



敏捷与秩序

—— RUP最佳实践

Per Kroll

著

Bruce Macisaac

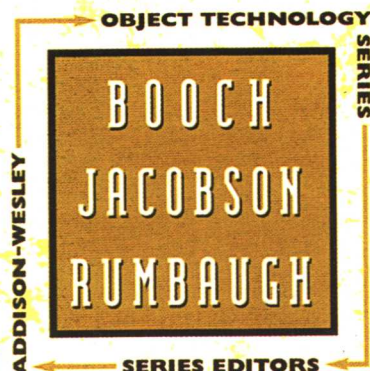
朱剑平

等译



eclipse

PROCESS
FRAMEWORK



清华大学出版社

敏捷与秩序

——RUP 最佳实践

Per Kroll
Bruce Macisaac 著
朱剑平 等译

清华大学出版社
北 京

Simplified Chinese edition copyright © 2006 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: **Agility and Discipline Made Easy** by Per Kroll & Bruce Macisaac, Copyright © **2006**

EISBN: 0-321-32130-8

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as **Addison-Wesley**.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 **Addison-Wesley** 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2006-4346 号

版权所有, 翻印必究。举报电话: **010-62782989 13501256678 13801310933**

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

敏捷与秩序——RUP 最佳实践 / (美) 克罗列 (Kroll, P.), (美) 麦希萨克 (Macisaac, B.) 著; 朱剑平等译.

—北京: 清华大学出版社, 2006.11

书名原文: **Agility and Discipline Made Easy**

ISBN 7-302-13963-6

I. 敏… II. ①克… ②麦… ③朱… III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2006) 第 120641 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

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市春园印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 230 印张: 18 字数: 398 千字

版 次: 2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷

书 号: ISBN 7-302-13963-6/TP · 8396

印 数: 1 ~ 3000

定 价: 36.00 元

献给 **Per** 可爱的女儿 **Natasha**，她几乎是在编写本书的同时被孕育的。

献给 **Bruce** 的祖母 **Mary MacIsaac**，在本书完成的时候她已经 112 岁高龄了。

序 言

本书的目的是描述一套定义明确的实践，这些实践是你和你的团队立刻就能采用的。你可以仅选择采用一种实践或在一段时期内采用全部实践。经证明，本书中的实践可以改进软件开发的质量、提高预见性、提升速度并降低成本。

为什么写这本书

在协助一些公司改进它们的软件开发实践和指挥 Rational 统一过程（Rational Unified Process, RUP）开发的十多年期间，我们有机会了解到哪些方法奏效及哪些方法不奏效。我们目睹了成功采用 RUP 所带来的好处，也看到了一旦采用这种方法后，项目和团队成员将要面临的挑战。在过去的几年中，随着敏捷开发对人员要求和软件开发中的低规范性的关注的增加，我们也从敏捷开发活动中学到了很多。最近，我们也成为一个开源项目——Eclipse 过程框架（Eclipse Processes Frame, EPF）中的主要操作人员，EPF 包括统一过程中的一个开源版本——开放统一过程（Open Unified Process, OpenUP）。EPF 的目标之一就是创建一个行业实践知识库，这个实践知识库可以用户化并且可以被装配成一套能满足一定需求的过程并反映不同的开发风格。通过这些工作，我们在什么方法奏效什么方法不奏效方面以及如何将这些知识包装成容易被采用的、实用的实践等方面已经获得了众多宝贵经验。

我们发现许多公司将改进它们的软件开发能力看做是一项不可思议的任务。似乎像是吃大象：从哪开始吃？怎么吃？答案是挑选一块嫩肉并且一次吃一口。“嫩肉”是指你最关注的地方；“一次吃一口”是指一种或者两种实践、软件规范或工具。换句话说就是吃“合适的量”。你不希望被噎着，而是想在吃下一口的时候还感觉有点儿饿。

本书描述了许多“合适的量”，它们会有助你从现在开始就改进软件开发能力。你可以根据自己的胃口，吃一口或者很多口。

我们希望提供一条简单的、持续的、面向改进成果的途径来降低最初与软件改进相关的焦虑和成本，而不是使你和你的团队不知所措。同时我们希望展示能够保持势头的早期成果，从而保持每一个相关成员的兴趣和奉献精神。

