软件工程技术丛书

HZ BOOKS

统一过程最佳实践
移交和产品化阶段

软件开发过程系列

（加） Scott W. Ambler
（澳） Larry L. Constantine 等著
兰雨晴 雷雷 高静 等译

The Unified Process Transition and Production Phases
Best Practices in Implementing the UP

机械工业出版社
China Machine Press

软 件 工 程 技 术 丛 书

统一过程最佳实践

移交和产品化阶段

软件开发过程系列

(加) Scott W. Ambler 等著
(澳) Larry L. Constantine
兰雨晴 雷 雷 高 静 等译

The Unified Process Transition and Production Phases

Best Practices in Implementing the UP

机械工业出版社
China Machine Press

本套书共有四本，其中介绍的最佳实践方法分别对应统一软件过程的四个阶段：初始阶段、细化阶段、构造阶段、移交和产品化阶段。本书是这套书的第4本，重点介绍与统一软件过程移交和产品化阶段有关的最佳实践，包括需求工作流、分析和设计工作流、实现工作流、测试工作流、部署工作流、操作和支持工作流、配置和变更管理工作流、项目管理工作流和基础设施管理工作流等内容。

本书可以作为软件项目管理人员、软件开发工程师、过程工程师、系统工程师等专业人员的指导用书，也可作为高等院校计算机及相关专业学生的参考书。

Scott W. Ambler, Larry L. Constantine, et al: The Unified Process Transition and Production Phases: Best Practices in Implementing the UP（ISBN 1-57820-092-X）

本书版权登记号：图字：01-2003-3913

图书在版编目（CIP）数据

统一过程最佳实践　移交和产品化阶段/（加）安布勒（Ambler, S. W.），（澳）康斯坦丁（Constantine, L. L.）等著；兰雨晴等译. －北京：机械工业出版社，2006.1

（软件工程技术丛书　软件开发过程系列）

书名原文：The Unified Process Transition and Production Phases: Best Practices in Implementing the UP

ISBN 7-111-17777-0

Ⅰ. 统…　Ⅱ. ①安…　②康…　③兰…　Ⅲ. 软件开发　Ⅳ. TP311.52

中国版本图书馆CIP数据核字（2005）第127843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：盛海燕　李云静

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2006年1月第1版第1次印刷

787mm × 1092mm 1/16 · 15.75印张

印数：0 001-3500 册

定价：35.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

译者序

软件产业的不断发展，使人们越来越认识到优化的过程在软件开发和生产过程中的重要作用，遵循良好的过程是高效率、高质量和低成本地开发和生产软件的必由之路。

近年来，我国软件业发展迅猛，在众多的软件项目和软件产品的开发过程中，人们越来越意识到优化的软件过程对于保证这些软件项目和产品质量的重要性，围绕软件过程改进进行的专题培训、企业内部培训等越来越多，业界的一些专家、学者围绕这一领域编著、翻译的书籍也很多。然而略感遗憾的是，人们在认识到软件过程重要性的时候，对如何有效且高效地实践这些过程却还未找到最佳方法，这方面的专著目前也比较匮乏。

在笔者翻译的本书及本套丛书的其他书[⊖]中，详细介绍了作者在他们大量的软件过程经验和收集的来自于10余位业界杰出人物以及10多年的《软件开发》和《计算机语言》杂志的更广泛经验的基础上提出的软件过程最佳实践方法，并将这些方法和统一过程的各阶段对应起来。

在本系列书中，作者综合Rational 统一过程（RUP）、OPEN联盟的OPEN过程 、面向对象软件过程（OOSP）、极限编程（XP）等软件过程形成了一个处理真实世界开发和产品需要的更完整、更健壮的统一过程。在详细阐述一个具有更完整的增强生命周期的统一过程之后，每一卷书介绍了当前实现统一过程各个阶段（初始、细化、构造、移交和产品化）最佳实践的大师们的经验智慧集合。通过本书作者提供的资源链接，读者可获得更广泛的“在线知识库”。

本书介绍的最佳实践方法与统一软件过程的移交和产品化阶段相对应，介绍了需求工作流、分析和设计工作流、实现工作流、测试工作流、部署工作流、操作和支持工作流、配置和变更管理工作流、项目管理工作流和基础设施管理工作流的最佳实践。在移交阶段向用户群发布新系统时，项目组的焦点在于测试和确认完成的系统，与遗留系统并行运行系统，转换遗留数据库和系统以适应新系统，培训用户、操作人员、支持人员和维护开发人员，将系统部署成产品。在产品化阶段，项目组会关注运行系统并给使用它的用户提供支持，监控系统并进行适当操作以保证系统的连续运转，运

