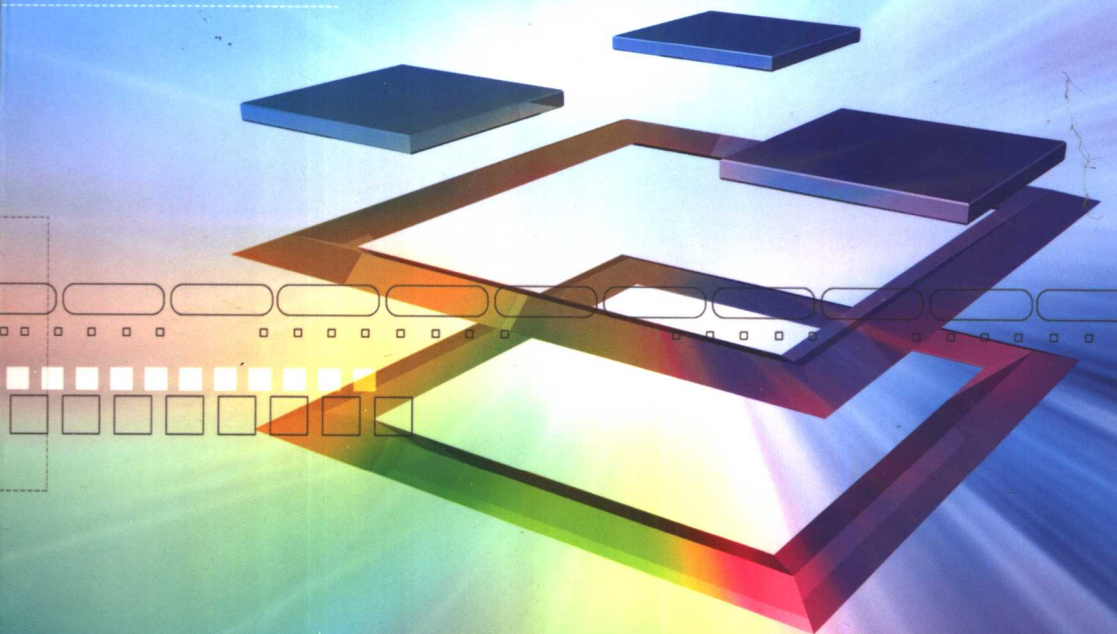
PEARSON
Addison
Wesley

统一过程精解

Kendall Scott 著
付宇光 朱剑平 译



Pearson Education

清华大学出版社

统一过程精解

Kendall Scott 著

付宇光 朱剑平 译

清华大学出版社

北京

图书在版编目(CIP)数据

统一过程精解/斯科特(Scott,K.)著;付宇光,朱剑平译. —北京:清华大学出版社,2005.4
书名原文:The Unified Process Explained
ISBN 7-302-10561-8

Ⅰ. 统… Ⅱ. ①斯… ②付… ③朱… Ⅲ. 软件开发 Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字(2005)第 013792 号

出 版 者: 清华大学出版社 **地 址**: 北京清华大学学研大厦
http://www.tup.com.cn **邮 编**: 100084
社 总 机: 010-62770175 **客户服务**: 010-62776969
责任编辑: 常晓波
封面设计: 立日新
印 刷 者: 北京四季青印刷厂
装 订 者: 三河市李旗庄少明装订厂
发 行 者: 新华书店总店北京发行所
开 本: 185×230 **印张**: 9.25 **字数**: 191 千字
版 次: 2005 年 4 月第 1 版 2005 年 4 月第1 次印刷
书 号: ISBN 7-302-10561-8/TP·7162
印 数: 1～3000
定 价: 19.80 元

序　言

本书的目的

自从统一过程(Unified Process)问世以来，我在 UML World 和各种公共论坛(例如 Rational 的对象技术用户组(Object Technology User Group，OTUG)邮件列表)中听到很多人谈论 UP(Unified Process，统一过程)，都说 UP 真的很庞大，很复杂。如果是与其他著名的过程相比，我同意上述观点，UP 真的很庞大，但我认为，总体来说，UP 仅仅是非常复杂。

在 *UML Explained* 的第 2 章，我花了 10 页来讲述隐藏在统一过程背后的基本概念。当我还在编写那本书时，我突然发现，可以再写一本并不比那本书厚多少的书来详细介绍有关统一过程的最重要的内容(也就是 200 页，而不是我习惯的 150 页左右)。因此，我开始编写本书的部分章节，试图揭示统一过程所包含的、普通人很难完全理解的概念，此外还试图说明统一过程并没有规定，在一个项目中人们不应该做或者必须设法完成的任务。

结果是，我专门将此书构思成 *UML Explained* 的姊妹篇。不是试图教授大量采用统一过程的 UML，而是插入许多引用，指向那本书中的某些章节，这些章节提供了各种 UML 图和技术的详细信息，这些 UML 图和技术在统一过程中均占有一席之地。我还从那本书中带了许多图到本书中，帮助读者融会贯通地理解这两本书。正如毕加索所说，“优秀的画家会借；伟大的画家会偷”。

