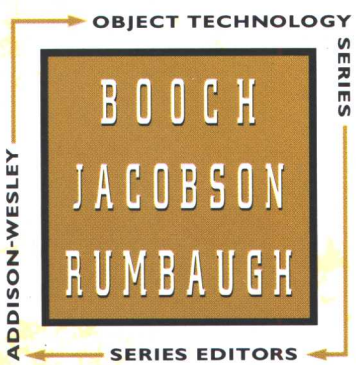
HZ BOOKS

软件工程技术丛书

Rational 统一过程引论 (原书第2版)

The Rational Unified Process An Introduction (Second Edition)



(美) Philippe Kruchten 著

周伯生 吴超英 王佳丽 译

机械工业出版社 China Machine Press

Addison-Wesley

软件工程技术丛书

Rational统一过程引论

（原书第2版）

（美） Philippe Kruchten 著

周伯生 吴超英 王佳丽 译

机械工业出版社
China Machine Press

Rational统一过程是由Rational软件公司开发和营销的一种软件工程过程，是开发组织用以分配与管理任务和职责的一种规范化方法，它能提高开发队伍的开发效率，并能给所有开发人员提供最佳的软件开发实践。

本书简明扼要地介绍了Rational统一过程的概念、结构、内容和动机。以本书为指导，开发人员可以在预定的进度和合理的预算范围内开发出高质量的软件产品。本书的作者是Rational统一过程这一产品的首席构架师，他在本书中与读者分享他所拥有的过程知识，并将重点放在掌握这种行之有效的软件开发方法的核心技术上。本书是为所有参与软件开发的人员撰写的，尤其适合那些已经或即将采纳Rational统一过程的开发组织的成员；同时，本书也可以作为广大读者学习软件开发相关课程的补充教材。

本书版权登记号：图字：01-2000-3393

图书在版编目（CIP）数据

Rational统一过程引论（原书第2版）/（美）克鲁奇特（Kruchten, P.）著；周伯生 等译. -北京：机械工业出版社，2002.5

（软件工程技术丛书）

书名原文：The Rational Unified Process: An Introduction, Second Edition

ISBN 7-111-09630-4

Ⅰ. R… Ⅱ. ①克… ②周… Ⅲ. 软件工程-建立模型 Ⅳ. TP311.5

中国版本图书馆CIP数据核字（2001）第089251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：姚 蕾

北京第二外国语学院印刷厂印刷 · 新华书店北京发行所发行

2002年5月第1版第1次印刷

787mm × 1092mm 1/16 · 16.75印张

印数：0 001-5 000册

定价：38.00元

译者序

50多年来计算事业的发展使人们认识到，要高效率、高质量和低成本地开发和生产软件，必须依靠高素质的人才、先进的技术和优化的过程。这三者是软件开发和生产的不可分割的三要素。早在20世纪50年代中期，人们就注意到人才和技术的因素，但是直到20世纪80年代中期，人们才认识到需要改善软件生产的过程。一些人认为，受过严格技能训练的人在工作时几乎不需要过程方面的指导，这种想法在绝大多数情况下是错误的，对于软件开发就更是如此。这就像一支交响乐队，单靠各个成员熟练的演奏技巧是远远不够的。每个乐队成员都必须遵循预先选定的乐谱，按照指挥的指令，协调地演奏，才能演奏出悦耳动听的音乐。软件开发群组是不能独立行动的，这不仅需要开发人员之间的密切配合，还需要与项目主管和用户的友好协作。因而，不仅要求有一个能指导开发人员协同工作的良好过程（乐谱），而且要求有一个既熟悉过程又具有管理才能的经理（指挥）。本书为读者提供了一个经过30年的发展和实际运用后在20世纪90年代中后期发展成熟起来的新一代软件开发过程。

一个过程定义了为达到某个确定的目标，需要什么人在什么时间以何种方式进行的一系列活动。用数学的语言来说，过程是这一系列活动的偏序集；其中每一个活动可用输入及其输入准则、输出及其输出准则、控制（或称约束）、资源（或称支持）、活动内容以及对活动性能的度量等8个元素来描述。一个过程需要对这8个方面的元素作出恰当、合理的规定。一个有效的过程是指能为有效地开发高质量的软件提供方针指导，是在当前技术条件下可行的最佳实践。按照这样的过程开发软件，所有与项目相关的人员都可以充分理解自己所扮演的角色，从而降低风险，增强预见性，在发生偏离时也比较容易进行纠正。

