软 件 工 程 技 术 丛 书

HZ BOOKS

统一过程最佳实践
初始阶段

软件开发过程系列

（美） Scott W. Ambler
（澳） Larry L. Constantine 著
兰雨晴 高静 等译

The Unified Process
Inception Phase
Best Practices
in Implementing the UP

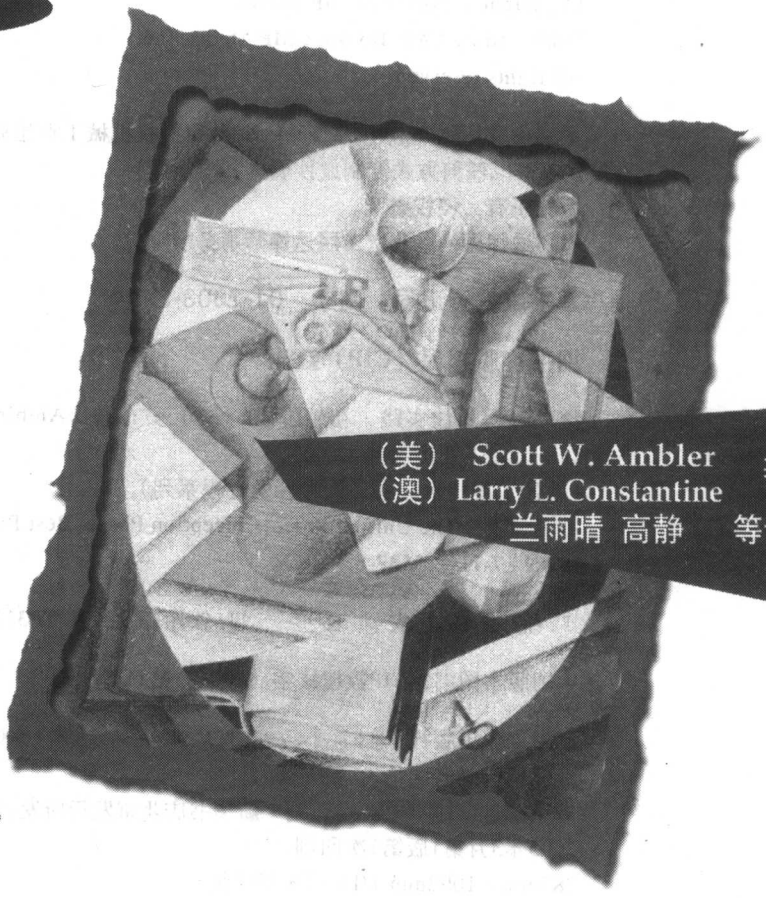
机械工业出版社
China Machine Press

软 件 工 程 技 术 丛 书

统一过程最佳实践

初始阶段

软件开发过程系列



（美） Scott W. Ambler 著
（澳） Larry L. Constantine
兰雨晴 高静 等译

The Unified Process
Inception Phase
Best Practices
in Implementing the UP

机械工业出版社
China Machine Press

本套书汇集了两位作者丰富的软件过程经验、10余位业界杰出人士的亲身体会以及《软件开发》和《计算机语言》杂志中的精彩论文，提出了软件开发过程中的最佳实践方法，指导读者有效而且高效地执行这些过程。同时，作者还综合了统一过程和其他软件过程，形成了一个处理真实世界软件开发和产品需要的更完整、更健壮的统一过程。

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。本书是这套书的第一本，重点介绍与统一软件过程初始阶段有关的最佳实践。

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: The Unified Process Inception Phase: Best Practices in Implementing the UP (ISBN 1-929629-10-9).

