

 WILEY

# 使用UML

## 进行面向对象的项目管理

Object-Oriented  
Project Management  
with UML

〔美〕Murray Cantor 著

徐 晖 许志敏 李 虎 谢 超 译



人民邮电出版社  
POSTS & TELECOM PRESS

# 使用 UML 进行面向对象的项目管理

[美]Murray Cantor 著

徐晖 许志敏 李虎 谢超 译



人民邮电出版社

## 图书在版编目 (CIP) 数据

使用 UML 进行面向对象的项目管理/ (美) 坎托 (Cantor, M.) 著; 徐晖等译。  
—北京: 人民邮电出版社, 2004.3  
ISBN 7-115-12034-X

I. 使... II. ①坎... ②徐... III. ①面向对象语言 UNL—程序设计  
②软件开发—项目管理 IV. ①TP312 ②TP311.52

中国版本图书馆 CIP 数据核字 (2004) 第 002473 号

## 版 权 声 明

Murray Cantor: Object-Oriented Project Management with UML

Copyright © 1998 by Murray Cantor.

All Rights Reserved.

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书中文简体字版由 John Wiley & Sons 公司授权人民邮电出版社出版, 专有出版权属于人民邮电出版社。

## 使用 UML 进行面向对象的项目管理

- ◆ 著 [美] Murray Cantor  
译 徐 晖 许志敏 李 虎 谢 超  
责任编辑 俞 彬
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
读者热线 010-67132705  
北京汉魂图文设计有限公司制作  
北京隆昌伟业印刷有限公司印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16  
印张: 18.75  
字数: 448 千字 2004 年 3 月第 1 版  
印数: 1-4 000 册 2004 年 3 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2002 - 4863 号

ISBN 7-115-12034-X/TP · 3814

定价: 36.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

本书介绍了如何使用面向对象方法学进行项目管理。

本书分三个部分。第一部分包括前 4 章，介绍了软件项目管理的重要原则、统一建模语言以及它在软件开发过程中的使用、软件开发生命周期模型等基础内容。本书的第二部分包括第 5 章至第 8 章，讨论了软件开发中与各阶段联系的各种概念的应用，详细讨论了 UML 软件开发过程的所有阶段和活动。第三部分介绍了如何评估及报告项目状态，并且提供各种平均值、预算和开发的评估方法检验项目是否沿着轨道前进。书中还有一个软件开发工程项目的例子，贯穿整本书中。

本书可作为管理面向对象开发的参考书，适合于开发人员、项目经理、软件工程研究人员阅读参考。

## 关于作者

---

Murray Cantor 博士在软件开发方面有十几年的管理经验，他目前在 TASC 任资深项目经理。TASC 是 Litton 公司的一家子公司，主要提供信息技术服务。Cantor 博士在 TASC 管理基于对象项目的开发，还是面向对象开发的内部顾问。在加入 TASC 之前，他曾经在 IBM 任产品开发经理，主要管理图形设计和开发及多媒体子系统。

## 致 谢

---

如果没有那么多人的帮助，本书是不可能完成的。有些人给予我直接的帮助，而其他人则是多年来一直与我共享他们的观点与学识。本书的完成让我可以有机会对他们表达我的谢意。

首先，我要感谢我的妻子 Judi。在整个写作过程中，她为我提供了有价值的建议、给予我支持并让我分享她的观点。另外她所做的远远超过了她该做的。下次该轮到她出书了。

我还要感谢 TASC 的管理人员，他们在我的写作过程中给予了我大力的支持。Frank Serna 是仿真器系统业务部门负责人，同时也是我的经理，我要特别感谢他在本书的整个写作过程中对我的鼓励和多次有价值的讨论。除此以外，本书还是与各位同事、员工及学生多次交流的结果。我特别要感谢 David Alvey、Roger Blais、Pat Durante、Robert McCarron、Hal Miller、William Myers、Michael Olson、David Pierce 和 Ralph Vincigeurra。Martin Svedlow 和 Todd Withey 也应该得到特别的感谢，他们在本书的技术应用方面给予了热心的支持。同样，我也要感谢 TASC 的图书管理员 Martha Dionne 及她的职员 Audrey Palmieri 和 Margaret Ouzts，感谢他们给予的宝贵的研究协助。还要感谢 Jan Lapper 和 Pauline Parziale，他们使我在写作中保持条理。