为了充分发挥过程在软件生产中的积极作用，过程工程师需要权衡与过程有关的技术、工具、人员和组织模式等四方面的因素。这种权衡不仅现在必要，而且在过程的整个进化过程中，都需要项目管理人员对这些因素进行动态权衡，以适应技术、工具、人员以及组织模式的变化，使其发挥最大的效益。

本书介绍的Rational统一过程（RUP）是《统一软件开发过程》这本书[⊖]中所描述的更通用的过程的一个特定、详尽的实例。通过本书，可以了解到RUP的概念、结构、内容和动机，认识到在这个过程中合成的最佳实践的优点，学会用RUP指导项目的管理。但本书仅是Rational统一过程的主要部分，有关RUP的各种技术的细节，可以从Rational软件公司的“在线知识库”获得。此外，本书是根据原书第2版翻译的，在内容的广度和深度上，都比原书的第1版有所扩充和改进。

本书是为所有参与软件开发的人员撰写的，尤其适合那些已经或即将采纳Rational统一过程的开发组织的成员。其中包括项目经理、开发人员、质量工程师、过程工程师、方法学专家、

⊖ 《统一软件开发过程》（*The Unified Software Development Process*）这本书是由Ivar Jacobson等著，其中文版由周伯生等译，机械工业出版社2002年1月出版。——译者注

系统工程师和分析员等。

在统一软件开发过程的进化中，最具贡献意义的两个因素是强调构架优先和迭代式开发。其中有关构架的文献一般都主张将构架表达为逻辑、过程、物理和开发等四个方面的视图，并加上一种用用况来阐述前面四种视图的视图。多视图法使得各类项目相关人员都能从适当的角度去发掘他们所想要的东西。其中有关生命周期的文献都主张采用初始、细化、构造和移交四个阶段，每个阶段包含一个或若干个迭代，并强制迭代按照一定的顺序进行。

现在软件产业界普遍认为，开发复杂软件项目必须采用基于UML的、以构架为中心、用况驱动与风险驱动相结合的迭代式增量开发过程，它是世界公认的开发复杂项目的最好的过程，已经成为软件界的“圣经”。这一开发过程目前已经稳定、成熟，然而由于过程是基于知识的创新，要将其精华转化为可用的技术，仍有相当大的一段距离，需要软件业内各界人士的共同努力以及有关主管的强有力支持。

我们翻译本书奉献给读者，目的是期望引起广大读者对这种方法的重视，共同为推进这种以构架为中心、用况驱动与风险驱动相结合的迭代式增量开发过程贡献力量，并注意对技术、工具、人员和组织模式等四种因素作出恰当的权衡，以便对我国的软件产业的发展和壮大起到有力的推动作用。最后，由于我们的水平有限，本书翻译中的缺点和错误在所难免，希望读者提出批评指正。

北京航空航天大学软件工程研究所
周伯生　吴超英　王佳丽
2002年3月25日

译者简介

周伯生，男，1935年生。北京航空航天大学计算机科学与工程系教授，计算机软件专业博士生导师，专攻软件工程与过程工程。1988年创办北航软件工程研究所；1984年2月—1986年7月，任国家科委与美国ISSI公司合作项目总体组组长；1986年1月—1995年12月，任国家七五和八五科技攻关软件工程及环境总体组成员；1991年10月以来，任北航与美国FunSoft公司“过程工程环境”合作项目技术负责人。

吴超英，女，1958年生，计算机软件硕士。北京航空航天大学软件工程研究所软件过程研究与实践项目组和软件质量保证项目组负责人。1982年以来，主要从事软件工程和过程工程领域的各项研究和环境的研制与开发工作。

前　言

Rational统一过程（Rational Unified Process，RUP）是由Rational软件公司开发和营销的一种软件工程过程，是开发组织用以分配与管理任务和职责的一种规范化方法。这个过程的目的是在预定的进度和预算范围内，开发出满足最终用户需要的高质量软件。