本书版权登记号：图字：01-2003-3914

图书在版编目（CIP）数据

统一过程最佳实践·初始阶段/（美）安布勒（Ambler, S. W.）等著；兰雨晴等译. -北京：机械工业出版社，2005.3

（软件工程技术丛书　软件开发过程系列）

书名原文：The Unified Process Inception Phase: Best Practices in Implementing the UP

ISBN 7-111-15537-8

Ⅰ.统…　Ⅱ.①安…　②兰…　Ⅲ.软件开发　Ⅳ.TP311.52

中国版本图书馆CIP数据核字（2004）第112255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：寇国华　姚　蕾

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2005年3月第1版第1次印刷

787mm×1092mm 1/16·15.75印张

印数：0 001-3000 册

定价：33.00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，人们也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，这方面的论著目前也比较匮乏。

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十余位业界杰出人物以及十多年的《软件开发》和《计算机语言》杂志提供的更广泛的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软件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）最佳实践的大师的经验智慧。而且，读者通过本书作者提供的资源链接，还可以获得更广泛的“在线知识库”。

本书介绍的最佳实践方法与统一软件过程的初始阶段相对应，介绍了业务建模工作流、需求工作流、测试工作流、项目管理工作流、环境工作流的最佳实践。初始阶段是所有阶段中最重要的一个阶段。在这个阶段，项目组打下成功或失败的基础。如果没有很好理解客户在功能性、可用性以及质量等方面的需要，那么后面的细化、构造以及移交和产品化过程开展得多么好都没有意义。如果以不清晰的业务目标、错误理解的用户需求、不切实际的计划和进度或不被看好的风险等作为开始，那么项目注定将失败。

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、高静、庞岩梅，全书由高静进行统稿，兰雨晴进行审定。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，Gaojing@cse.buaa.edu.cn

北京航空航天大学软件工程研究所

兰雨晴　高静

2004年中秋

序　言

在过去，一两个有才华的程序员就能够完全凭自己的力量创造有用的甚至是极富创造力的应用程序。但是，今天，要开发大多数应用程序，都需要其开发团队中具有多个掌握不同技能的专业人员，他们能够创建并优化代码及支持文档。高效的团队会慎重地选用适合他们项目特点、环境、约束以及企业文化的软件开发过程。

Rational公司的统一过程（Unified Process）是已经建立的现代软件开发过程之一。你可以买到详细描述统一过程的书籍，这些书会告诉你做什么，但是它们对如何有效且高效地实践这些过程所提供的指导却比较匮乏。在本书及本系列丛书的其他书中，编辑Scott和Larry弥补了这一不足，他们为本系列书带来了大量的软件实践背景、见识和见解。在本书中，他们收集了来自于10余位作者以及10多年的《软件开发》（Software Development）和《计算机语言》（Computer Language）杂志的更广泛的经验、见识和见解。本书中的41篇文章加上Scott和Larry附加的注释，为如何处理增强的统一过程生命周期中的第一阶段（初始阶段）提供了大量指导。这些文章中少数的几篇描述专门遵循统一过程的项目。但是，Scott和Larry还选了一些关键文章。这些文章强调了在软件开发过程中不断重现的重要主题——最佳实践，并将它们和统一过程的各阶段对应起来。每个软件工程师和项目经理都应该了解这些最佳实践，应该知道如何在自己项目所采用的过程中应用它们。

初始阶段可能是所有阶段中最重要的一个阶段，它处理项目的“模糊前端”。在这个阶段，项目组打下成功或失败的基础。如果没有很好地理解客户在功能性、可用性以及质量等方面的需求，那么后面的细化、构造，以及移交和产品化过程开展得多么好都没有意义。如果以不清晰的业务目标、错误理解的用户需求、不切实际的计划和进度或不被看好的风险等作为开端，那么项目注定将失败。

统一过程识别出跨越软件生命周期多个阶段的关键的过程工作流，包括需求、实现、测试和项目管理。本书包括的文章为如何实现初始阶段的各工作流的关键活动提供了具体建议。Scott Ambler和Larry Constantine同样意识到没有任何一个独立的过程或方法能够为软件开发组织碰到的各种挑战和情况提供一个公式化的解决方案。不过，本书将致力于帮助你应用统一过程到你的项目中。而且，即使你并不关心统一过程，这些文章也会为你思考当前软件工程中的最佳实践提供一个坚实的基础。