下面是本书其他一些主要特色。

- 我构建了其他 UP 图书中很少提及的域模型和业务模型，此外还设计了从属于需求工作流的各种角色、相关的工件和活动。
- 我将项目管理素材的数量减少到最低限度。Walker Royce 的 *Software Project Management: A Unified Framework*(Addison-Wesley，1998)是关于如何使用统一过程实施项目管理的权威著作，所以我想没有必要再多说什么了。
- 第 7 章~第 9 章讲述了如何设计和构建作为示例的 Internet 书店。然而，第 2 章~第 6 章中包含了 *UML Explained* 书中与本例有关的图，后面的章节则阐述了项目团队在执行迭代和递增开发中如何将系统分解为若干块。*Applying Use Case Driven Object Modeling with UML*(Rosenberg 与 Scott, Addison-Wesley, 2001)中介绍了关于该书店项目的其他问题。(我的态度是：相信你所知道的一切！)

我的目标是编写这样的一本书，能够帮助那些喜欢将 UP 称为“真正庞大的过程”的人们揭开统一过程的神秘面纱。但愿你能认可我的努力。

本书的组织

本书正文按内容可以分成 4 个部分。

第 1 部分就是第 1 章。第 1 章简明扼要地介绍了统一过程，其中包括统一过程的简要描述、历史、主要主题(用例驱动、体系结构中心、迭代和递增升级)以及对主要术语(工作流、阶段、迭代和递增升级、工件、工作人员和活动)的定义。

第 2 部分由第 2 章~第 6 章组成。这几章详细介绍了统一过程定义的 5 种工作流(需求、分析、设计、实现和测试)。每章都包含以下内容。

- 简介，简要地介绍工作流中包含的内容及其主要目标。
- 关于该工作流所涉及的各种工件的描述。
- 关于人们在该工作流所扮演的各种角色(用术语“工作人员(workers)来表示)的描述。
- 关于工作人员在工作流期间为生产工件而执行的各种活动的描述。

每一个“活动”小节都包含展示给定工作流非线性特性的 UML 图。图中的实线显示了活动执行的逻辑顺序；在某些示例中，一个活动是另一活动的基本前提，在其他示例中则不然，团队为某个活动执行的工作可能会导致返工到以前执行的一个或多个活动。虚线表示数据流。此外，在一个活动完成并反馈给下一个活动或上一个活动时，还讲述了工件的内容。

第 3 部分由第 7 章~第 9 章组成。这几章详细介绍了统一过程定义的 4 个阶段中的 3 个(初始阶段、细化阶段和构造阶段)。每章都包含以下内容。

- 简介，简要地介绍了项目团队在该阶段要执行的任务(包括该阶段的主要目标)，并从较高的层次上俯瞰每个工作流如何交叉贯穿该阶段(你可能会想像将工作流和阶段组成一个矩阵：用垂直轴代表工作流，方向向下；用水平轴代表阶段，方向向右)。
- 描述在开发团队开始该阶段定义的活动之前项目经理应该执行的任务。
- 描述项目经理在该阶段执行的由若干工作流定义的活动。这些描述都是特定于该阶段所需执行的活动(项目执行活动的多少取决于相关的环境)，同时也是特定于我们的网上书店。此书店的项目团队在初始阶段执行了一次迭代，在细化阶段执行了 3 次迭代，在构造阶段执行了两次迭代。每章都描述了它们在每次迭代期间所执行的每项活动以及整个阶段中建立的各种模型。

每章都在适当的位置列出了该阶段的交付产品。

第 4 部分就是第 10 章。本章描述了交付阶段，即项目团队向顾客展示系统的阶段。本章的格式与第 7 章~第 9 章的格式相同。本章之所以被列为独立的一章，乃是考虑到与其他 3 个阶段不同，工作流活动并没有贯穿交付阶段，并且本章并不讨论书店项目。

本书结尾还包括下列内容。

- 附录 A，讲述 Rational Unified Process(RUP)相对于核心统一过程所增加的内容。
- 附录 B，比较和对比 RUP 与极限编程(eXtreme Programming，XP)的基本特性。
- 附录 C，介绍与统一过程同源的 ICONIX 进程。
- 参考文献，列出了我在正文中所提及的所有图书以及其他一些在正文中未提及的书，以供读者参考。
- 术语表，对书中提及的所有术语的定义。