本书的大部分是基于我在 IBM 工作中得到的宝贵经验。我特别感谢 Khoa Nguyen（现在任职于 Videoserver 公司），他在我早期开始进行项目管理的时候指导过我。同时，在与 Robert Swann 和 Tred Scheibl 的共事过程中，我也学到了很多。

我尤其感激 John Wiley 的编辑们。感谢 Theresa Hudson 对本书的信任和支持，也感谢 Sandra Bontemps 和 Kathryn A.Malm 在排版过程中的耐心和努力。

作 者

本书可以作为一本现代软件开发的实践指南。它产生于我在 TASC 偶然开始教授的一门课程。那时正好有人要我评价一套关于管理面向对象开发的在线课程。我不喜欢那套课程，因为它是对陈旧的、不可信的瀑布式软件开发方法老调重弹式的讨论，只是替换了一些设计和编码步骤以及一些对象术语。当我表达这样的看法时，TASC 的管理部门对我发出挑战：“既然你这么聪明，那么就由你来教这门课程。”

在准备这门课程的时候，我认识到这样一些事情：首先是在业界中的我们确了解如何管理软件开发，因此没有任何理由使软件开发成为一个充满危机和具有高度压力的工作。然而在另一方面，绝大多数软件项目看上去并没有从我们最近几年积累的经验中受益。其次，在教授软件开发课程的时候，必须涉及多方面的知识，而这些知识中的多数通常是不会在大学的课程中讲授的。同时，由于众多开发人员仍是软件经理的最好来源，因此提供一本教授同业者的书籍是很重要的。正如本书所写，我了解到还没有其他书籍覆盖了那些我感觉对每个项目经理都很重要的资料。

第三个认识是，对象和对象设计语言（如统一建模语言，UML）的使用不仅能够改善开发的技术质量，而且可以促进项目经理的工作。我想所有的新的软件开发项目都应该采用对象方法学，并根据本书中涉及到的技术进行项目管理。

## 目标

本书以我的工作经验，可获得的文献，以及我与同事、员工及学生们的交流为基础，讨论了软件开发项目使用面向对象方法学进行项目管理的效率问题。我不仅说明了在整个项目的各个阶段中项目经理应该做的工作，而且解释了这些技术为什么是有效的。最后，本书还介绍了计划、人员安排、机构组织、开发以及面向对象软件产品的交付。由于要领导和管理团队，项目经理被深深地埋在每天的活动中。在书中我介绍了足够的理论背景，可以使你逐渐理解为什么以及如何做。这种理论将帮助你把这些技术应用到你的特定项目中。

作为一个项目经理，你应该拥有足够多的参考书籍，这是因为没有任何人能够知道一切你需要知道的事，作为扩展你的工作经验的基础。尽管本书

包括很多方面的内容，我仍把使组织和质量管理成熟这样的论题留在其他书中讨论，并仅从外围包括了其他内容。我还为那些想更多了解本书次要的主题的读者提供了参考资料列表。

## 本书的读者对象

本书主要的对象是软件工程及项目经理，不论是初学者还是专家，还有那些对如何使他们的角色适合整个开发过程感兴趣的开发人员，特别是那些渴望从事项目管理的人员。另外，那些项目经理的管理者们也应该阅读本书，因为用它可以渐进地对项目经理正在从事的工作给出一个正确的评价，并因此提供监督的基础。此外，本书对于那些负责签定软件合同的人同样有帮助。

我尽力让这本书对那些小项目的开发者有用，如同本书对那些正在大型项目中工作的开发团队的作用一样。我所描述的技术能够应用到对项目的研究和开发、内部工具开发、软件合同的签定，以及各种套装软件产品上。本书在很大程度上是独立的，我假定读者已经有一些软件开发的经验，并且对软件对象有一些概念，同时也假定读者至少参加过一个软件项目的开发，并且能够阅读甘特图（Gantt chart）。