Karl E. Wiegers
Process Impact公司的首席顾问

前　言

在《软件开发》杂志和它的前身《计算机语言》中已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、Steve McConnell、Ellen Gottesdiener、Jim Highsmith、Warren Keuffel和Lucy Lockwood。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，在几乎所有的组织中，对软件过程改进的关注越来越多了。这一部分是因为千年虫（Y2K）问题、大规模软件项目的高失败率以及人们渐渐意识到遵循成熟的软件过程是软件项目成功的关键因素。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）的过程模式以及极限编程（XP）。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》中包含的多年收集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，后者更有可能成功地交付给你一间符合你实际需要的房子。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。但不幸的是，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。（在这种情况下，项目的失败被定义为严重超出成本预算或已经被彻底取消。）也许这两种现象有一定的关联。

实际上，问题甚至更糟。可能你试图造一间房子，而所能用的所有承包商却都只有盖花园凉棚的经验。甚至更糟，他们可能只在热带地区工作过，从来没有处理过霜冻地方的情况，但是你却生活在加拿大偏僻的森林地带中。更进一步说，他们根本不熟悉加拿大政府所规定的各种不同的法规，这些法规完全不同于他们曾经面对的简单常识。这个例子再一次说明，第一个承包商杂乱无章的方法有可能陷入麻烦之中。

初始阶段

在统一过程的增强生命周期中，初始阶段是5个阶段（初始、细化、构造、移交和产品化阶段）中的第1个阶段，每个软件的发布版本在其生命周期内都将遍历这些阶段。初始阶段的主要目标是为你的项目构造坚实的基础。要实现这点，需要：

- 证明系统本身和开发/获得该系统的方法是正确的。
- 描述系统的最初需求。
- 确定系统的范围。
- 确定和该系统交互的人员、组织及其他外部系统。
- 对系统进行最初的风险评价、进度安排以及估算。
- 对统一过程进行满足确切需要的初步定制。

当你回头再想的时候，你所需要做的最重要的事是确保你的系统以及应用于系统的方法都是被证明可行的（即，你有业务用例）。如果项目没有意义，可能是经济方面的原因，可能是技术方面的原因，也可能是可操作性方面的原因。不管是哪个方面的原因，项目都不应该继续。7/8的项目都失败了。如果没有坚实的基础，没有能够运作的架构，没有现实的项目计划以及专业的项目团队，那么你的项目很可能会成为7个失败项目中的一个。

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：第一卷介绍初始阶段，第二卷介绍细化阶段，第三卷介绍构造阶段，第四卷介绍移交和产品化阶段。每卷都可独立成书，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。

本套丛书的全局组织

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			X
基础设施管理		X	X	X

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。如果时间和容量没有限制，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。而且，为了增加各书中材料的论题，我们对每一本书覆盖的工作流数目进行了限制。是的，大多数工作流都是和各阶段相关的，但是正如前面的表所指明的，每本书都覆盖了一个子集，这些子集为你提供了那些工作流所涉及的非常广泛的材料。

关于编者

Scott W. Ambler

这是我最喜欢的话题了！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），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》的联系是经久而富有成果的，但比起我与软件开发以及计算机语言的联系，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，以及我们所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材料的理解和掌握，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专栏和文章也被收录到这套书中。其他的文章收录在《The Peopleware Paper》(Prentice Hall，2000）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《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的混和主题。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。

致谢

我要感谢下列人员，他们在本书的出版过程中给出了重要的意见和建议，他们是：Susan Ambler、John Nalbone、Mark Peterson、Neil Pitman、Doug Smith、Art Staden以及Robert J.White Jr。

目　录

第1章 简 介

什么是软件过程？它是项目的状态、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、测试用例及手册等）的实践集合。你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——一个可定制为满足你需求的软件过程。