致谢

在感谢 Luck 13 中参与 *UML Explained* 的部分成员的同时，我还要感谢 Luck 13 的其他成员：Ross Venables，是他帮助我领取稿费，支持我编写本书；Paul Becker，一直都是一位耐心而给我带来很多帮助的编辑；Doug Rosenberg，是他鼓励我继续工作，我不能不再次感谢他；我的审稿人(Jim Conallen、Joel Erickson、Jeffrey Hammond 和 Jeff Kantor)，感谢他们富有见地的评论；Daphne Head，无须多说；Ivar Jacobson，感谢他的鼓励；Russ Coleman，感谢他的宽宏大量 Kim Arney Mulcahy，感谢他使本书的外观更吸引眼球；还有我的两只小狗 Samson 和 Smokey，是它们在这偏僻的角落陪伴我写完本书。

Kendall Scott
Harrison, Tennessee
kendall@usecasedriven.com
http://usecasedriven. com

目　录

第1章 概 述

简介

统一过程(Unified Process)符合过程的一般定义：是团队为将一组顾客的需求转换为软件系统而执行的一组活动。然而，统一过程也是一种通用过程框架，可以根据项目的特殊需要和可用资源来增加或删除资源，从而自定义该过程。

Rational Unified Process(RUP)是一种专门设计的统一过程范例，向通用框架中加入了某些元素，附录 A 中介绍了这些元素。本书主要讨论一家网上书店，以此作为项目范例，介绍了如何量身定制统一过程：负责该项目的团队可以跳过某些并不能创造价值的活动。示例系统描述了应用统一过程的关键技术：应该使用那些能够为特定项目增值的元素，而忽略那些不能增值的元素。请查阅附录 C，该附录中根据从核心元素开始并按需添加其他元素的原则对统一过程进行了简化处理。

统一过程广泛运用了统一建模语言(Unified Modeling Language，UML)。UML 的核心是模型，在软件开发过程中，UML 是实际事物的简化模型，可以帮助项目团队理解软件内部复杂结构的方方面面。

设计 UML 的目的是帮助软件开发的参与者构建模型，使系统对团队可见，指定系统的结构和行为，构造系统并随时编写文档记录开发过程中所做出的决策。统一过程的许多任务都需要涉及到 UML 的使用，还需要至少一种模型。

请参见 *UML Explained* 的第 1 章以了解软件开发环境中的模型及其价值。

本章余下的几节将讲述如何应用统一过程，介绍隐藏在过程背后的关键原则(用例驱动、体系结构为中心以及迭代和递增)以及用于描述过程细节的词汇。

历史

统一过程源于 20 世纪 60 年代 Ivar Jacobson 在前爱立信公司的工作。Jacobson 和他

的同事使用多层“块(blocks)”结构构建了一个非常庞大的电信系统，低层的“块”作为高层子系统的基础(这些“块”与现在所谓的组件有些类似)。这支团队通过探索他们称之为“通信案例(traffic case)”(现在，在UML中称之为“用例(use case)”)的事物来构建低层块，所有后续的分析、设计、实现和测试工作或多或少都依靠那些通信案例的驱动。目前的UML中的许多关键元素(即各种图表)均源自前爱立信公司的工作，包括顺序图(那时叫做对象交互图)和活动图(前爱立信公司称之为状态转换图)。有一份名为软件体系结构描述的文档，文档中包含了方便开发人员之间以及开发人员和顾客之间交流的关键材料。

Jacobson在1987年自立门户，开始创建了一家公司，自己将其命名为Objectory AB。他和他的同事花了几年时间开发Objectory，这是一个过程，也是一种产品，这就是Rational Unified Process(下文将进行讨论)。他的著作*Object-Oriented Software Engineering*(Addison-Wesley，1995)，详细描述了Objectory过程，被视作面向对象(object-oriented，OO)团队的里程碑。这是第1本由著名的方法学家撰写并用实际的用例阐述自身理念的著作，其理念是：顾客需求是软件开发的最大动力。本书还阐述了统一过程为何首先将重点放在体系结构上(请参见后面的“体系结构中心”)。该过程的完整的在线版本在1995年与该书一起被提供给广大读者。

不久以后，Rational推出了Objectory AB。Jacobson及其众多同事开始推广Objectory过程的应用领域，例如项目管理和开发工具。那时，Grady Booch和Jim Rumbaugh也已经进入Rational工作。Booch几乎是Rational成立之初就加盟了，而Rumbaugh则从1994年下半年开始效力于Rational，这两人和Jacobson一起成为著名的“三人组”，领导着Rational Objectory Process(ROP)的构建工作，同时将Unified Method扩展成后来的统一建模语言(Unified Modeling Language)。

请参见*UML Explained*的第1章以了解UML的简要历史。

在ROP和UML项目的进展过程中，Rational忙于收购和兼并了许多公司，这些公司开发了许多软件开发工具。这些工具为ROP产品提供了附加价值，主要表现在需求管理(Requisite的工具，后来发展成Requisite Pro)、万能测试(SOA的软件，后来扩展成几个工具)以及其他领域，如性能测试、配置管理和变化管理。1998年，Rational把产品的名称改成RUP，附录A中介绍了统一过程(是RUP的核心概念)与RUP在产品上的区别。