⊖ 本套书一共四本，另外三本分别为：《统一过程最佳实践 · 初始阶段》、《统一过程最佳实践 · 细化阶段》、《统一过程最佳实践 · 构造阶段》，均已由机械工业出版社出版。——编辑注

行和维护相关的工作、日志和支持系统，处理帮助请求，提供错误报告和用户的特征需求，管理变更控制过程。本书提供了恰当的部署、操作和支持新系统所必需的信息。

翻译本书的目的是为实践这一过程提供最佳实践参考。本书可作为软件项目管理人员、软件开发工程师、过程工程师、质量工程师、系统工程师、系统分析员等的软件过程实践指导用书，也可作为大专院校计算机及相关专业学生的软件工程实践参考书。

参加本书翻译工作的有兰雨晴、雷雷、高静、韩芳，全书由高静统稿，兰雨晴进行了审订。由于水平和时间有限，本书翻译中的缺点和错误在所难免，个别用词还有待商榷，真诚欢迎读者批评指正。

联系：Lanyuqing@buaa.edu.cn、LL@buaa.edu.cn、Gaojing@cse.buaa.edu.cn

北京航空航天大学软件工程研究所

兰雨晴　雷雷　高静

2005年秋

前　言

在《软件开发》杂志（Software Development）和它的前身《计算机语言》（Computer Language）中曾经刊登了大量关于如何成功开发软件的文章。为这一杂志撰稿的人包括许多业界最著名的专家，比如：Karl Wiegers、Ellen Gottesdiener、James Bach、Jim Highsmith、Warren Keuffel和Martin Fowler。简而言之，信息产业的大师们在这些年里一直在这本值得尊敬的杂志中与我们分享他们的智慧成果。

近来，几乎所有的组织对软件过程改进的关注越来越多了。从20世纪90年代中期开始，Rational公司控股和合并了其他一些软件工具公司；随着公司的发展，这些工具所支持的各种过程也被合并成一种开发方法，称为“统一过程”（Unified Process）。是否有可能让整个软件过程自动化？如果有可能，那么Rational公司是否拥有一套完整的工具集？对上述问题我们并不确定。但幸运的是，其他人也在定义软件过程，所以我们还可以从多个角度来看事物应怎样运作。这些过程包括：OPEN联盟的OPEN过程、面向对象软件过程（OOSP）的过程模式、极限编程（XP）和敏捷建模（AM）。这些不同的视角可以用来推动统一过程观点，使其更加健壮，结果就产生了一个更能准确反映你所在组织现实需要的增强的统一过程生命周期。因为我们相信《软件开发》中包含的多年积累下来的智慧能够用来充实统一过程——真正将我们产业的最佳实践统一起来，所以我们编写了本系列丛书。

为什么软件过程如此重要呢？让我们先设想一下：假如你想请人给你建造一间房子，让两位承包商来竞标。第一位承包商告诉你，通过使用一项最新的建筑技术给你盖房，如果从明天就开始的话，他能在两个星期内就把房子建好，造价只有10万美元。这个承包商手下有一流的木匠和水管工，他们以前用这项新技术建造过一个花园凉棚，他们愿意日夜加班以按期交付你的新屋。而第二位承包商告诉你，她需要先和你讨论你想要建一间什么类型的房子。然后，一旦她确定明白你的需要，她将在一个星期内提供一整套设计蓝图供你审阅和反馈。这个初始阶段只会花你1万美元，当你决定了最终方案，对于其余的工作她将给出详细计划和成本进度。

你会觉得选哪个承包商更让人放心呢？是想马上开始建房的那个，还是先搞清楚要建什么样的房子，再建模型，再详细计划，最后动工修建的那个？显然，后者更有可能理解你的需要——这是成功地交付给你一间符合你实际需要的房子的第一步。现在，设想你要构建的是软件——这通常是复杂好几个级别而且远比房子更昂贵的项目，再设想你还是面对两个与前面采取相同方法的承包商。选择哪个你会更放心呢？希望