为什么需要软件过程？一个组织采用有效的软件过程可以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你做出更明智的决定。例如，知道如何避开SnakeOil v2.0这一声称可以自动化软件过程各部分的令人困惑的工具。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，软件过程使你可以标准化投入，提高可重用性、再现性以及项目组之间的一致性；第三，软件过程为你引入代码审查、配置管理、变更控制以及构架建模等良好的产业实践提供了机会；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在很多方面改进组织的维护和支持工作——也称为产品化工作。首先，它应该定义如何管理变更并为软件将来的发布而恰当地分配变更维护，以便使得变更过程更有效率；其次，它应该定义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应该不仅考虑开发的需要，还应该考虑产品的需要。

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已存在的软件过程？目前的现实是，软件越来越复杂。如果没有有效的方法开发和维护软件，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。大多数组织在同一时间要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，我们所构造的软件特征也在不断地发生改变——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、基于构件技术所针对的交互式、国际化、用户界面友好、7×24小时服务、高事务及高有效性的在线系统。并且除了这些之外，还要提高所发布系统的质量，要尽可能地重用，以便可以用更少的资金更快地工作。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、并且几乎不可能实现的要求。软件过程为实现这些目标提供了基础。

> 软件越来越复杂，而不是越来越简单。

1.1 统一过程

统一过程是Rational公司（Kruchten，2000）的一项最新研究成果，同样一些专家所创建的统一建模语言（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

Rational公司几年前和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公司用其自己的过程增强了对象工厂，并形成了于1998年12月发布的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统一过程的最初生命周期。沿着图的底部可以看出贯穿整个统一过程的所有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工作在恰当的工作流中，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，你一定要尽早中止它。当然，最重要的决策点通常是初始和细化阶段的末期（到构造阶段的末期再取消项目已为时太晚）。对于大型关键项目的失败率超过80%~90%的行业（Jones，1996），具备这一思路是非常重要的。

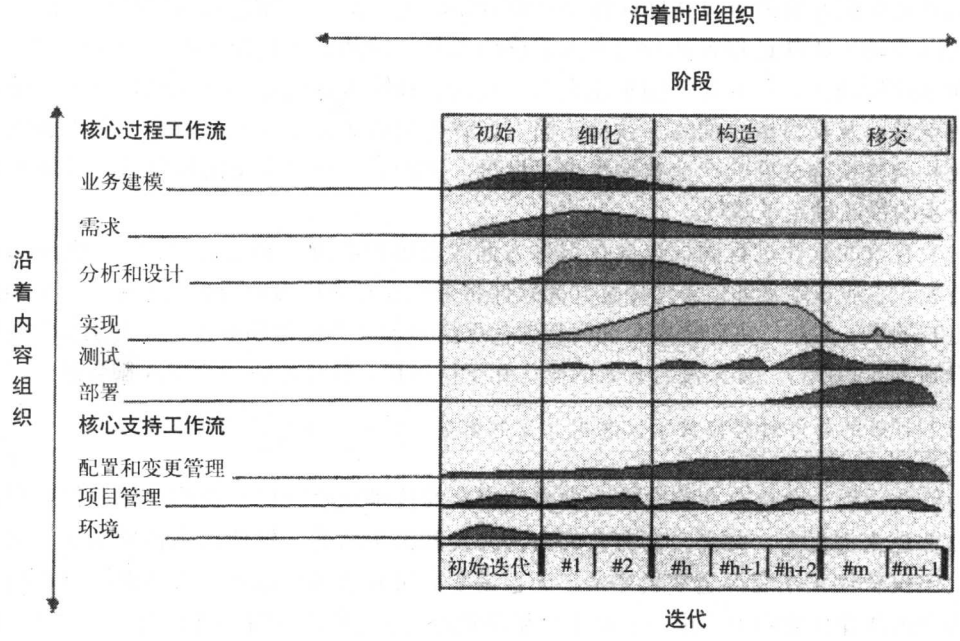


图1-1 统一过程的初始生命周期

初始阶段——本册讨论的重点，主要用于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的价值，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，包括最初的风险评价、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、工程需求、环境的最初定义，以及工具的选择和过程的定制等。

在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，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。

> 在细化阶段定义系统的架构基础。

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，项目组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基础，因此这个错误通常对项目有害。在初始和细化阶段，应该为问题域和解决方案域的理解提供必要的资源。在构造阶段，令人吃惊的事情（比如重大的需求变更或新的构架方法）应该少一点，因为此阶段的主要目标就是构造系统。

