设计与编码分离的。但是,实现这样的分离对设计人员的要求可能会过高,而在国内这种高水平的设计人员目前还十分稀缺。

《基于 Rose 的全程建模实例》全文发表之后,笔者开始撰写《基于 Rose 的全程建模之二——Rose 中的协作开发》一文。当时,只是打算把 Rose 中提供的协作开发方式和配置管理工具的配合使用方式全部写出来,但是,随着公司的软件在全国范围内的铺开,笔者发现自己无法将三种主要的配置管理工具的环境都搭建起来,不得不放弃了对配置管理工具部分的撰写,最终就只剩下本书第 5 章(协作开发)部分的内容了。

在撰写《基于 Rose 的全程建模之二——Rose 中的协作开发》的同时,笔者开始规划《基于 Rose 的全程建模之三——如何进行需求分析》部分的撰写。笔者于 2002 年 10 月开始调研中国电信集团公司的业务需求,并将调研过程中总结的方法和内容写入这篇文章中。这部分内容撰写的时间最长。在笔者第一次到拉萨出差时,借着期间等待合同小签的空闲时间完成了这部分的主要内容,回北京后完成了全部内容。最后形成了本书的第 2 章(需求工程)部分的内容。

由于笔者第一篇文章的内容获得不错的反响,而第二篇文章的内容较多,CSDN 的孟迎霞认为发表在杂志上不是很合适,于是建议笔者将这些内容全部写出来然后出一本书。因此,笔者在 2003 年 3 月决定将所有的文章合并起来作为一本书来考虑。

2003 年 7 月下旬,在完成了需求阶段的内容后,笔者开始整理全书的基础目录部分,于 7 月底成功地完成了全书的第一次整合。

凭着笔者在电信集团公司不错的人缘关系,在项目合同基本全部签订之后,笔者开始在集团公司负责实施、处理 5 个开发商之间的系统接口和与集团系统的上报下发关系,同时开始规划系统第二个版本(BI 版本)的功能。笔者在电信集团公司停留的前两个月,在空闲时间不断地对全书的内容进行修订,经过对书稿的 9 次打印稿的纸面修改,最终在 9 月初完成了对全书的一次整体修订,并向出版社交付了全书初稿。

本书在 2004 年 5 月的第 1 版印刷后,笔者从电信四川设计院辞职,此后分别在一家外企和中科院自动化所工作了几年,期间独立对外提供技术咨询和培训服务。

其中 2006 年元旦的培训中，很多朋友提供了他们的建议和考虑，尤其是来自山东的朋友满桂平自从 2004 年 7 月看过我的书后在后续的几年中不断给笔者提供了他的实践信息和补充材料。

第 2 版中融合了 2004 年~2010 年初笔者所总结的除了已经采集了数据还没有进行文章撰写以外的几乎全部方法。其中包括：软件开发心理学的 5 章的内容，其中 2 章是首次对外发布；软件组织方法论的 4 章的内容，其中实用知识库体系构建一章也是首次发布；在技巧和相关文档一章中，删除了已经过时的对 Rational 在 IBM 中的顾虑、关于 Rose 与 Together 的 2 章内容，补充了项目计划如何做、评审如何做、配置管理工具如何使用以及 Use Case 阐述模型化问题及延伸的内容，同时给读者提供了一个最简化开发过程模型的建议——因为在培训和咨询中发现很多企业和院校都有人提到过，不知道什么样的过程是最简化的，哪些是软件开发中可以省略的，哪些是必不可少的。

同时，在这个版本的撰写过程中，还单独增加了软件工程概论一章，以期从整个软件工程的整体层面进行细分，阐述在新技术和环境下软件工程自身形态与过去的区别和变化过程，给读者一个关于软件工程的整体印象，同时对现有的传统软件 engineering 体系进行了一系列的反思并建议了一个初步的针对整个软件工程体系的划分方式。

针对第 1 版中已经存在的需求工程、分析设计、代码模型一致性和协作开发 4 个核心章都进行了修订，尤其是对需求工程进行了较大篇幅的修订，将需求代码间的映射关系进行了着重的补充，这也是这几年针对整个方法论中的最重要的补充之一。