本书中所写到的实践与具体的过程无关。但是总的来看，这些实践覆盖了 RUP 和 OpenUP 的许多主要方面。我们相信这本书对于那些有兴趣部分或全部采用 RUP 和 OpenUP 的项目和公司来说将是一笔资产。本书中描述的实践都是从 RUP、OpenUP 和其他资源的知识中提炼出来的，目的是指导如何解决具体的软件开发问题。它们通过一次解

决一个问题来帮助你改进软件。每一个实践都描述了 RUP 和 OpenUP 如何帮你采用实践并且也涉及了其他方法，例如极限编程（eXtreme Programming, XP¹）和 Scrum²，从而使你了解它们与其他方法的共同之处和区别。对于那些对 RUP 和 OpenUP 不感兴趣的项目和公司，这本书也很有价值。你可以只选择这本书中的一些实践并单独采用它们。

你将从这本书中学到什么

这本书将使你熟悉以下内容：

- 许多软件开发的关键原则，这些关键原则已经被许多不同行业、不同规模的成功软件项目验证。
- 许多你和你的团队现在就能采用的实践，这些实践支持关键原则。所描述的每一个实践提供如下种类的信息：
 - ◆ 实践要解决的问题
 - ◆ 如何实际着手采用实践
 - ◆ 相关实践
 - ◆ 从哪里了解关于实践的更多知识
 - ◆ 如何以最低的风险循序渐进地采用实践，如何利用 RUP 和 OpenUP

谁应该读这本书

本书特别面向以下读者：

- 所有希望学习一些立即就能够使用的方法的软件开发团队的成员。
- 希望学习 RUP 或者 OpenUP 的团队（一次采用一种实践）。这本书不是让你理解 RUP 和 OpenUP 的所有内容，但是如果你理解了本书中描述的实践，那么在学习 RUP 和 OpenUP 的道路上你已经走得很远啦。
- 管理者、工程师和那些希望了解如何在他们的组织中采用关键实践（一次采用一种实践）的人。

本书的结构和内容

这本书分为 8 章。第 1 章概述作为本书结构基础的 6 项关键软件开发原则。第 1 章也概述了 RUP、OpenUP、XP 和 Scrum。

第 2~7 章依次回顾了这 6 项关键原则。对于每一项原则，我们都描述了许多支持它的实践。大部分实践可被单独采用，帮助你了解如何在开发中取得快速进展，而不用做很多变动。第 8 章描述了如何采用这些实践以及如何从这些实践中受益。

1 见 Beck 2004。

2 见 Schwaber 2002。

怎样阅读这本书

可以使用以下三种方法阅读本书：

1. 只阅读对你的团队有用的部分。从第 1 章开始阅读。确定你认为哪一项关键原则对你和你的团队最有价值。到相应的章节浏览那一章列出的实践，然后读你最感兴趣的内容。每一个实践要解决的问题是否与你面临的问题相同？这个问题是否值得确立？如果是的话，就与你的队友共同分享这些问题并在采用适合你们情况的实践上达成一致。一旦采用了你所确定的实践，通过继续研读来确定接下来的实践集。这种方法将在第 8 章中详细描述。
2. 阅读与你的角色相应的部分。要注意，由于软件开发是团队工作，最好要稍微熟悉一下与其他角色相应的实践。下面的列表确定了一些对软件开发团队中不同的角色来说可能最适合他们的章节和实践。

- 项目经理：阅读第 1 章、第 8 章和实践 1~4、7~10、12~13 和 19~20。
- 计划者：阅读第 1 章和实践 1、2、10~11、15~18 和 20。
- 分析人员：阅读第 1 章和实践 1~2、7~10、12 和 20。
- 开发人员：阅读第 1 章和实践 1~2、5~9、11~12、14~18 和 20。
- 测试人员：阅读第 1 章和实践 1~2、4~9、12 和 20。

3. 从头到尾阅读本书，尽可能多地获悉对所有团队成员都适用的实践。

更多信息

为了更多地获悉或者下载 EPF 和 OpenUP，请登录 Eclipse Foundation 的网站：www.eclipse.org/epf/。

RUP 是通过 IBM 公司的 Rational Method Composer (RMC) 的产品交付的。关于 RUP 或者 RMC 的更多信息（包括数据表和产品样本），可从 IBM 的网站获得：<http://www.ibm.com/software/awdtools/rmc/>。

关于 Rational Software Global User Group Community 的信息可从 <http://www.rational-ug.org/index.jsp> 获得。