你的回答仍是第二个；她有一个更明智的过程。但遗憾的是，实践显示：在大多数时间里，组织似乎喜欢选择第一个承包商的方法；任意删改过程。当然，实践也显示：在我们的产业里，建造大型的、具有关键任务的系统的失败率在85%以上。也许这两种现象有一定的关联。

现实世界中，形势甚至更糟。你可能尝试要建造一间房子，而所能用的所有承包商却都只有建造花园凉棚的经验。甚至他们仅仅在热带地区工作过，并且从来没有处理过霜冻地区的情况，但是你却生活在加拿大偏僻的森林地区。而且，不同级别的加拿大政府颁布不同的经常变化的法规，承包商根本就不熟悉这些法规。再次强调，第一个承包商的杂乱无章的方法有可能使人陷入麻烦之中。

移交阶段

在企业级统一过程（EUP）的增强生命周期中，移交阶段是5个阶段（初始、细化、构造、移交和产品化）中的第4阶段，每个软件的发布版本在其生命周期内都将遍历这些阶段。移交阶段的主要目标是把系统交给用户群体。要完成这步，需要做以下工作：

- 测试和验证完整的系统。
- 如果是可应用的系统，让它与其他要替换的遗留系统一起并行运行。
- 转换遗留的数据库和系统，使之支持新发布的版本。
- 培训用户、操作人员、支持人员和维护开发人员。
- 部署系统进入产品化阶段。

移交阶段要完成的最重要的事情是保证系统在产品化阶段能够运行。这不仅仅意味着系统是可操作的，还表明了系统能够实现用户的真实操作需要。你要保证你的系统不会损害其他系统，保证用户明白怎样使用系统并且在遇到麻烦时知道怎样寻求帮助，保证系统在用户需要时可以使用。

产品化阶段

系统移交到用户手中了，但是系统的生命周期并没有结束。事实上，大多数用户可能会说系统才刚刚开始它的生命周期。产品化阶段是EUP的5个阶段的最后一个阶段。产品化阶段的主要目标是操作系统并且支持使用该系统的用户。要完成这步，需要做以下工作：

- 监控系统，采取恰当的操作来确保系统连续运行。
- 操作和维护相关的工作、日志记录和支持系统。
- 对用户的帮助请求、错误报告和特性请求做出响应。
- 管理变更控制过程，使得缺陷和新的特性可以被划分优先级并分配到将来的发布版本中。

你要认识到仅仅交付一个系统使之进入产品化是不够的，你还要让它保持下去，这一点非常重要。EUP跟Rational统一过程（RUP）一样，是统一过程的一个实例，它包括产品化阶段，这是因为曾经交付给用户的操作和支持系统对于真实世界的组织来说太冷漠、太难以理解。最好的开发人员在构建一个系统时会考虑维护问题，同理，最好的组织在管理一组计算机系统时会考虑产品化问题。

本书向读者呈现了业界专家所撰写的描述软件领域最佳实践的文章。本书乃至本系列丛书的一个目标是提供已证实的统一过程所包含技术的可替代方案。另一个目标是弥补统一过程中的一些缺陷。因为统一过程是一个开发过程，而不是软件过程，它不可避免地遗漏或缺少了一些对软件专业人员来说非常重要的概念。幸运的是，《软件开发》杂志的作者们已经对过程范围有了更广泛的了解，并已经为我们弥补了许多缺陷。

关于本套丛书

本套丛书由四卷组成：第1卷介绍初始阶段，第2卷介绍细化阶段，第3卷介绍构造阶段，第4卷介绍移交和产品化阶段。每卷都已经独立成书，但是如果想对整个软件过程有一个完整的认识，你需要通读全套丛书。本套丛书的文章覆盖了整个过程，在每卷之间没有重复。我们在为本书选择材料时确实费了一番心思，有大量可以选择的材料，但是篇幅有限，缩小选择范围并不总是那么容易。如果时间和篇幅没有限制，那么每一本书都可能会有现在的两倍那么厚。通过缩小选择范围，我们相信留下来的文章一定都是精华中的精华。此外，为了增加每本书的重点内容，我们限制了每本书覆盖的工作流的数量。没错，大部分工作流对于每个阶段都是相关的，但正如表1中所指出的，为了提供范围更广的资料，每本书覆盖了工作流的一个子集。

