高职高专计算机规划教材·任务教程系列

UML与Rose建模应用

潘志安　袁　瑛　主编

项目导向
任务驱动

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机规划教材·任务教程系列

UML与Rose建模应用

潘志安 袁 瑛 主 编
陈希球 沈小波 陈 焜 副主编
李 岱 蔡 明 参 编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书较全面地阐述了 UML 及软件工程的各种概念、方法和新技术，在介绍 UML 建模语言基本理论的基础上，重点突出了 UML 建模语言的应用。

全书共分 4 个学习情境：学习情境 1“UML 与 Rose 认知”，介绍了 UML 的基本知识和 Rational Rose 2003 的安装及界面；学习情境 2“桌面系统建模——ATM 机”、学习情境 3“嵌入式软件建模——MP3 播放器”、学习情境 4“Web 软件建模——在线销售系统”则通过 3 个实际软件项目的建模，由浅入深地展示了 UML 建模过程和 Rational Rose 2003 的使用方法。

本书适合作为高职高专学校计算机应用与软件技术专业的课程教材或教学参考书，也可作为软件设计与开发人员的培训教材或自学参考书。

图书在版编目（CIP）数据

UML 与 Rose 建模应用 / 潘志安，袁瑛主编. —北京：中国铁道出版社，2011.1

高职高专计算机规划教材. 任务教程系列
ISBN 978-7-113-12369-7

Ⅰ.①U… Ⅱ. ①潘… ②袁… Ⅲ. ①面向对象语言－程序设计－高等学校：技术学校－教材 Ⅳ.①TP312

中国版本图书馆 CIP 数据核字(2010)第 257049 号

书　　名：UML 与 Rose 建模应用
作　　者：潘志安　袁　瑛　主编

策划编辑： 翟玉峰
责任编辑： 翟玉峰　　**读者热线电话：** 400-668-0820
特邀编辑： 田学清　　**编辑助理：** 贾淑媛
封面设计： 大象设计 • 小　戚　　**封面制作：** 白　雪
版式设计： 于　洋　　**责任印制：** 李　佳

出版发行： 中国铁道出版社（北京市宣武区右安门西街 8 号　邮政编码：100054）
印　　刷： 三河市兴达印务有限公司
版　　次： 2011 年 2 月第 1 版　　2011 年 2 月第 1 次印刷
开　　本： 787mm×1092mm　1/16　**印张：** 12.5　**字数：** 300 千
印　　数： 3000 册
书　　号： ISBN 978-7-113-12369-7
定　　价： 22.00 元

版权所有　侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社计算机图书批销部联系调换。

PREFACE

序

湖北职业技术学院《UML 与 Rose 建模应用》是一本按照工作过程系统化的课程论进行深度开发的专业教材。

本教材的深度开发体现在：

其一，课程目标的实用性和职业性强。通过限期合作，对软件需求分析员、软件设计员、数据库设计员等典型岗位的调研，选择源于软件项目的教学内容，而教学过程又模拟实际软件开发的工作过程，使得课程定位突出了职业能力和职业素质的培养。

其二，课程设计的逻辑性和科学性强。学习情境“UML 与 Rose 认知”、“桌面系统建模——ATM 机”、“嵌入式软件建模——MP3 播放器”、“Web 软件建模——在线销售系统”体现了从入门到熟练、从新手到专家、从单一到综合的职业成长过程，又体现了从简单到复杂、从外围到核心、从形象到抽象的学习过程，着眼于学习者的全面发展。

其三，教学活动的实践性和操作性强。教学设计按照“需求分析—用例分析—系统逻辑设计”的工作过程展开，通过 3 个典型项目的学习，以“做”为中心，“教、学、做、评”为一体，操作性强，有利于学习者在掌握实际项目的过程中逐步培养职业能力。

软件技术的学习难度较大，而本教材的开发和设计却在解决这一难题上迈出了坚实的一步，很有特色。

2010 年 10 月

FOREWORD

前言

20世纪末，面向对象技术得到了深入研究并被广泛应用，目前，面向对象技术已成为软件开发中分析、设计、实现的主流方法和技术。在面向对象技术发展的同时，伴随着面向对象技术的各种软件设计工具、规范等也获得了较大发展。其中，统一建模语言（Unified Modeling Language，UML）以其显著优势成为该领域的主流技术，并得到工业界和学者们的一致认可。