## 本书的组织结构

本书分三部分。第一部分是本书其他部分的基础。为了成功地进行软件项目的管理，你需要了解一些重要的原则。这些基本原则在第1章中介绍。接下来第2章介绍了统一建模语言以及它在软件开发过程中的使用。第2章还为项目经理提供面向对象设计和UML的概述，熟悉UML的人可以跳过这一章。这一章还有一些关于管理复杂度的建议，你可能会发现UML专家也在使用这些建议。第3章通过对软件开发生命周期模型的讨论结束这个基础部分。第4章详述如何计划和组织你的项目。

本书的第二部分讨论了软件开发过程中与各阶段联系的各种概念的应用，详细讨论了UML软件开发过程的所有阶段和活动。第5章到第8章详尽描述软件开发生命周期，以及如何管理整个过程的各个阶段的各种活动、出口准则和特殊的管理和沟通问题。

第三部分仅包括第9章。它告诉你如何评估及报告你的项目状态，并且提供各种平均值、预算和开发的评估方法以检验你的项目是否沿着预定轨道前进。第9章还概述了一种报告格式，项目经理可以用它向上级管理层和用户进行简明的状态报告及项目健康程度报告。

本书除主要内容外，还提供了一个软件开发项目（隐形战斗机由人操纵的驾驶员座舱仿真器）作为一个例子贯穿整本书中。这个例子尽管很详细，但也只是功能性的，而且不是基于任何真实的项目。充分复杂的描写是为了说明本书中的方法是如何工作的。举例的目的是帮助你理解如何安排各种计划、如何组织各个团队、如何管理风险，以及如何开发代码和交付代码，而不是告诉你如何实际构建一个仿真器。我和一些开发人员一起审查了仿真器的架构，他们告诉我这个架构是合理的，但是我不能确定我将在真实的项目中实际地使用这个架构。

## 在 Web 站点上

由于本书的目标之一是为你日复一日的项目管理提供帮助，所以出版商 John Wiley & Sons 公司已经建立了一个 Web 站点 [www.wiley.com/compbooks/cantor](http://www.wiley.com/compbooks/cantor)，在那儿你能够下载一些资料，当你应用本书中的某些技术时，这些资料也许有用。站点中的主要内容有：

- 一个用于仿真器示例的简单项目文件，包括在第 4 章描述的用于软件项目开发的工作任务拆分结构（WBS）；
- 一个在第 5 章描述的用于管理用例的数据库模板；
- 在第 7 章讨论的在构造阶段用于跟踪开发过程的 Svedlow 图；
- 第 9 章讨论的开发输出产品状态模板。

除此以外，这个站点还有指向其他有用 Web 站点的链接。你也可以通过站点就本书中的各种思想给我写评论，并分享你的经验。与所有的 Web 站点一样，这个站点的内容也是随时更新的。

## 小结

本书的价值在于思想和技术的应用，因此要阅读它并使用 Web 页面。试着实现这些想法。它们为我工作，我想他们也将为你工作。感谢你阅读本书。我希望你对它感到满意。

## 第一部分 面向对象项目管理的基本原理

|                           |    |
|---------------------------|----|
| 第 1 章 面向对象开发的管理工具 .....   | 3  |
| 1.1 遇到的困难 .....           | 4  |
| 1.1.1 需求不充分且经常变更 .....    | 5  |
| 1.1.2 缺乏与客户的沟通 .....      | 8  |
| 1.1.3 团队内部沟通不畅 .....      | 9  |
| 1.1.4 不必要的复杂度 .....       | 10 |
| 1.1.5 团队工作缺乏效率 .....      | 11 |
| 1.2 用面向对象技术解决困难 .....     | 12 |
| 1.3 协作 .....              | 16 |
| 1.4 团队沟通 .....            | 21 |
| 1.4.1 公共词汇表和公共语言 .....    | 22 |
| 1.4.2 适当规模的沟通 .....       | 22 |
| 1.5 团队动态 .....            | 30 |
| 1.6 小结 .....              | 34 |
| 第 2 章 将统一建模语言作为管理工具 ..... | 37 |
| 2.1 使用抽象 .....            | 38 |
| 2.2 统一建模语言 .....          | 39 |
| 2.3 归档需求 .....            | 41 |
| 2.3.1 用例 .....            | 41 |
| 2.3.2 用例图 .....           | 44 |
| 2.3.3 编写包需求文档 .....       | 46 |