> 在构造阶段完成要部署的系统。

移交阶段的主要目标是将系统交付给用户，通常会先给用户一个软件的β版本——在大多数企业中称为引导版本发布（pilot release），即在将软件发布给所有用户之前先由一小部分用户使用。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。也就是在这时，非功能性的需求，包括技术约束（比如性能方面的问题），变得十分重要。此时将主要关注负载测试、安装测试及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，正如将在第3章中所说明的，开发需求远不止简单地编写用例。

> 在移交阶段交付系统。

统一过程有几个值得一提的地方，首先，它基于的是采用迭代、需求驱动、基于构架的方法来逐步递增式开发软件这样的软件工程原理；其次，它提供了几种机制，比如各次迭代末期的工作原型，在各阶段末期的“继续/中止”决策等，它们为管理层提供了可见性来监控开发过程；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，因为在3000多个HTML页面中所说明的远远不止是任何一个项目或者一个组织所需要的活动。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对于组织来说是非常有效的。如果脱离这些定制的规则来接受RUP，好一点的结果可能被认为是天真，坏一点的结果就很可能导致失败。

> 如果脱离定制的规则来使用RUP就很可能导致痛苦和冲突。
>
> ——Neil Pitman

统一过程同样也存在几个弱点。第一，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，正如在图1-1中所看到的，很明显它遗漏了当软件发布为产品之后操作和支持软件的概念；第二，统一过程并没有明确地支持多项目基础设施的开发工作（例如组织/企业范围构架建模），损失了企业内大规模重用的机会；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。

> 软件行业有强大的自我欺骗的能力。
>
> ——Capers Jones

我们将在本系列的第2本《统一过程最佳实践·细化阶段》中详细描述，人们可以更轻松地利用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——一个比较好的开始就是能力成熟度模型（CMM）；其次，应该看看竞争者——在这方面是OPEN过程（Graham, Henderson-Sellers, and 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重用哪些特性。图1-2描述了OPEN过程的契约驱动的生命周期，图1-3描述了由一系列过程模式[⊖]所组成的面向对象软件过程（OOSP）的生命周期。最后，应该基于所学的给出一个增强的生命周期，并且用已经证明是良好的实践来支持该生命周期。