本书第 2 版的积累长达 6 年多的时间，实际整合和撰写时间长达一年半（可以参看本书修订历史记录信息），但是其中肯定还存在很多不足之处，也希望这些年从事实践全程建模方法论的朋友们能够不吝赐教，以便于后续的补充和修订，笔者希望下次本书修订再版的时候，您也会出现在感谢名单中或者出现在作者一栏中。

本书特色

本书中所有的示例都来自 2001 年初~2009 年间笔者经历的、采用 UML 进行建模支持的 3 个大中型工程软件项目和近 10 个中小型工程软件项目。通过十多年来实际工程项目的经验总结，本书采用 UML 建模方式进行软件的开发，覆盖了软件工程从需求开始，经过分析（概要设计）、设计（详细设计）、编码的导出，以及

设计模型的维护、协作开发等所有主要阶段的内容，本书还包括最近4~5年内参加培训和咨询的学员们提供的各种改进建议和方法。

本书中的所有例子都源于国内实际的软件工程项目，因此具有很高的现实意义和参考价值。书中的示例和操作技巧，大部分都是第一次公开，其中小部分内容曾经在CSDN的软件工程版块出现过，但是在CSDN上公开的部分都无法张贴示例图，所以，无法形象地描述清楚，本书的撰写也为此提供了一次完整解释的机会。

本书中关于软件开发方法论以外的大部分内容都曾以对话或者单独成文的方式分别发表于《程序员》杂志、IT168/ITPUB技术社区，最后都曾整理发布于青润的两个主要blog空间，分别是：

青润心情 <http://blog.csdn.net/qingrun>

青润风度 <http://space.itpub.net/257598/>

青润的其他blog空间都不是主要空间，内容更新会有所不同，而上述两个空间将同步更新所有最新内容，2008年4月以前的内容基本上都在青润心情发布。

由于篇幅所限，本书中没有对UML的基础知识和相关符号进行更多的解释和说明。关于这部分内容，希望读者能主动学习一些UML的基本知识后再来学习本书中的内容。

本书的组织结构

第1章（软件工程概论）从软件工程的总体上讨论软件工程知识的细分问题，其中涉及软件工程方法、软件工程过程，以及软件工程理论的发展和现状。

第2章（需求工程）详细地阐述实现目标所需要进行的前期需求工程的准备，以及需求调研过程中经常遇到和需要注意的一些问题。需求工程的内容可以协助开发人员顺利完成需求调研工作，让用户和开发人员都能得到一个满意的结果。

第3章（分析设计）和第4章（代码模型一致性），详细阐述笔者所持的关于设计过程与编码实施过程可以实现完全分离开发的观点。

第5章（协作开发）介绍了一些关于团队开发过程中涉及的内容。如协作开发步骤、多人协作开发的方式和文档模型结合等第1版中已经发布过的内容。

第6章（软件组织方法论）是本书的一个尝试性的全新扩展，包括了自2005年起发布过的交换编程和可度量绩效管理模型等内容，还有从未发布过的实用性知

识库体系构建方面的知识，这些都是软件团队组织形态方面的内容，为整个软件工程理论提供了支撑。

第7章（软件开发中的心态对抗问题研究），如果提升到一定层面那就是软件开发心理学问题。这一章可以看做是软件开发心理学部分的雏形篇章。由于作者本人的经历和精力有限，在这个方向上的研究还比较薄弱，目前只能完成这仅有的5个方面的分析，还有至少两类角色的心态研究内容正处于撰写中，已经完成了市场人员心态和游戏玩家心态的数据积累，但是还不能完成这方面全面的文字阐述。

第8章（技巧和相关文档）是为本书中实现团队开发所需的一些必备资料或技巧。这些文档和资料将协助你和团队实现更快速更高效的建模开发过程。

致谢

本书撰写过程中，尤其是在第1版的撰写过程中，很多朋友对笔者提供了无私的帮助和支持。书中的很多内容也来源于和这些朋友的对话、讨论、争执，甚至部分内容就是笔者偷师学艺所得。本不敢藏私，但在第1版出版的过程中由于过度仓促，没有能够在书中对这些朋友表示感谢，在这里一并对下面的朋友表示感谢：