|              |                         |           |
|--------------|-------------------------|-----------|
| 2.4          | 编写软件设计文档 .....          | 49        |
| 2.5          | 一个良好设计的特征 .....         | 62        |
| 2.6          | 构件和子系统 .....            | 63        |
| 2.7          | 可追溯性 .....              | 66        |
| 2.8          | 小结 .....                | 68        |
| <b>第 3 章</b> | <b>选择开发生命周期模型 .....</b> | <b>69</b> |
| 3.1          | 生命周期模型原理 .....          | 70        |
| 3.2          | 四种生命周期模型 .....          | 75        |
| 3.2.1        | 瀑布模型 .....              | 76        |
| 3.2.2        | 螺旋模型 .....              | 79        |
| 3.2.3        | 快速应用程序开发：时间盒模型 .....    | 81        |
| 3.2.4        | 受控的迭代模型 .....           | 83        |
| 3.3          | 增量构建 .....              | 87        |
| 3.4          | 小结 .....                | 91        |
| <b>第 4 章</b> | <b>面向对象的项目计划 .....</b>  | <b>93</b> |
| 4.1          | 开发 SDP .....            | 95        |
| 4.1.1        | 研究计划 .....              | 96        |
| 4.1.2        | 实际建议 .....              | 111       |
| 4.2          | 设计 SDP .....            | 112       |
| 4.2.1        | 可交付的产品 .....            | 112       |
| 4.2.2        | 开发环境 .....              | 113       |
| 4.2.3        | 项目规模和人工量评估 .....        | 117       |
| 4.2.4        | 风险策划 .....              | 117       |
| 4.2.5        | 生命周期模型 .....            | 118       |
| 4.2.6        | 工作拆分结构（WBS） .....       | 118       |
| 4.2.7        | 进度表 .....               | 124       |
| 4.2.8        | 人员安排和组织 .....           | 128       |
| 4.2.9        | 产品小组 .....              | 133       |
| 4.2.10       | 时间段预算 .....             | 136       |
| 4.2.11       | 计划的评估方法 .....           | 138       |
| 4.3          | 小结 .....                | 138       |

## 第二部分 生命周期管理

|                          |     |
|--------------------------|-----|
| 第 5 章 管理初始阶段 .....       | 141 |
| 5.1 管理概述 .....           | 142 |
| 5.2 开发活动 .....           | 145 |
| 5.2.1 用例数据库 .....        | 146 |
| 5.2.2 用例图 .....          | 148 |
| 5.2.3 用户接口原型 .....       | 149 |
| 5.2.4 顶级类图 .....         | 151 |
| 5.2.5 系统测试计划 .....       | 152 |
| 5.3 进程任务 .....           | 157 |
| 5.3.1 职责分配 .....         | 158 |
| 5.3.2 阶段出口准则 .....       | 159 |
| 5.4 协调活动 .....           | 159 |
| 5.4.1 微观进度表 .....        | 159 |
| 5.4.2 项目协调会 .....        | 160 |
| 5.5 客户沟通 .....           | 161 |
| 5.5.1 可交付的产品的管理 .....    | 161 |
| 5.5.2 召开需求审查会议 .....     | 162 |
| 5.6 阶段完成 .....           | 164 |
| 5.7 小结 .....             | 165 |
| 第 6 章 管理细化阶段 .....       | 167 |
| 6.1 管理概述 .....           | 168 |
| 6.1.1 团队问题 .....         | 170 |
| 6.1.2 领导能力 .....         | 172 |
| 6.2 开发活动 .....           | 176 |
| 6.2.1 用例的细化 .....        | 178 |
| 6.2.2 类的设计 .....         | 180 |
| 6.2.3 系统设计说明 .....       | 186 |
| 6.2.4 子系统集成和单元测试计划 ..... | 186 |