学术机构可以与 IBM 联系以便得到在软件工程专业课程中有关特殊项目的信息，网址：<http://www.ibm.com/university/>。

致谢

这本书中的观点和思想来源广泛，我们从来自 IBM 内外的众多人才那里学到了很多东，这些人才帮助设计了 RUP 并且使其运行。我们也从敏捷方法、Eclipse 和各种过程方法中学到了很多东西。我们特别感激那些对我们的观点提出挑战、帮助我们完善本书的朋

友们。

如果没有 RUP 产品及其现在的团队就不能写出这本书。他们是：Amanda Brijpaul, Ricardo Balduino, Trevor Collins, Carlos Goti, Michael Hanford, Peter Haumer, Margaret Hedstrom, Kelli Houston, Russell Pannone, Thierry Paradan, Cecile Peraire, Dan Popescu, Michael Stern, Pawan Rewari, Jim Ruehlin, Jeff Smith, John Smith 和 Gordaon Schneemann。

过去几年当中，Rational 领域中的团队和技术专家在实现和应用 RUP 方面积累了大量经验。我们非常感谢来自这些专家的真知灼见，他们对工作中哪些方法有效哪些无效很有洞察力。我们要特别感谢 David Alvey, Kurt Bittner, Bill Cottrell, Peter Eeles, Bill Higgins, Kelli Houston, Saif Islam, Walker Royce 和 John Smith。

我们的外部评论者为我们提供了宝贵的反馈，使我们重新考虑本书的结构和内容，并满怀希望地帮助我们创作一本读起来既简明而且又能够处理大部分关于敏捷开发与原则问题的书。提出见解的这些人：Scott Ambler, Joshua Barnes, Lisa Crispin, Jim Dunion, Ivar Jacobson, Philippe Kruchten, Tom Poppendieck, Dan Rawsthorne, Chris Soskin, Christoph Steindl, Gargh Richmond, Dave Thomas, David Trent 和 Michael Vizdos。

特别感谢 Heath Newburn, Ted Rivera 和 Scott，他们通过专业的测试知识协助我们，每个人都为本书贡献了一个实践。

特别感谢 Catherine Southwood 和 Deborah Fogel，他们和 Mike Perrow 所做的工作一样，将我们的草稿整理成了合适的手稿。

写书要花费的时间总是比你想象的要长，我们要对我们的妻子和孩子 Susan、Natasha Kroll 还有 Kathy、Blaise Courtney 和 MacIsaac 表示感谢，感谢她们对我们无数个周末和夜晚花很多时间写书和改写所表示出的耐心。

最后，非常感谢我们的编辑 William Zobrist 在写作过程中为我们编辑；感谢 Mary O'Brien 为本书作记录；感谢 Addison-Wesley 的生产和营销团队，特别是 Tyrrell Albaugh, Stephane Nakib 和 Kim Silvestro 帮助我们出版这本书。

前 言

自第一支软件开发团队合作制造出第一个经济上可行的软件密集型系统以来，有关如何最好地组织团队和如何协调团队工作的哲学上的和实际应用上的论战就持续不断。一种极端是，有人提倡高正规性过程，包括严格定义的工件和精确的活动排序；另一种极端是，有人采用低正规性过程，包括极度地关注编码问题，而把任何其他问题都视为不相关或者不重要。多年以来，这种局面在两种极端之间摇摆不定。

好消息是，随着数十年来在软件密集型系统制造方面的经验积累，我们现在已经有可能识别是什么导致了成功的组织，并且摒弃那些失败的组织的通病。这是统一过程的精髓，无论是在 Rational 统一过程（Rational Unified Process, RUP）中还是在它的较简单的开源版本——开放的统一过程（Open Unified Process, OpenUP）中，RUP 和 OpenUP 都是一组组在实际中已经得到证实的最佳实践的精华。

在本书中，Per 和 Bruce 对作为 RUP 和 OpenUP 基础的 6 项原则进行了解释。对于每一项原则，作者都讨论了该原则要解决的问题、原则的背景以及对应用该原则的实际建议。因为对不同领域和文化发展的所有可以想象的结合，并没有任何一个单独的过程能以同样的方式奏效，所以作者进一步描述了每一项原则是如何适用于由低规范性到高规范性的不同应用的。

Per 和 Bruce 都非常适合这项工作，他们过去都从事大量的项目，并且都深入参与过 RUP 和 OpenUP 的制定过程。我认为没有比他们更合格的作者能带给你这些过程的精髓了。