UML是可视化（Visualizing）、规范定义（Specifying）、构造（Constructing）和文档化（Documenting）的建模语言，它为设计人员、开发人员、用户和领域专家之间的交流提供了便利，已成为面向对象软件系统分析与设计的必要工具，并在IT界得到了广泛应用。而Rational Rose则是由Rational软件开发公司设计、开发的一种重要的可视化建模工具。

在我国的高等职业教育中，从传统的基于学科结构系统化的综合课程方案转向基于工作过程导向的学习领域课程方案，已成为高职教育课程建设与改革的主流方向。在这一背景下，行动导向教材的开发也是方兴未艾，本教材就是这一高等职业教育变革下的产物，它以软件行业与企业的人才需求为导向，重点分析“需求分析员”、“软件设计员”、“数据库设计员”等典型职业岗位的工作标准和应具备的职业能力，明确人才规格（知识、能力及素质），运用工作过程系统化理论和行动导向的教学原则，按照“项目驱动、过程贯穿、学做一体”教学模式来设计。

本教材遵循学生职业能力培养的基本规律，在对职业工作任务和工作过程的实际情况进行解析、归纳、重构的基础上，合理选择软件项目为载体，并根据教学论的原则进行理论系统化的处理，即将所选取的教学内容进行整合、序化，设计了“UML与Rose认知”、“桌面系统建模——ATM机”、“嵌入式软件建模——MP3播放器”、“Web软件建模——在线销售系统”4个学习情境。其中学习情境1介绍UML基本概念和Rose安装及基本使用方法，为后续情境中项目的实施作准备；后3个学习情境以3个软件项目为载体，项目来源于工作又高于工作，是实现教学论目标的情境化案例，体现了教学内容的实用性。3个项目以工作过程分别贯穿在3个学习情境中，具有很强的教学可行性。这3个学习情境包含了软件建模的所有内容，既有覆盖性、典型性，又有综合性、挑战性。

从梯度上来说，学习情境2是简单软件的建模，学习情境3是中等难度软件的建模，学习情境4是复杂软件的建模。后一个情境与前一个情境的教学过程相同，都包括系统需求分析、用例分析、静态结构建模、动态结构建模等，但每个环节所包含的内容有所增加。随着学习情境的深入，建模内容越来越多，难度也越来越大，但始终围绕模型绘制这一关键能力的训练，但又不是简单的重复，而是层层递进。重复的是建模的过程、步骤和方法，不重复的是建模的具体内容，在比较中进行学习能使学生熟能生巧、举一反三，体现了学习的可迁移性和成长性。

本书学习情境1由袁瑛编写，学习情境2由潘志安编写，学习情境3由沈小波编写，学习情境4由陈焜编写。沈平对本书的编写提出了很多宝贵意见，王英、叶芳华、姜焱、杜恒、罗肖、

试读结束：需要全本请在线购买： www.ertongbook.com

李岱、蔡明等对本书的资料收集、模型绘制等也做了大量工作。在本书编写过程中，还得到了湖北职业技术学院信息技术学院宋振云院长的大力支持和无私帮助。在此向所有对本书的编写和出版提供了帮助的人士表示衷心的感谢！

本书适合作为高职高专学校计算机应用与软件技术专业的软件建模课程教材或教学参考书，也可作为软件设计与开发人员的培训教材或自学参考书。

由于作者水平所限，书中难免存在疏漏和错误之处，恳请专家和广大读者批评指正。

编　者

2011 年 1 月

CONTENTS

目 录

学习情境 1 UML 与 Rose 认知

情境（项目）描述

某校朱老师带领 3 个项目小组进行课程设计，而课程设计需要使用 UML 相关知识来设计项目的模型图，方便项目小组学生进行项目开发。软件建模设计需要了解和掌握以下内容：