表1　本套书的整体结构

工作流/主题	初始阶段（第1卷）	细化阶段（第2卷）	构造阶段（第3卷）	移交和产品化阶段（第4卷）
业务建模	X	X		
需求	X	X		
分析和设计		X	X	
实现			X	
测试	X	X	X	X
部署				X
操作和支持				X
配置和变更管理			X	
项目管理	X	X	X	X
环境	X			
基础设施管理		X	X	X

关于编者

Scott W. Ambler

作为《计算机语言》和后来的《软件开发》杂志的热心读者，我于1995年开始为杂志写稿，并在1997年终于成为对象专栏的作家。我从20世纪80年代早期开始从事软件开发，用Fortran和Basic等语言编写代码；到20世纪80年代中期，我开始使用Turing、C、Prolog和Lisp语言编码。在20世纪80年代末期，我发现生活中除了编程之外还有更多的东西，于是在用COBOL和几种第四代语言为IBM大型机编程的同时，我又开始学习用户界面设计、数据建模、过程建模以及测试等多方面的技巧。到了20世纪90年代，在我对结构化/过程化技术大失所望之后，我发现了对

象技术，并跳跃到Smalltalk开发，然后转为C++开发，最后又转回Smalltalk。

我曾先后在多个组织中担任顾问和架构师，之后我决定结合这些经验，并运用我在多伦多大学当助教时所获得的技巧，并且在20世纪90年代中期开始做职业培训。我很快学到了几件事情。第一，尽管我喜欢教培训课程（直到现在我还在做这项工作），但是我自己并不想全职做这件事。第二，也是更重要的，我学会了如何用简单易懂的方式交流复杂的概念，比如如何开发面向对象软件。这就促使我写下了我最初的两本书：《The Object Primer》（Cambridge University Press，1995/2000）和《Building Object Applications That Work》（Cambridge University Press，1997/1998）（获得Jolt生产效率奖），这两本书以开发人员的观点描述了对象技术的基本原理。然后我决定再写两本书来描述面向对象的软件过程（OOSP），这两本书是《Process Patterns》（Cambridge University Press，1998）和《More Process Patterns》（Cambridge University Press，1999），其中讲述了我在加拿大一家顶尖的对象技术咨询公司工作时所获得的来之不易的经验。从那时起，我帮助过业界的几个组织机构（大的和小的，新成立的和历史悠久的），帮助它们改善其内部的软件过程。我最新的作品包括本系列丛书以及与别人合著的《The Elements of Java Style 》（Cambridge University Press，2000）。我现在是Ronin International公司（www.ronin-intl.com）的总裁，这是一家提供软件过程和软件构架指导和咨询的公司，总部设在丹佛；同时我还是我自己的网站（www.ambysoft.com）的自由作家，在这个网站上我张贴了许多白皮书。我还是敏捷建模（AM）方法学（详见www.agilemodeling.com）的思想领袖。我想，我终于找到适合自己的小天地了。

Larry L. Constantine

我与《软件开发》及其前身《计算机语言》的合作是长久而富有成果的，但比起我与软件开发以及计算机语言打交道的时间，又是小巫见大巫了。从我在计算机的摸索时代所写的第一个FORTRAN程序开始，我就一直热衷于寻找能把事情做得更好并能帮助其他人做得更好的方法——正是这种兴趣让我不久就超越了技术领域而进入了管理和过程，以及设计和构建软件的可用性的本质问题。我在将近40年间在这一领域内摸爬滚打，一直在与经常把人与技术相分离的这种论调抗争。在我看来，软件开发的成功取决于对这两方面的理解和掌握，这一点也同样反映在我为杂志和其他刊物所写的文章中。我已经写了150多篇文章和论文以及14本书，其中包括现在这本与Scott Ambler合作编辑出版的书。经过Scott的同意，我自己在杂志中的一些文章也被收录到这套书中。其他的文章收录在《The Peopleware Paper》（Prentice Hall，2001）和《Managing Chaos: The Expert Edge in Software Development》（Addison-Wesley，2000）中。前者重印了我的长期专栏“人件”中的内容；后者集中了广受欢迎的“软件开发管理论坛”每期印在最后的精华。

在最近几年里，我的专业兴趣集中在增加软件的可用性上，这导致了我在以使用为中心的设计上的发展，也促使我与Lucy Lockwood合著的《面向使用的软件设计》（Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design）（Addison-Wesley，1999）一书的诞生。《软件开发》杂志评选该书为1999年最佳书籍，授予我们Jolt大奖。最近，我好像