Grady Booch
IBM 成员

关于作者



Per Kroll 是开源项目——Eclipse 过程框架项目的项目带头人，也是 RUP 和 IBM Rational Method Composer 的开发经理。他负责 IBM Rational 在过程方面的战略制定，包括方法和工具的结合以及 IBM 内部的方法结合。Per 在供应链管理、电信、通信和软件产品开发等领域已经有 20 年的软件开发经验了。他是非常受欢迎的 *The Rational Unified Process Made Easy* (Addison-Wesley) 的执笔者。

Per 和他的妻子 Susan、女儿 Natasha 还有金毛猎犬 Copper 生活在加利福尼亚的 Los Gatos。他喜欢跑步、打地板球（一种普通的北欧体育运动，大家集中在一起用杆越过其他中年男子的小腿之间击球）、远足和任何类型的棋盘游戏。他偶尔会喝点啤酒，喜爱跳舞。



Bruce MacIsaac 是 IBM Rational 的 RUP 和 OpenUP 内容开发的带头人兼管理者，在过去的 20 年中，他曾担任过各种管理角色和技术方面的角色，既带领过小规模的开发团队也带领过大规模的开发团队。最近，他开发了 IRUP 的第一个版本——源自于 RUP 的 IBM 内部定制的的程序，协助开发了 OpenUP 的最初版本并且在 Rational 方法创作中领导了 RUP 向其最新具体形式的移植。现在他负责实现 IBM 过程策略、整合可缩放框架中的现有方法并且开发满足不同项目和业务需求的新内容。

Bruce 和他的妻子 Kathy、两个孩子 Blaise 和 Courtney 于 2004 年从多雨的英属哥伦比亚省的温哥华来到了干燥的加利福尼亚的圣何塞。他喜爱滑冰、滑雪、网球和所有的舞蹈。尽管 Bruce 与 Per 喜爱很多相同的事物，但还是很容易区分他们——Bruce 的头发多一些而且舞跳得更好些。

目 录

第 1 章 利用关键开发原则	1
1.1 实践来自哪里	1
1.2 应用实践描述	2
1.3 采用实践：迭代开发、正规性级别和敏捷	3
1.3.1 实践采用的级别	3
1.3.2 过程图	4
1.3.3 敏捷性与正规性	5
1.3.4 采用级别将把你带到过程图中的哪些位置	6
1.4 关键开发原则	6
1.5 统一过程生命周期	9
1.6 OpenUP/Basic	10
1.7 Rational 统一过程（RUP）	12
1.8 极限编程（eXtream Programming, XP）	13
1.9 Scrum	14
1.10 小结	16
第 2 章 迭代展示价值	17
2.1 管理风险	19
2.1.1 问题	19
2.1.2 背景	19
2.1.3 应用实践	20
2.1.4 其他方法	25
2.1.5 采用级别	26
2.1.6 相关实践	26
2.1.7 额外信息	27
2.2 在迭代中执行你的项目	27
2.2.1 问题	27
2.2.2 背景	28

2.2.3	应用实践	32
2.2.4	其他方法	36
2.2.5	采用级别	37
2.2.6	相关实践	38
2.2.7	额外信息	38
2.3	采纳并管理变更	39
2.3.1	问题	39
2.3.2	背景	40
2.3.3	应用实践	41
2.3.4	与其他方法的比较	48
2.3.5	采用级别	48
2.3.6	相关实践	49
2.3.7	额外信息	49
2.4	客观地度量进展	50
2.4.1	问题	50
2.4.2	背景	50
2.4.3	应用实践	51
2.4.4	其他方法	57
2.4.5	采用级别	58
2.4.6	相关实践	58
2.4.7	额外信息	59
第 3 章	持续关注质量	60
3.1	测试你自己的代码	62
3.1.1	问题	62
3.1.2	背景	62
3.1.3	应用实践	63
3.1.4	其他方法	72
3.1.5	采用级别	73
3.1.6	相关实践	74
3.1.7	额外信息	74
3.2	适当利用自动测试	75
3.2.1	问题	75
3.2.2	背景	75
3.2.3	应用实践	77