Rational统一过程综合了很多最佳的现代软件开发实践，用一种可剪裁的形式表达，以适应各种各样的项目和组织的需要；它以一种详尽、实用的方式将这些最佳实践在线提供给项目组。

本书介绍了Rational统一过程的概念、结构、内容和动机。

本书的目的

通过本书，你将：

- 学习什么是(以及什么不是)Rational统一过程；
- 掌握Rational统一过程的词汇并理解其结构；
- 认识到我们在这个过程中合成的最好实践的优点；
- 理解在一个项目中Rational统一过程如何向你提供完成特定职责所需要的指导。

本书中的内容是Rational统一过程的主要部分，但并不是它的全部，而只是其中很小的一个子集。在完整的Rational统一过程中，你可以找到进行开发工作所需要的详细指导。完整的Rational统一过程产品在线的“知识库”可以从Rational软件公司获得。

本书大量引用统一建模语言（UML）的内容，但它并不是一本介绍UML的书。如果读者希望了解UML，请参考以下两本书：《UML用户指南》和《UML参考手册》[㊀]。

这本入门书还讲授了建模和面向对象技术，但并不讲授设计方法，也不教你如何建模。有关嵌入RUP中的各种技术的详尽步骤和指南，只能在Rational软件公司的过程产品中找到。

本书有几章讨论了项目管理问题，描述迭代开发计划和风险管理等主题。然而本书决不是一个完整的有关项目管理和软件经济学的手册。如果要得到有关这方面的更多信息，请参考《软件项目管理：一个统一的框架》[㊁]这本书。

Rational统一过程是《统一软件开发过程》[㊂]这本书中所描述的更通用的过程的一个特殊而详尽的实例。

㊀ 《UML用户指南》（*The Unified Modeling Language User Guide*）和《UML参考手册》（*The Unified Modeling Language Reference Manual*）英文原版由Addison Wesley公司于1999年出版，中文版由机械工业出版社出版。——编辑注

㊁ 《软件项目管理：一个统一的框架》（*Software Project Management: A Unified Framework*）英文原版由Addison Wesley公司于1998年出版。——编辑注

㊂ 《统一软件开发过程》（*The Unified Software Development Process*）英文原版由Adddison Wesley公司于1999年出版，中文版由机械工业出版社出版。——编辑注

谁应该读这本书

《Rational统一过程引论》（第2版）是为参与软件开发的各种人员撰写的，其中包括：项目经理、开发人员、质量工程师、过程工程师、方法专家、系统工程师和分析员。

本书尤其与那些已经采纳或即将采纳Rational统一过程的开发组织中的成员相关。通常，这些组织将会对Rational统一过程进行裁剪，以适应自己的需求；但是本书所描述的核心过程，在Rational统一过程的所有实例中都通用。

本书对于学生学习由Rational软件公司及其工业界和学术界的伙伴发表的专业教程也是一本很好的补充教材。它提供了这些教程中涉及的一些主题的相关内容。

本书假定读者已对软件开发具有一定的理解，但不要求读者具备编程语言、面向对象方法或统一建模语言（UML）方面更深入的知识。

如何使用这本书

本书各章采用了自然进展的组织结构。因此，在一个全部采用或部分采用Rational统一过程的开发组织中工作的软件专业人员应该顺序地阅读本书。

项目经理可以只读第1、2、4和7章。这些章节介绍了风险驱动的迭代开发过程的含义。

过程工程师或方法学专家可能需要裁剪Rational统一过程，并在他们的开发组织中建立起这种过程。他们应该认真学习第3章和第17章，这些章节描述了过程结构以及全面实现Rational统一过程的方法。

书的组织和特征

本书分为两部分。

第一部分介绍过程、内容、历史、结构和软件开发生命周期。它还描述Rational统一过程区别于其他软件开发过程的关键特征。

- 第1章：最佳的软件开发实践
- 第2章：Rational统一过程
- 第3章：静态结构：过程描述
- 第4章：动态结构：迭代开发
- 第5章：以构架为中心的过程
- 第6章：用况驱动的过程