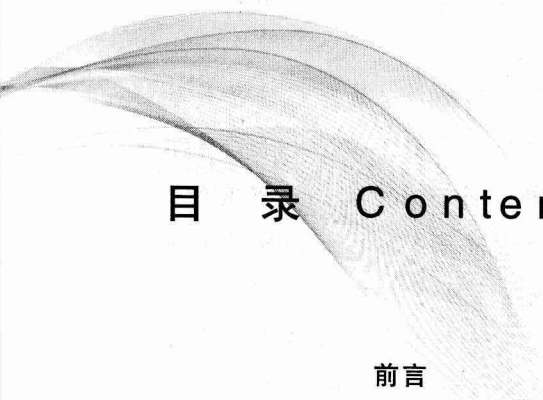
感谢在托普工作期间，与笔者一同为了项目而拼命工作的成都中央研究院和上海研究院的朋友们。

感谢在中国电信集团网络发展部工作的朋友和网发部的各位领导，以及曾经在2002年~2004年与笔者一同借调到中国电信集团网络发展部工作的各省电信公司以及几大电信设计院的朋友们。这里就不一一提名致谢了。

感谢2001年~2003年在CSDN软工/管理板块上与笔者讨论的朋友们，他们是：吴晓军（Wonder）、马仲男（Mach）、刘寅虢（Lynx）、杨威（Ltp3）、刘新生（Ozzzzzz）、周永丽（Rose）、周海峰等。还要感谢孟岩、蒋涛、孟迎霞等CSDN的老朋友们。如果不是孟岩兄2004年初推心置腹的一番话，估计青润这辈子都不会写书了。

另外，还要感谢2005年后和青润合作的曾在中科院自动化所一起工作和学习的朋友们。

最后，感谢我的妻子朱丹女士在青润没有什么收入的几年中给我的工作和理想的默默支持和奉献。



目 录 Contents

前言

第1章 软件工程概论/1

1.1 概述/2

1.1.1 全程建模下的软件工程分类/2

1.1.2 传统软件工程分类/3

1.1.3 SWBOOK2004 中软件工程体系划分/5

1.2 软件工程过程/7

1.2.1 软件工程管理过程/7

1.2.2 软件工程实施过程/8

1.3 软件工程方法/8

1.3.1 软件开发方法论/9

1.3.2 软件组织方法论/14

第2章 需求工程/16

2.1 需求的定义/17

2.2 需求调研/18

2.2.1 调研中存在的问题/18

2.2.2 过去和现在的对比/21

2.2.3 新现象的分析/22

2.2.4 软件外包与编码/23

2.3 如何和用户交流/23

- 2.3.1 交流四要点/23
- 2.3.2 什么是用户/27
- 2.3.3 参与人员建议/30
- 2.3.4 输入输出/30
- 2.3.5 UML 元素/30
- 2.3.6 常见问题/31
- 2.4 业务建模/31
 - 2.4.1 目的/31
 - 2.4.2 基本操作步骤/33
 - 2.4.3 建议/34
 - 2.4.4 参与人员建议/35
 - 2.4.5 输入输出/35
 - 2.4.6 UML 元素/36
 - 2.4.7 常见问题/36
 - 2.4.8 示例说明/38
- 2.5 Use Case 模型/39
 - 2.5.1 Use Case 图的绘制/39
 - 2.5.2 如何查找 Use Case 和 Actor/43
 - 2.5.3 Use Case 的处理/49
 - 2.5.4 如何分包/52
 - 2.5.5 参与人员建议/59
 - 2.5.6 输入输出/59
 - 2.5.7 UML 元素/60
 - 2.5.8 常见问题/60
 - 2.5.9 楚凡 UML 工具模型示例/67
- 2.6 Use Case 阐述/72
 - 2.6.1 目的/72
 - 2.6.2 要求/73
 - 2.6.3 Use Case 阐述的格式/73
 - 2.6.4 参与人员建议/75