3.2.4 其他方法	83
3.2.5 采用级别	83
3.2.6 相关实践	84
3.2.7 额外读物	84
3.3 产品属于每一个人	85
3.3.1 问题	85
3.3.2 背景	86
3.3.3 应用实践	87
3.3.4 其他方法	93
3.3.5 采用级别	94
3.3.6 相关实践	95
3.3.7 额外信息	95
第 4 章 平衡利益相关者优先级	97
4.1 了解领域	99
4.1.1 问题	99
4.1.2 背景	99
4.1.3 应用实践	101
4.1.4 其他方法	106
4.1.5 采用级别	107
4.1.6 相关实践	108
4.1.7 额外信息	108
4.2 从用户的角度描述需求	108
4.2.1 问题	109
4.2.2 背景	109
4.2.3 应用实践	111
4.2.4 其他方法	117
4.2.5 采用级别	117
4.2.6 相关实践	118
4.2.7 额外信息	118
4.3 按优先级实施需求	119
4.3.1 问题	119
4.3.2 背景	120
4.3.3 应用实践	121
4.3.4 其他方法	127

4.3.5	采用级别	128
4.3.6	相关实践	128
4.3.7	额外信息	129
4.4	利用遗留系统	129
4.4.1	问题	129
4.4.2	背景	130
4.4.3	应用实践	131
4.4.4	采用级别	136
4.4.5	相关实践	136
4.4.6	额外信息	136
第 5 章	团队间的协作	138
5.1	建立高绩效的团队	140
5.1.1	问题	140
5.1.2	背景	140
5.1.3	应用实践	142
5.1.4	其他方法	147
5.1.5	采用级别	148
5.1.6	相关实践	149
5.1.7	额外信息	149
5.2	围绕架构进行组织	150
5.2.1	问题	150
5.2.2	背景	150
5.2.3	应用实践	152
5.2.4	其他方法	156
5.2.5	采用级别	157
5.2.6	相关实践	157
5.2.7	额外信息	157
5.3	管理版本	158
5.3.1	问题	158
5.3.2	背景	158
5.3.3	应用实践	159
5.3.4	其他方法	166
5.3.5	采用级别	166
5.3.6	相关实践	166

5.3.7 额外信息	167
第 6 章 提高抽象级别	168
6.1 利用模式	170
6.1.1 问题	170
6.1.2 背景	170
6.1.3 应用实践	174
6.1.4 其他方法	176
6.1.5 采用级别	176
6.1.6 相关实践	177
6.1.7 额外信息	177
6.2 组件与服务架构师	178
6.2.1 问题	178
6.2.2 背景	178
6.2.3 应用实践	180
6.2.4 其他方法	183
6.2.5 采用级别	184
6.2.6 相关实践	184
6.2.7 额外信息	184
6.3 积极推进重用	186
6.3.1 问题	186
6.3.2 背景	186
6.3.3 应用模式	188
6.3.4 其他方法	192
6.3.5 采用级别	193
6.3.6 相关实践	193
6.3.7 额外信息	193
6.4 对主要观点建模	194
6.4.1 问题	194
6.4.2 背景	195
6.4.3 应用实践	196
6.4.4 其他方法	201
6.4.5 采用级别	201
6.4.6 相关实践	202
6.4.7 额外信息	202

第 7 章 调整过程	204
7.1 合理精简过程	206
7.1.1 问题	206
7.1.2 背景	206
7.1.3 应用实践	207
7.1.4 其他方法	211
7.1.5 采用级别	212
7.1.6 相关实践	213
7.1.7 额外信息	213
7.2 不断重新评价你在做什么	214
7.2.1 问题	214
7.2.2 背景	214
7.2.3 应用实践	215
7.2.4 其他方法	222
7.2.5 采用级别	223
7.2.6 相关实践	223
7.2.7 额外信息	224
第 8 章 实际应用最佳实践	225
8.1 我应该首先采用哪些实践	225
8.2 从基本采用级别开始	226
8.3 采用相关实践	229
8.4 RUP 与 EPF 如何能够帮助我	232
8.5 选择正确的试点项目	233
8.6 结束语	234
附录 A Eclipse 过程框架 (EPF)	235
A.1 为什么选用 EPF	235
A.2 什么是 EPF	235
A.3 潜在的 EPF 用户	236
A.4 可扩展的过程内容	238
A.5 软件过程工程元模型	239
A.6 可扩展的过程工程工具	243
A.7 参与 EPF 开发	245

附录 B IBM Rational Method Composer (RMC)	246
B.1 对于各种不同项目的过程	247
B.2 用于企业的过程	249
B.3 从业者如何使用 RMC	250
B.4 项目经理如何使用 RMC	250
B.5 过程经理如何使用 RMC	252
B.6 关于发展 IBM Rational Method Composer 的指导原则	253
术语表	255
参考书目	258