第二部分给出各种过程构件或工作流的概况。每一章对应一个核心过程工作流。

- 第7章：项目管理工作流
- 第8章：业务建模工作流
- 第9章：需求工作流
- 第10章：分析和设计工作流
- 第11章：实现工作流
- 第12章：测试工作流

- 第13章：配置和变更管理工作流
- 第14章：环境工作流
- 第15章：实施工作流
- 第16章：典型的迭代计划
- 第17章：配置和实现Rational统一过程

大多数关于工作流的章节由六部分组成：

- 工作流的目的
- 定义和关键的概念
- 工作人员和制品
- 一个典型的工作流：活动的概况
- 工具支持
- 总结

书后的两个附录总结了从第7章到第15章中介绍的所有工作人员（过程中的角色）和制品（过程的工作产品）。最后提供了缩写词和术语表以及一个简短的带注释的参考文献。

获取更多的信息

关于Rational统一过程的资料，如数据表和可下载的演示版本，可以通过因特网从Rational软件公司的网站*www.rational.com/rup_info/*获得。

假如你已经在使用Rational统一过程，可以从Rational统一过程资源中心获取附加的资料，这个资源中心负责向合作伙伴提供额外的有用资料、更新和链接。从在线版本可以找到与资源中心的超级链接。

学术机构可以与Rational软件公司取得联系，获取某个特定计划中有关Rational统一过程的课程方面的资料。

第2版

《Rational统一过程引论》（第2版）按照“Rational统一过程2000”进行了更新。在这个最新版本中，Rational统一过程在广度和深度两个方面都有所扩充。在内容的广度上，增加了关于业务工程、非功能性需求管理以及多级分布式应用程序的开发与实施等内容。原有内容也有所改进，包括应用程序接口设计（尤其适用于开发有效的Web应用程序）以及使用模式与框架设计系统的内容，同时增加了开发实时和反应式系统的知识。测试工作扩展到从确认构架原型到验证交付产品的整个生命周期。最后，还增加了过程路线图，以概述如何将Rational统一过程应用于各种不同项目和各种不同技术。此外，在内容的深度上，扩展了关于过程制品、活动和阶段的检测表与指南等方面的内容。

致谢

Rational统一过程凝聚了来自Rational软件公司和其他各方面的软件专业人员的共同智慧。本书第2章介绍了这一过程的历史。但是要把各方面人员的共同智慧综合成一本书，即使是写这

样一本引言，也需要一批人投入很大的热诚和努力。在这里，我把他们一一介绍给读者。

Rational过程开发组的成员组合了Rational统一过程，并对本书许多章节做出了贡献。其中做出贡献的人与本书各章节的关系如下：

- Kurt Bittner对分析和设计以及项目管理和测试等章节做出了贡献，并开发了数据工程的有关内容。
- Maria Ericsson开发了业务工程和需求管理的有关内容，她曾是过程构架的管理员。
- Leslee Probasco对需求管理工作流做出了贡献。
- Stefan Bylund对分析和设计以及集成用户界面设计等章节做出了贡献。
- Håkan Dyrhage对过程的组织和结构及其实现和配置等章节贡献了很多设想，同时他还协调了在线版本的开发工作。
- John Smith扩展了RUP 2000的项目管理方面的内容。
- Jas Madhur对配置管理、变更管理和实施贡献了很多见解。
- Bruce Katz对过程的测试方面做出了贡献。
- Margaret Chan负责产品集成以及本书绝大多数原图的汇编工作。
- Debbie Gray是这支横跨9个时区的工作组的尽职尽责的行政助理。

我们非常感谢Grady Booch为本书撰写了第1章。

Per Kroll是Rational统一过程的营销经理，Paer Jansson是Rational统一过程的产品经理（最近由Matt Herdon与她共同管理）。Christina Gisselberg和Eric Turesson共同设计和开发了Rational统一过程的在线版本。Stefan Ahlqvist开发了用户界面设计的概念。Chinh Vo协助汇编了本书。

Dave Bernstein、Grady Booch、Geoff Clemm、Catherine Connor、Mike Devlin、Christian Ehrenborg (Dr. Usecase)、Sam Guckenheimer、Björn Gustafsson、Ivar Jacobson、Ron Krubek、Dean Leffingwell、Andrew Lyons、Bruce Malasky、Roger Oberg、Gary Pollice、Leslee Probasco、Terri Quatrani、Walker Royce、Jim Rumbaugh、John Smith和Brian White等人对Rational统一过程和本书进行了评阅，提出了很多见解，获益匪浅。