穿越大洋比穿越河流还要频繁，因为虽然我住在美国，但是我同时还要到澳大利亚悉尼理工大学授课，我是这所大学计算机科学系的兼职教授。尽管题目是计算机科学，我讲授的都是管理和设计问题相结合的主题。我还是一位勤奋的职业培训师、设计师以及咨询顾问，帮助世界各地的客户建造更易使用的软件。我与Lucy Lockwood一起创立了Constantine & Lockwood有限公司（www.forUse.com），由我担任研发主管。现在除了原有的工作，我们正致力于将以使用为中心的设计与统一过程和统一建模语言（UML）结合起来。

目　录

第1章 概 述

什么是软件过程？软件过程是项目的状态、阶段、方法、技术以及人们用于开发和维护软件相关产品（计划、文档、模型、代码、测试用例、手册等）的实践的集合。我们需要的不仅是一个软件过程，而且是一个已经被证明在实践中有效的软件过程——一个可定制来满足特定需求的软件过程。

为什么需要软件过程？一个有效的软件过程可使组织在开发软件时提高生产率。首先，理解软件是如何开发的可以帮助我们做出更明智的决策，例如，不要过度依赖诸如SnakeOil v2.0这样的工具，该工具声称可以自动化软件过程各部分。诚然，工具很重要，但是它们必须支持并适合所选择的过程，并且其开销不能太大。其次，软件过程使得我们可以标准化投入、提高可复用性、再现性以及项目组之间的一致性。第三，软件过程对于引入代码审查、配置管理、变更控制以及构架建模等良好的行业实践提供了机会。第四，一个已定义的软件过程为更好的一致性和进一步的提高确定了基线。

一个有效的软件过程同样也会改进组织的维护和支持工作——也称为产品化工作。首先，它应该定义了如何管理变更并为软件将来的发布而恰当地分配了变更维护，以便使得变更过程更有效率。其次，它应该定义了如何将软件平滑地转变为操作和支持，以及操作和支持的工作如何得到实际的执行。如果没有有效的操作和支持过程，软件很快就会变成搁置品。

> 一个有效的软件过程不仅要考虑开发的需要还应该考虑产品化的需要。

为什么采用一个已存在的软件过程，或使用新的技术来改进现有的软件过程？摆在人们面前的现实是，软件越来越复杂，如果没有有效的方法来开发和维护软件，软件达到预期质量的可能性就会降低。不仅软件正在变得越来越复杂，而且需要同时开发多个软件。大多数组织要同时开发多个软件并且它们中不止一个处于产品化状态——这些项目需要进行有效的管理。此外，我们所构建的软件的特征也在不断的发生改变——从结构化技术所适用的20世纪70年代的简单批处理系统到面向对象的、基于构件技术所针对的交互式、国际化、用户友好、7×24小时服务、高事务率、高可用性的在线系统。并且除了这些之外，还要提高所交付的系统的质量，要尽可能复用以便软件开发更迅速和廉价。如果不能有效地组织和管理工作人员，那么这将是一个非常苛刻的要求。软件过程为实现这些目标提供了基础。

1.1 统一过程

统一过程是软件过程可能被实例化的框架（Jacobson,Booch&Rumbaugh,1999）。RUP是统一过程的一个实例，是Rational公司（Kruchten, 2000）最近的一项努力，该公司所引入的统一建模语言（UML）已经成为业界标准的建模符号。统一过程的核心是对象工厂过程，它是几年前Rational公司和Ivar Jacobson的对象工厂组织合并时获得的产品和服务中的一个。Rational公司用

它们自己的过程增强了对象工厂，并于1998年12月形成并发布了统一过程的最初版本（5.0）。

图1-1给出了由4个连续阶段和9个核心工作流所组成的最初的统一过程生命周期。沿着图表的底部可以看出贯穿整个统一过程的所有的开发周期都应该以迭代的形式组织。基本的思路是：工作组以迭代的方式在恰当的工作流中工作，以便各次迭代结束时都可产生和用户方共同工作的内部可执行产品。这样通过增加与用户的交互降低了项目的风险。另一个内置于统一过程中降低风险的技术是应该在各阶段末尾做出“继续/中止”的决策，如果项目将会失败，我们一定想尽早中止它。当然，最重要的决策点实际上是初始和细化阶段的末期（到构造阶段的末期再取消项目就为时太晚了）。对于大型关键项目的失败率达到80%～90%以上的行业（Jones, 1996），具备这一思路是非常重要的。