- 2.6.5 输入输出/75
- 2.6.6 UML 元素/76
- 2.6.7 例子/76
- 2.6.8 常见问题/78
- 2.7 交互建模/82
 - 2.7.1 目的/82
 - 2.7.2 要求/82
 - 2.7.3 例子/83
 - 2.7.4 参与人员建议/86
 - 2.7.5 输入输出/86
 - 2.7.6 UML 元素/86
 - 2.7.7 常见问题/87
- 2.8 界面设计/88
 - 2.8.1 什么是界面设计/88
 - 2.8.2 界面设计的原则/89
 - 2.8.3 RUP 中的相关描述/89
 - 2.8.4 界面设计的基本要求/90
 - 2.8.5 界面设计的步骤/91
 - 2.8.6 界面设计的定位/92
 - 2.8.7 界面设计的时间特点/92
 - 2.8.8 界面设计与原型法/93
 - 2.8.9 需要注意的问题/93
 - 2.8.10 常见问题分析/95
 - 2.8.11 参与人员建议/95
 - 2.8.12 输入输出/96
 - 2.8.13 UML 元素/96

第3章 分析设计/97

- 3.1 概述/98
 - 3.1.1 原因/98

- 3.1.2 两个项目简介/98
- 3.1.3 初步结论/99
- 3.2 分析设计的定义/99
 - 3.2.1 什么是分析/99
 - 3.2.2 什么是设计/100
- 3.3 分析模型/101
 - 3.3.1 模式的选择与应用/101
 - 3.3.2 构建分析类/102
 - 3.3.3 分析模型时序图/110
 - 3.3.4 参与人员建议/113
 - 3.3.5 输入输出/113
 - 3.3.6 UML 元素/114
 - 3.3.7 常见问题/114
 - 3.3.8 楚凡 UML 工具模型示例/117
- 3.4 设计模型/119
 - 3.4.1 设计模式的选择与应用/120
 - 3.4.2 设计类的构建/122
 - 3.4.3 设计模型时序图/136
 - 3.4.4 一个示例/139
 - 3.4.5 参与人员建议/142
 - 3.4.6 输入输出/142
 - 3.4.7 UML 元素/142
 - 3.4.8 楚凡 UML 工具模型示例/142

第4章 代码模型一致性/147

- 4.1 概述/148
- 4.2 生成框架代码/148
 - 4.2.1 类的语法检查/148
 - 4.2.2 ClassPath 的设置/151

- 4.2.3 导出代码/154
- 4.3 维护设计模型/159
 - 4.3.1 目的/159
 - 4.3.2 维护方式/160
 - 4.3.3 维护步骤/160
- 4.4 反工/161
 - 4.4.1 作用/161
 - 4.4.2 操作步骤/162
- 4.5 参与人员建议/166
 - 4.5.1 输入输出/167
 - 4.5.2 UML 元素/167
- 4.6 附加说明/167
 - 4.6.1 需求变化时的策略/167
 - 4.6.2 需要注意的问题/167

第5章 协作开发/169

- 5.1 协作开发过程/170
 - 5.1.1 准备工作/170
 - 5.1.2 分包/172
 - 5.1.3 合并分包/174
 - 5.1.4 分包特殊操作/175
- 5.2 多人协作开发/179
 - 5.2.1 第一种方法/179
 - 5.2.2 第二种方法/179
 - 5.2.3 一个特殊情况/179
- 5.3 文档模型结合/180
 - 5.3.1 概述/180
 - 5.3.2 文档模型的结合/180