1. UML 的发展历程、内容、特点、功能及组成等概况。
2. 正确安装 Rational Rose 2003 软件，熟悉其框架结构与简单使用方法。
3. 了解 Rational Rose 的 4 种视图。

子情境 1.1 UML 概述

1.1.1 子情境描述

老师在指导学生课程设计之前，需先带领学生学习统一建模语言 UML。了解 UML 的特点、优势、历史及发展前景，熟悉 UML 的主要内容、功能、组成及其应用，以便为后面的学习打好基础、做好铺垫。

1.1.2 任务 1：统一建模语言（UML）

在面向对象分析与设计方法的发展过程中产生了 UML。为了对 UML 有一个大体的认识，必须了解与熟悉以下几个方面的内容：

- 面向对象方法的引入。
- UML 的兴起。
- UML 主要内容。
- UML 主要特点。
- UML 功能描述。
- UML 组成简介。
- UML 应用领域。

任务实施

步骤 1．了解面向对象的方法

面向对象开发作为一种新兴的软件开发方法，正在逐渐取代传统的方法，日益成为当前软件工程领域的主流方法。

面向对象（Object-Oriented，OO）不仅是一些具体的软件开发技术和策略，而且是一整套关于如何看待软件系统与现实世界的关系、用什么观点来研究问题并进行求解以及如何进行系统构造的软件方法学。

面向对象的设计方法是一种新兴程序方法，其基本思想是使用类、对象、封装、继承、关联、消息等基本概念来系统地进行分析与设计。

步骤 2．UML 的兴起

（1）方法大战

从 20 世纪 80 年代末到 90 年代中期出现了一大批面向对象的分析与设计方法。各种流派的方法相互之间各不相同，它们之间的差异为面向对象的程序开发方法的发展与应用带来了不便。

其中最流行的面向对象的方法是：OMT 方法[由 James Rumbaugh（见图 1-1）提出]、Booch 方法[由 Grady Booch（见图 1-2）提出]和 OOSE 方法[由 Ivar Jacoboson（见图 1-3）提出]，每个方法都有自己的价值和重点。分析是 OMT 方法的强项，设计是 OMT 方法的弱项；设计是 Booch 方法的强项，分析是 Booch 方法的弱项；OOSE 擅长的是行为分析，而在其他方面比较弱。

图 1-1　James Rumbaugh

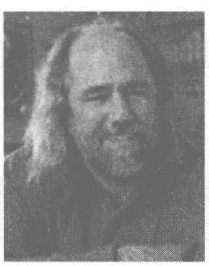


图 1-2　Grady Booch

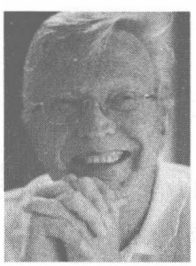


图 1-3　Ivar Jacobson

在 20 世纪 90 年代中期，Grady Booch、Ivar Jacobson、James Rumbaugh 开始借鉴彼此的方法，Booch 采用了 James Rumbaugh 和 Ivar Jacobson 所提出的许多很好的分析技术，James Rumbaugh 的 OMT-2 采用了 Booch 所提出的很好的设计方法，但是，不同符号体系的使用给软件行业带来了混乱。因为不同的符号对于不同的人可能意义不同，而同一事物可能用不同的符号表示，因此引起了很多混乱，人们用“方法大战”形象地描述了这种混乱的局面。

（2）UML 的发展

由于 Booch 方法和 OMT 方法都已经独自成功地发展成为世界上主要的面向对象方法，因此 Grady Booch 和 James Rumbaugh 于 1994 年 10 月共同合作把他们的工作统一起来。1995 年成为“统一方法（Unified Method）”0.8 版。之后，Ivar Jacobson 加入，吸取了他的用例（Use Case）思想，于 1996 年成为“统一建模语言（Unified Modeling Language，UML）”0.9 版。1997 年 1 月，UML 1.0 版本被提交给“对象管理组（Object Management Group，OMG）”组织，作为软件建模语言标准化的候选。随后一些重要的软件开发商和系统集成商成为“UML 伙伴（UML Partners）”，其中有 Microsoft、IBM