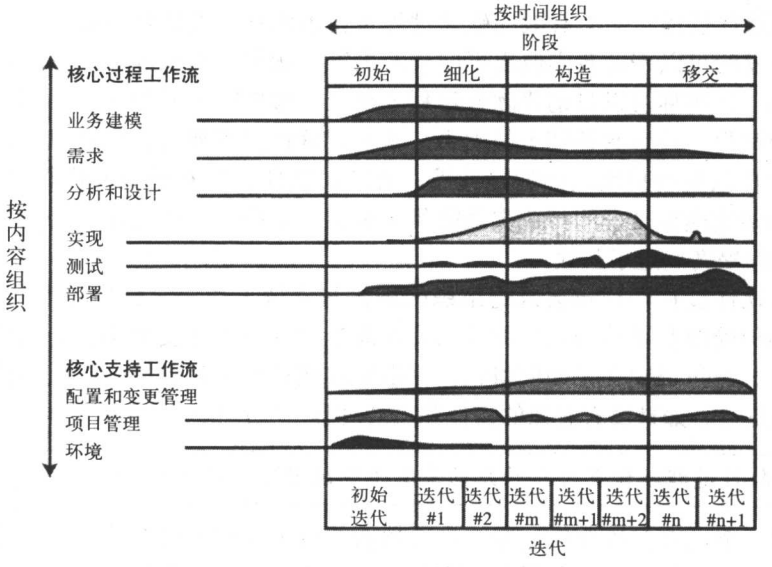

图1-1 最初的统一过程生命周期

初始阶段定义系统的项目范围及业务用例。在本阶段要确定软件的最初用例并简要描述关键的用例。用例是业界定义系统功能需求的一项标准技术。由于用例所关注的是系统对于用户的价值而不是对于产品特征，因此，它们比传统的需求文档显著地提高了生产率。基本的项目管理文档是在初始阶段开始的，包括最初的风险评估、估算及项目进度等。正如人们所预想的，本阶段的关键任务包括业务建模和需求工程以及环境的最初定义，包括工具的选择和过程的定制等。

> 初始阶段定义项目范围和业务用例。

细化阶段关注问题域的详细分析以及项目构架基础的定义。由于用例对于定义全部需求是不充分的，因此需要定义补充规格说明，用来描述系统所有的非功能性的需求。用于构造阶段的详细的项目计划也是在这个阶段制定的，详细项目计划的基础是初始阶段的管理文档。

> 细化阶段定义系统的构架基础。

构造阶段用于进行系统详细设计并同时生成相应的源代码。此阶段的目标是生产交付给用户的软件及相应的支持文档。项目团队所犯的一个最常见的错误就是将精力主要集中在这个阶段，由于组织没有为前两个阶段提供足够的资源投入，缺乏成功开发满足用户需求的软件的基础，因此通常这个错误对项目有害。在初始和细化阶段，应该为问题域和解决域的理解提供必要的资源。在构造阶段，应该避免出现重大的需求变更或新的构架方法，因为此阶段的主要目标就是构造系统。

> 构造阶段完成最后将部署的系统。

本卷所描述的移交阶段的主要目标是将系统交付给用户。通常会先为用户提供一个beta版本，在大多数企业中称为试验型发布——即在将软件发布给所有用户之前先由一小部分用户使用系统。在此阶段识别主要的缺陷并逐步进行修复。最后对投入进行评估，确定是否需要下一个开发周期以便进一步改进系统。这时，非功能性的需求成为最重要的问题，包括技术约束，比如性能方面的问题等。此时将主要关注压力测试、安装测试、系统测试等这类用于确认系统是否实现了其非功能需求的所有活动。

> 移交阶段交付系统。

统一过程有几个优点。首先，统一过程基于合理的软件工程原理，例如在开发中采用迭代、需求驱动、基于架构的方法，在开发过程中进行增量发布。其次，它提供了几种机制，比如每次迭代末期的工作原型，在各阶段末期的“继续/中止”决策等，它们使得开发过程具有管理可见性。第三，Rational已经并且将继续对其统一过程（RUP）产品进行大力投资（http://www.rational.com/products/rup）。RUP产品是对于可定制为满足需求的统一过程的基于HTML的描述。事实上，必须将它定制为满足自己需要的过程，因为RUP中有3000多页的HTML说明，远远超过任何一个项目或者组织所需要的活动。从RUP中选择要用的活动，然后用本系列书中所描述的最佳实践及其他定制过程所需的资源来改进过程，这对于组织来说将是非常有效的。仅仅接受模板中的RUP是不现实的，甚至可能导致失败。