> 统一过程是一个好的开始，但是它可能需要被定制和增强以适合企业特定需求。

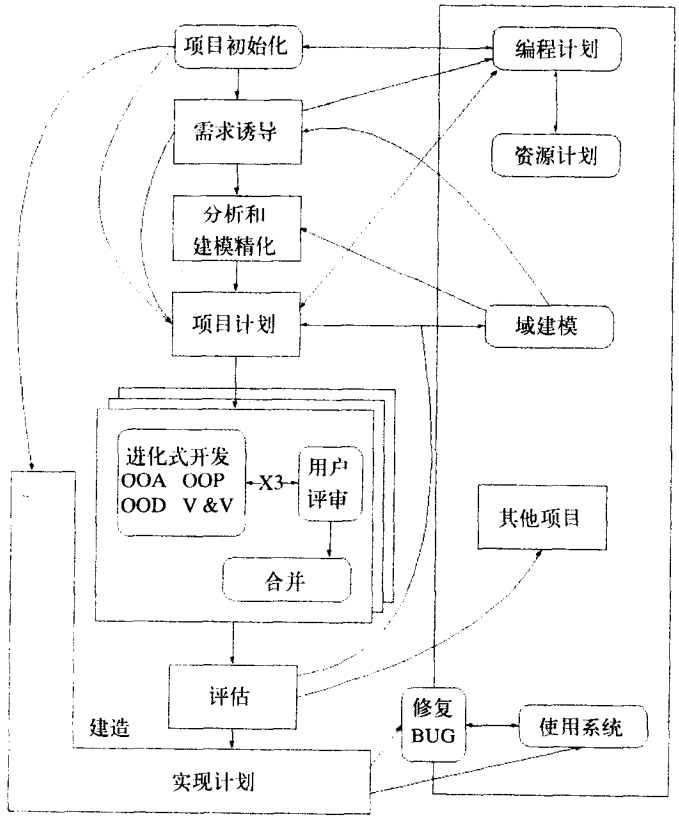


图1-2 OPEN过程的契约驱动的生命周期

⊖ 过程模式（process pattern）是一般的方法、行为或任务（活动）的集合，该集合通过考虑相关影响力/因素解决一个特定软件过程问题。如同设计模式描述了通常软件设计问题被证明了的解决方案一样，过程模式描述了通常的软件过程问题被证明了的解决方案。关于过程模式的更多信息可以在过程模式页面中找到：http://www.ambysoft.com/processPatternsPage.html。

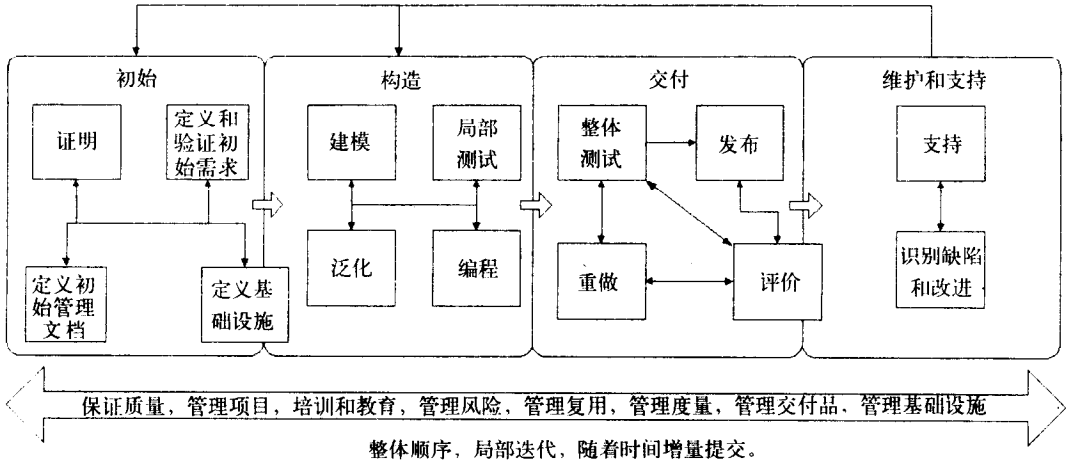


图1-3　面向对象软件过程（OOSP）生命周期

1.2　统一过程的增强生命周期

我们已经看到了一个成熟软件过程的需求概述以及软件过程的两个竞争类型。了解这些之后，应该如何完成统一过程？首先，重新定义软件过程的范围，使其包括完整的软件过程，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，若对当前组织来说是充分的，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”。以上两步带来了图1-4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》杂志中找到相应的文章。

增强的生命周期包括第5阶段——产品化阶段，代表在系统的一个版本部署之后的软件生命周期的一部分。因为软件平均要花费其生命所有时间的80%左右在产品化阶段上，所以产品化阶段是实际软件过程必备的一个特征。明确地包括一个产品化阶段会提高用于开发的生命周期的20%，因为这样做会使开发人员清晰地认识到他们需要考虑产品化问题，同时它也提供了更大的动力来努力创造一个多个项目都可使用的通用构架。正如此阶段的名称所暗示的，该阶段的目的就是让软件一直保持在产品化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版本发布——或者完成其使命退出产品化状态。注意：在这个阶段并没有迭代（或者说仅有一次迭代，这取决于你如何看待它），这是因为这个阶段应用于软件某一发布版本的整个生命期。要开发和部署软件的新版本，需要重新经历四个开发阶段。

> 产品化阶段包括生命周期部署之后的部分。

图1-4同样也说明了两个新的工作流，称为“操作和支持”的核心工作流以及称为“基础设施管理”的支持工作流。“操作和支持”工作流的目的正如其名字所暗示，即操作和支持软件。操作和支持都是复杂的工作，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覆盖