和 HP。经过应用并吸收了开发商和其他方面的诸多意见，于 1997 年 9 月再次提交给 OMG 组织，同年 11 月 7 日正式被 OMG 采纳作为业界标准。2001 年，UML 1.4 版本被核准推出。2003 年 UML 2.0 版发布，UML 2.0 根据工业界使用 UML 1.x 的经验进行了相应改进，以帮助简化模型的开发。

UML 是一种可视化的建模语言，它能让系统构造者用标准的、易于理解的方式建立起能够表达出他们想象力的系统蓝图，并且提供一种机制，以便于不同的人之间有效地共享与交流设计结果。

统一了的建模语言 UML 是一种定义良好、表达丰富、功能强大且普遍适用的建模语言。它融入了软件工程领域的新思想、新方法和新技术。它不但支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。

统一建模语言 UML 代表了面向对象软件开发技术的发展方向，具有巨大的市场前景和重大的经济价值。

步骤 3．了解 UML 的语义和表示法

作为一种建模语言，UML 的定义包括 UML 语义和 UML 表示法两个部分。

（1）UML 语义

UML 语义给出了基于 UML 的精确的元模型定义。元模型为 UML 的所有元素在语法和语义上提供了简单、一致、通用的定义性说明，使开发者能在语义上取得一致，消除了因人而异的表达方法所造成的影响。此外，UML 还支持对元模型的扩充定义。

（2）UML 表示法

UML 表示法定义了 UML 符号的表示方法，为开发者或开发工具使用这些图形符号和文本方法给系统建模提供了标准。这些图形符号和文字所表达的是应用级的模型，在语义上它是 UML 元模型的实例。

步骤 4．了解 UML 的特点

（1）统一标准

UML 不仅统一了 Booch、OMT、OOSE 和其他面向对象方法的基本要领和符号，还吸取了面向对象技术领域中其他流派的长处，其中也包括非 OO（面向对象）方法的优点。同时，UML 汇集了面向对象领域中很多人的思想，如图 1-4 所示。UML 使用的符号表示考虑了各种方法的图形表示，删除了大量易引起混乱的、多余的和极少使用的符号，也添加了一些新的符号，提供了标准的面向对象的模型元素的定义和表示法。

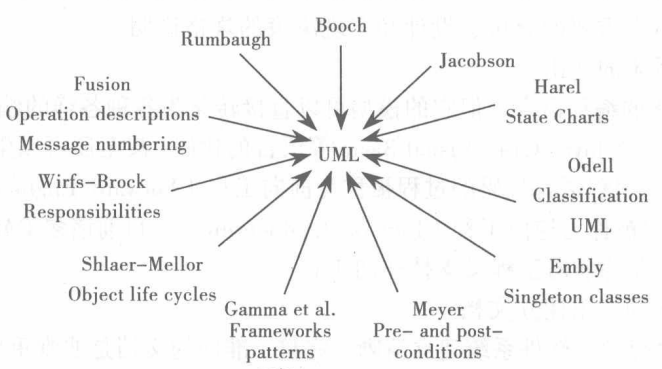


图 1-4　UML 的形成

试读结束：需要全本请在线购买：　www.ertongbook.com

（2）支持面向对象

UML 支持面向对象技术的主要概念，它提供了一批基本的表示模型元素的图形和方法，能简洁明了地表达面向对象的各种概念和模型元素。

（3）可视化表达且表达能力强

UML 是一种图形化语言，用 UML 的模型图能清晰地表示系统的逻辑模型或实现模型。它不只是一堆图形符号，在每一个图形符号后面都有良好定义的语义；UML 还提供了语言的扩展机制，用户可以根据需要增加自己定义的构造型、标记值和约束等，它的强大表达能力使它可以用于各种复杂类型的软件系统的建模。

步骤 5. 了解 UML 的功能

UML 是一种建模语言，该语言具有如下功能：

（1）为软件系统的产出（Artifact）建立可视化模型

对于大多数程序员来说，在脑海中设想一个软件的实现与用代码来实现这个软件是没有距离的，怎么想就怎么用代码来实现它。事实上，有些东西最好直接转换为代码，因为文本是写表达式和算法的最直接的方式。但过去即使在这种情况下，程序员仍要做一些简单的建模工作，例如，在白板或纸上画一些简单的模型，这种做法会产生下列问题：