统一过程同样也存在几个缺点。首先，它仅仅是一个开发过程。统一过程的最初版本并没有覆盖整个软件过程，图1-1所示，很明显它遗漏了当软件发布为产品之后的操作和支持的概念。其二，统一过程并没有明确地支持多项目基础设施开发，例如组织、企业范围构架建模，错过了组织内大规模复用的机会。第三，生命周期的迭代特性对于许多有经验的开发者来说是全新的，因此接受它们有一定难度，而且图1-1所示的生命周期图对此并没有提供帮助。

> 软件业有强大的自我欺骗能力。——Capers Jones

在本系列的第二本《统一过程最佳实践 · 细化阶段》（Ambler 和 Constantine，2000a）中详细描述了可以容易地增强统一过程来满足现实世界开发的需要。我们认为应该从对过程的需要开始，一个非常好的开始就是能力成熟度模型（CMM）。其次，应该看看竞争者，即在这方面是OPEN过程（Graham，Henderson-Sellers和Younessi，1997）（http://www.open.org.au）和面向对象软件过程的过程模式（Ambler 1998，Ambler 1999），看看可以从这些过程中复用哪些特性。最后，基于所学的内容给出一个增强的生命周期，并且用经过验证的最佳实践来支持该生命周

期。这个生命周期概述了统一过程的实例，我们称它为企业统一过程（EUP）。[⊖]

> 统一过程是一个好的开始，但是需要定制和增强以满足组织的特定需要。

1.2 企业统一过程（EUP）

EUP对标准统一过程（或RUP当前版本）添加了重要内容，把统一过程的范围重定义到全部的软件过程，而不仅仅是开发过程。这意味着需要加入操作、支持及维护的软件过程。其次，为了满足当前组织的充分需要，统一过程同样需要支持项目的组合管理——类似于OPEN过程中所说的“程序管理”和OOSP中所说的“基础设施管理”。增加以上两步形成了图1-2所描述的生命周期增强版。最后，需要用已经证明是良好的实践来充实统一过程，关于这点，可以在《软件开发》中找到相应的文章。

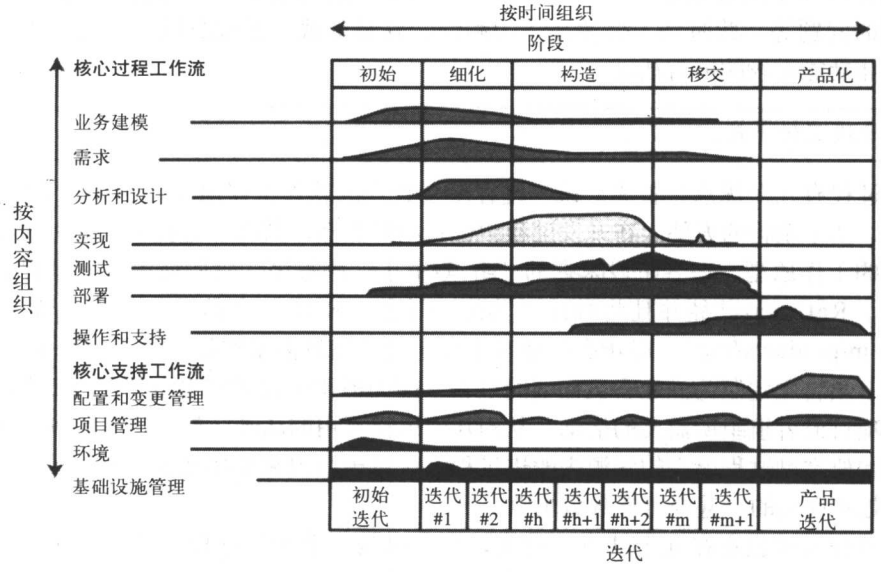

图1-2 增强的统一过程生命周期

EUP包括第5个阶段——产品化阶段，描述了软件生命周期在系统部署后的部分。因为软件生命周期平均要花费80%左右的时间在产品化阶段上，因此产品化阶段是实际的软件过程必备的一个特征。明确地增加产品化阶段会提高用于开发的生命周期的20%，因为这样做使得开发者清晰地认识到他们需要考虑产品问题，同时它也为通过项目向通用的构架而工作提供了更大的动力。正如此阶段其名字所暗示的那样，该阶段目的就是让软件一直保持在产品状态，直到它被更新的版本替代（从类似于修复bug这类的小的新版本发布到重要的新的版本发布），或者完成