我们还要感谢Scott Ambler、Ensemble Systems、IBM Global Services和Content Integration等合作伙伴对Rational统一过程和本书做出的贡献。

特别要感谢Rational领域的所有工作人员，尤其要感谢我们的英国朋友Ian Gavin、Ian Spence和Mike Tudball，他们始终对Rational过程怀有极大的兴趣。

Frenglish和Swenglish这两个词是由Joy Hemphill和Pamela Clarke铸造的。

最后要深深感谢本书的编辑J. Carter Shanklin、Kristin Erickson、Marilyn Rash与她的工作小组以及Addison Wesley Longman公司的所有工作人员，正是由于他们的努力才使本书尽快问世。

Philippe Kruchten
Vancouver, B.C., Canada

目　录

第二部分 过程工作流

第一部分

过程

第1章

最佳的软件开发实践

——由*Grady Booch*撰写

在这一章中，我们探索了最佳的软件开发实践，并对Rational 统一过程做了大体的介绍。

1.1 软件的价值

软件是运营现代化的业务、政府管理以及加强社会各方联系的动力源泉。软件以一种前所未有的方法和形式帮助我们产生和获得信息，并使信息可视化。从全球范围来看，软件业的飞速发展推动着世界经济更快地向前发展。从个人角度来看，软件密集型产品可以治愈疾病，使聋哑人听到声音，使残疾人可以行动，使不健全的人获得更多的机会。总之，软件已经成为这个世界不可或缺的一部分。[1]

对于软件专业人士来说，世界经济越来越依赖于软件是一个好消息。各种在技术上可行且有社会需求的软件密集型系统在规模、复杂性、分布和重要性上都在不断扩张。

这种扩张的不利一面是使我们熟悉的软件开发方法受到了限制。同时将传统的系统升级为新的技术产品也会带来一系列技术和组织问题。综合而言，随着软件产业越来越快的发展，人们需要更高的生产力和产品质量。但同时，合格的开发人员的数量又跟不上这样的迅猛发展。

网络使生产和维护软件产品变得越来越难；以循环的、可预测的方式生产高质量的软件仍很困难。

1.2 软件开发问题的症状和根本原因

不同的软件开发项目在不同的方面有着不足和缺陷，并且不幸的是，有太多的项目最终都失败了；但是我们可以从这些项目中找出一些共同的症状:[1]、[2]

- 对于最终用户的需要理解得不够精确

---

[1] Grady Booch. "Leaving Kansas" *IEEE Software* 15(1) 1998(1~2): 32~35.

[2] Gaper Jones. *Patterns of Software Systems Failure and Success.* London: International Thompson Computer Press, 1996.

[3] Edward Yourdon. *Death March: Managing "Mission Impossible" Projects.* Upper Saddle River. NJ: Prentice-Hall, 1997.

- 对需求的改变束手无策
- 程序块不兼容
- 软件不易维护或不易扩展
- 对项目严重缺陷的发现较晚
- 软件质量低劣
- 软件性能令人无法接受
- 开发组中的人员按各自的方式进行开发，如果有人改变了部分软件，将很难再进行重组
- 一个不可靠的构造和发布过程

不幸的是，治愈这些症状不等于治愈疾病。例如，对项目严重缺陷的发现较晚只是一个大问题的一个症状。这个大问题可能是对于项目状况的评估过于主观，并且没有发现项目需求、设计和实现中的不一致。

尽管不同的项目以不同的方式失败，但是似乎大多数的失败是由于以下几点根本原因综合造成的：

- 特别的需求管理
- 模糊和不精确的交流
- 脆弱的构架
- 过度复杂
- 未检测出需求、设计和实现之间的不一致
- 测试的不足
- 对于项目状况的评估过于主观
- 未解决存在的风险
- 无法控制变化的产生和传播
- 自动控制不足

1.3 最佳的软件实践

如果解决了这些根本问题，那么不仅治愈了那些症状，而且可以更好地以一种循环的、可