- 不利于交流。如果公司或项目组使用他们自己的语言来描述模型，这个模型很难为公司或项目组外的人员所理解
- 如果不建立模型，软件系统中的有些东西很难用文本的编程语言来表达清楚。
- 如果程序员在修改代码时，没有将他脑海中的模型记录下来，这个信息可能会永远丢失，不便于软件维护。

而统一建模语言 UML 很好地解决了这些问题，其特点如下：

- UML 符号具有定义良好的语义，不会引起歧义。UML 是一个标准的、被广泛采用的建模语言，因此，用 UML 建模有利于交流。
- UML 是可视化的建模语言，它为系统提供了图形化的可视模型，使系统的结构变得直观，易于理解。
- 用 UML 为软件系统建立模型不但有利于交流，还有利于对软件的维护。

（2）规约软件系统的产出

规约（Specifying）意味着建立的模型是准确的、无歧义的、完整的。UML 定义了在开发软件系统过程中所做的所有重要的分析、设计和实现决策的规格说明。

（3）构造软件系统的产出

UML 不是可视化的编程语言，但它的模型可以直接转化为各种各样的编程语言，也就是说，可以从 UML 的模型生成 Java、C++、Visual Basic 等语言的代码，甚至还可以生成关系数据库的表。

从 UML 模型生成编程语言代码的过程被称为前向工程（Forward Engineering），从代码实现生成 UML 模型的过程被称为逆向工程（Reverse Engineering）。目前诸多 CASE 工具，如 Rational Rose 和 Prosa 等，既支持前向工程又支持逆向工程。

（4）为软件系统的产出建立文档

在软件的开发过程中为软件系统建立清晰、完整、准确的文档是非常重要的。UML 可以为系统的体系结构及其所有细节建立文档。UML 还为描述需求、测试、项目规划活动和软件发布管理活动的建模提供了语言（即建模图形元素和符号）。

步骤 6. 熟悉 UML 的组成

UML 的词汇表包括 3 种构造模型：元素、关系、图。元素是模型中重要的抽象；关系将这些元素连接起来；而图则将元素的集合分组。

（1）元素

UML 中的元素又可分为结构元素、行为元素、分组元素、注释元素 4 种。

- 结构元素：是 UML 模型中的名词。结构元素是模型中主要的静态部分，代表了概念的或物理的元素。在 UML 中，共有 7 种结构元素，分别是类（Class）、接口（Interface）、协作（Collaboration）、用例（Use Case）、活动类（Active Class）、构件（Component）和结点（Node）。
- 行为元素：是 UML 模型中的动态部分，它们是模型中的动词，代表了跨越时间和空间的行为。在 UML 中，有两种主要的行为元素，即交互作用（Interaction）和状态机（State Machine）。
  - ➢ 交互作用是由在特定上下文中为完成特定目的而在对象间交换的消息集组成的行为。交互作用包括许多其他元素，如消息、动作顺序（由消息激活的行为）、链（对象间的连接）等。
  - ➢ 状态机是这样一种行为，这种行为规定了对象在其生命周期为响应事件而经历的状态顺序，以及对事件的响应。状态机也包括许多其他元素，如状态、跃迁、事件和活动。
- 分组元素：是 UML 模型中用来组织元素的元素。在 UML 中，有一种主要的分组元素——包（Package）。
- 注释元素：是 UML 模型中解释性的部分。它们是可以用于描述、例解、注解模型中任何元素的注释。在 UML 中，有一种主要的注释元素——注解（Note）。

（2）关系

在 UML 模型中，主要有 4 种关系：

- 依赖（Dependency）关系。
- 关联（Association）关系。
- 泛化（Generalization）关系。
- 实现（Realization）关系。