⊖ 这里第一次使用EUP的概念，这个新名词源于我们认识到需要一个名字来说明我们的统一过程实例，“增强的统一过程生命周期”这个概念还不够。因为这个实例和其他实例（特别是RUP）之间的主要不同，是我们把范围定在实际企业而不仅仅是一个开发小组的需要，我们认为企业统一过程这个名词比较合适。

其使命退出产品状态。注意在这个阶段并没有迭代（或者说仅有一次迭代，这取决于如何看待它），这是因为这个阶段应用于软件某一发布版本的整个生命过程。要开发和部署软件的新版本，需要再重新遍历前4个开发阶段。

> 产品化阶段包括生命周期的部署之后的部分。

图1-2同样也说明了两个新的工作流：一个称为“操作和支持”的核心工作流，一个称为“基础设施管理”的支撑工作流。“操作和支持”工作流的目的正如其名字所暗示的，即操作和支持软件。操作和支持都是复杂工作，都需要为其进行过程的定义。此工作流和其他的工作流一样都跨越了几个阶段。在构造阶段需要开发出操作和支持计划、文档、培训手册。在移交阶段会持续地开发制品并基于测试的结果不断完善它们。最后，在产品化阶段，操作人员会让软件运行起来，进行必要的备份和所需的批处理工作，而支持人员会就软件的使用与用户进行交流。此工作流基本上包括了OOSP的发布和支持阶段以及OPEN过程的“实现计划”和“系统使用”两个活动。7 × 24小时是Internet经济要求的规则，高质量和高可用性是成功的关键，因此，需要操作和支持工作流。

> 操作和支持工作流对于确保软件的高质量和高可用性是必需的。

基础设施管理工作流专注于开发、发展和支持诸如组织/企业范围的模型、软件过程、标准、指南等此类的组织基础设施制品及其他可复用制品所必需的活动。软件的组合管理同样也是在此工作流中进行的。在所有的阶段都存在基础设施管理，细化阶段的标志是用于保证项目的构架恰当地反映组织全局构架的构架支持工作。这包括基础建模活动，比如企业需求/业务模型、域构架模型及技术构架模型的开发。这 3 个核心模型组成了基础模型，基础模型描述企业长期软件目标及共享/可复用基础。企业的软件工程过程组（SEPG）遵循的过程也包括在此工作流中，SEPG主要负责软件过程、标准和指南的支持和推进。此工作流也包括可复用的过程是因为实践证明，为了实现过程的高效性，必须进行跨项目的复用管理。想要获得规模经济开发、提高所开发软件的一致性和质量、增加项目间的复用，需要有效地管理公共基础设施，需要基础设施管理工作流。

> 基础设施管理支持诸如复用管理此类的跨项目/程序级的活动以及组织/企业级的构架。

将图1-2所示的增强的生命周期和图1-1所示的最初的生命周期进行比较会发现，几个已经存在的工作流也得到了更新。首先，测试工作流扩展为包括初始阶段中的活动。可以在此阶段制定初始的高层需求——可以使用类似于走查、审查及场景测试等技术进行需求确认。OOSP两个基本的原则是：第一，应该尽早测试和频繁测试；第二，值得开发的都是值得测试的。因此，测试应该贯穿于整个生命周期。同样，应该使用OOSP的“局部测试”和“整体测试”技术来增强测试工作流。这两个过程模式应用了面向对象测试全生命周期方法学（Ambler,2001）。

> 早测试，频繁测试。如果产品值得创建，那么它就值得测试。

第二个修改是针对部署工作流的——将它扩展到初始和细化阶段。这种修改反映了这样一个事实：部署，至少业务应用系统的部署，是一项让人畏缩的工作。遗留数据资源的数据转换通常其本身就是一个项目——需要重要的计划、分析及实施。而且，我们认为部署建模应该是