|                         |            |
|-------------------------|------------|
| 6.2.5 更新输出产品 .....      | 190        |
| 6.3 进程任务 .....          | 191        |
| 6.4 协调活动 .....          | 192        |
| 6.4.1 内部设计审查 .....      | 192        |
| 6.4.2 项目会议 .....        | 195        |
| 6.5 客户和经理的沟通 .....      | 196        |
| 6.6 阶段的完成 .....         | 199        |
| 6.7 小结 .....            | 200        |
| <b>第7章 管理构造阶段 .....</b> | <b>201</b> |
| 7.1 管理概述 .....          | 202        |
| 7.1.1 重新安排基线 .....      | 203        |
| 7.1.2 团队问题 .....        | 206        |
| 7.1.3 领导能力 .....        | 207        |
| 7.2 开发任务 .....          | 209        |
| 7.2.1 集成策略 .....        | 210        |
| 7.2.2 类开发 .....         | 214        |
| 7.2.3 水平和垂直开发 .....     | 218        |
| 7.2.4 用户接口开发 .....      | 220        |
| 7.2.5 增量子系统集成和测试 .....  | 220        |
| 7.2.6 增量系统集成 .....      | 223        |
| 7.2.7 最终集成 .....        | 223        |
| 7.2.8 用户文档 .....        | 224        |
| 7.2.9 更新的输出产品 .....     | 226        |
| 7.3 构造进程任务 .....        | 226        |
| 7.4 协调并跟踪活动 .....       | 227        |
| 7.4.1 风险管理 .....        | 228        |
| 7.4.2 安排微观进度表 .....     | 229        |
| 7.4.3 变更控制 .....        | 232        |
| 7.4.4 过渡准备 .....        | 235        |
| 7.4.5 团队会议 .....        | 236        |
| 7.4.6 运行测试准备审查 .....    | 239        |
| 7.5 构建阶段出口准则 .....      | 239        |
| 7.6 小结 .....            | 240        |

第 8 章 管理过渡阶段 ..... 243

8.1 管理回顾 ..... 243

8.2 过渡阶段的开发活动 ..... 245

8.2.1 问题报告进程 ..... 245

8.2.2 处理缺陷 ..... 248

8.2.3 设计、编码和测试 ..... 252

8.3 过渡进程的任务 ..... 253

8.4 活动的协调 ..... 253

8.4.1 构建会议 ..... 254

8.4.2 项目会议 ..... 254

8.5 客户沟通 ..... 255

8.6 过渡阶段出口准则 ..... 256

8.7 小结 ..... 257

第三部分 测试进程并走向成功

第 9 章 跟踪与监督 ..... 261

9.1 预算的跟踪 ..... 262

9.2 开发的度量 ..... 268

9.3 月度状态报告 ..... 270

9.4 小结：成功的守则 ..... 275

附录 Web 页面 ..... 277

参考书目 ..... 279

# 第一部分

## 面向对象项目管理的基本原理



# 第 1 章 面向对象开发的管理工具

“知己知彼，百战不殆。”

孙子，中国古代军事家

公元前 490 年

那种带领一群分工不明的编程高手去完成一个大型系统的时代已经结束了。尽管那种组织项目开发的方式快速，但与此相应带来的恶果常常是混乱和持续不断的危机，并使开发人员的热情迅速消耗殆尽。为使软件的组织管理具有生命力，让这些开发人员在相对长久的时期内保持工作效率是绝对必要的。因此，如何协调好一个大的程序员团队开发出大型的复杂系统，同时又能避免不断出现的危机，是当今项目管理面临的挑战。

随着计算机系统功能的日益强大，有两种趋势值得我们关注：

- 对那些需要程序员团队开发的日益庞大、复杂的系统需求的增加；
- 用于编写较高级程序语言的自动化工具以良好软件设计包形式开发出来。

这两种趋势要求应用某些设计方法，可以使程序员团队能够快速、有效地开发出健壮且易于维护的程序代码。但最重要的趋势也许是，人们关注的焦点正在从机器向人员转变。早期的程序语言的设计受限于机器，程序员们不得不关心机器是如何处理数据的。我们尊敬那些精于此道的人，并为他们冠以“hacker”的头衔。今天被我们称作“意大利面条”（spaghetti）式的代码通常是这种编程方式的结果，这种编程方式的致命缺陷是：没有人能接手后续的工作，无法修改也无法完善它，甚至连原来编写那些代码的程序员经过多日后，也会想不起来它们是如何被写出来的。拆分程序很困难，修改一个程序漏洞很可能引起另一个错误。对代码进行功能扩展的尝试也是徒劳