第6章 软件组织方法论/183

- 6.1 结对编程/184
 - 6.1.1 产生/184
 - 6.1.2 结对编程形成/184
 - 6.1.3 结对编程的变形模式/185
- 6.2 交换编程/188
 - 6.2.1 引言/188
 - 6.2.2 相关工作/189
 - 6.2.3 方法定义/191
 - 6.2.4 优劣分析/193
 - 6.2.5 试验结果/195
 - 6.2.6 小结/196
- 6.3 可度量绩效管理模型/197
 - 6.3.1 引言/197
 - 6.3.2 团队组成与管理划分/197
 - 6.3.3 绩效管理办法基础/199
 - 6.3.4 绩效管理实现模型/200
 - 6.3.5 薪资收入组成模型/204
 - 6.3.6 绩效管理算法模型/206
 - 6.3.7 小结/206
- 6.4 实用性知识库体系构建/206
 - 6.4.1 基本原则/207
 - 6.4.2 内容分类/207
 - 6.4.3 企业基本制度和新员工学习材料库/207
 - 6.4.4 技术分类文档和学习材料库/208
 - 6.4.5 企业代码库/209
 - 6.4.6 行业客户资料和业务资料库/210
 - 6.4.7 企业知识库管理创建规则/210
 - 6.4.8 小结/211

第7章 软件开发中的心态对抗问题研究/212

7.1 项目管理者心态问题研究/214

7.1.1 引言/214

7.1.2 角色定位/215

7.1.3 心态对抗解析/217

7.1.4 小结/220

7.2 高层管理者心态问题研究/220

7.2.1 引言/220

7.2.2 角色定位/221

7.2.3 心态对抗解析/223

7.2.4 小结/230

7.3 管理中的权力使用分析/230

7.3.1 管理中的放权与分权/230

7.3.2 管理中的权力限制/234

7.3.3 项目管理者如何与组员合作/236

7.3.4 一些常见问题/237

7.4 项目经理分类/239

7.4.1 项目管理之传统/240

7.4.2 强势项目经理/241

7.4.3 弱势项目经理/242

7.4.4 题外话/244

7.4.5 小结/244

7.5 论本本主义/244

7.5.1 2009年的几件事情/244

7.5.2 引发的思考/245

7.5.3 产品、技术、标准/247

7.5.4 软件开发本本主义/251

7.5.5 最后的分析/252

第8章 技巧和相关文档/253

8.1 cat 包的修改/254

8.1.1 示例环境/254

8.1.2 操作步骤/257

8.1.3 结果/261

8.2 ClearCase 安装指南/261

8.2.1 概述/261

8.2.2 安装前的准备/262

8.2.3 配置安装/264

8.2.4 关于 License 的申请安装/267

8.3 关于 CMM 的一些观点/267

8.3.1 SQA 和 SQC/267

8.3.2 CMM, 评估、认证/269

8.3.3 只对组织中的某一个部门进行评估/269

8.3.4 评估是反复不断的/269

8.3.5 关于 CMM 过级时间的个人观点/269

8.4 什么是系统分析/271

8.4.1 定义/271

8.4.2 现实状况/271

8.4.3 一些建议/271

8.5 项目计划如何做/272

8.5.1 什么是计划/272

8.5.2 评价你们的计划状态/273

8.5.3 项目中的计划/273

8.5.4 计划的生命周期/274

8.5.5 计划中的常见错误/279

8.5.6 小结/280

8.6 评审如何做/280

8.6.1 什么是评审/280

- 8.6.2 对评审状态的评价/280
- 8.6.3 评审过程/281
- 8.6.4 评审常见错误/282
- 8.6.5 小结/283
- 8.7 配置管理工具如何用/283
 - 8.7.1 配置管理工具的使用方法/284
 - 8.7.2 小结/287
- 8.8 软件人员的分化/287
 - 8.8.1 人员分化原因/287
 - 8.8.2 需求人员的分离/288
 - 8.8.3 设计人员的分离/289
 - 8.8.4 编码人员的分离/290
 - 8.8.5 关于测试人员/291
 - 8.8.6 小结/295
- 8.9 Use Case 阐述模型化问题及延伸/295
- 8.10 最简化开发过程/297

附录 A 全程建模方法论/299

附录 B 幼儿教育系统/307

附录 C 全程建模培训介绍/312

附录 D 作者随笔一/316

附录 E 作者随笔二——毕业十年记/322

附录 F 文档信息和文档修订历史/327

后记/336

参考文献/337