（3）图

统一建模语言 UML 的图可以分为下列 5 类（共 9 种图形）：

- 用例图（Use Case Diagram）：用例图从用户角度描述系统功能，并指出各功能的操作者。
- 静态图（Static Diagram）：静态图包括类图（Class Diagrams）和对象图（Object Diagrams）。
  - ➢ 类图描述系统中类的静态结构。类图不但定义了系统中的类，表示了类之间的联系（如关联、依赖、聚合等），还描述了类的内部结构（类的属性和操作）。
  - ➢ 对象图是类图的实例，使用与类图类似的标识。它们的不同点在于对象图显示类的多个对象实例，而不是实际的类。一个对象图是类图的一个实例。
- 行为图（Behavior Diagram）：行为图描述了系统的动态模型和系统对象间的交互关系。行为图包括状态图（Statechart Diagram）和活动图（Activity Diagram）。
- 状态图描述了类的对象所有可能的状态以及事件发生时状态的跃迁条件。
- 活动图描述了满足用例要求所要进行的活动以及活动间的约束关系，活动图有利于识别并发活动。
- 交互图（Interactive Diagram）：交互图描述了对象间的交互关系。交互图包括顺序图（Sequence Diagrams）和协作图（Collaboration Diagram）。

- 顺序图描述了对象之间的动态合作关系，它强调对象之间消息发送的时间顺序，同时显示对象之间的交互。
- 协作图描述了对象间的协作关系，协作图与顺序图相似，描述了对象间的动态协作关系。除显示信息交换外，协作图还显示对象以及对象之间的关系。

- 实现图（Implementation Diagrams）：实现图包括构件图（Component Diagrams）和部署图（Deployment Diagrams）。
  - 构件图描述代码构件的物理结构及各构件之间的依赖关系。
  - 部署图定义系统中软、硬件的物理结构。部署图不但描述了实际的计算机和设备（用结点表示）以及它们之间的连接关系，还描述了连接的类型及构件之间的依赖性。

从应用的角度看，当采用面向对象技术设计系统时，第一步是描述需求；第二步是根据需求建立系统的静态模型，以构造系统的结构；第三步是描述系统的行为。其中在第一步与第二步中所建立的模型都是静态的，包括用例图、类图、对象图、构件图和部署图 5 个模型图，是统一建模语言 UML 的静态建模机制。而第三步中所建立的模型或者可以执行，或者表示执行时的顺序状态或交互关系，它包括状态图、活动图、顺序图和协作图 4 个模型图，是统一建模语言 UML 的动态建模机制。因此，统一建模语言 UML 的主要内容也可以归纳为静态建模机制和动态建模机制两大类。

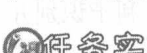
步骤 7. 熟悉 UML 的应用领域

UML 可用于任何面向对象系统的开发建模，不仅可以使用 UML 进行软件建模，同样可以使用 UML 对其他非计算机领域系统进行建模，UML 经常应用在以下领域：

- 信息系统（Information System）：向用户提供信息的存储、检索和提交，处理存放在关系或对象数据库中大量具有复杂关系的数据。
- 技术系统（Technical System）：处理和控制技术设备，如电信设备、军事系统或工业过程。
- 嵌入式系统（Embedded System）：它以软件的形式嵌入到硬件设备中从而控制硬件设备的运行，通常为手机、家电或汽车等设备上的系统。
- 分布式系统（Distributed System）：分布在一组机器上运行的系统，数据很容易从一个机器传送到另一个机器上，需要同步通信机制来确保数据的完整性，通常建立在对象机制上，如 CORBA、COM/DCOM 或 Java Beans/RMI 上。
- 商业系统（Business System）：描述目标、资源、规则和商业中的实际工作。

1.1.3 任务 2：Rational 统一过程

一个项目的成功需要 3 方面的支持：符号、过程和工具。为了更好地了解 UML 及 UML 在整个项目开发过程中的作用，需要了解 RUP 过程和过程的描述方法。



步骤 1. RUP 的简述

RUP（Rational Unified Process，Rational 统一软件开发过程）是一个面向对象且基于网络的程

序开发方法论。它提供了在开发机构中分派任务和责任的方法，它的目标是在可预见和预算前提下确保最终用户需求的高质量软件的产生。

RUP是由Rational软件公司开发和维护的过程产品。根据Rational（Rational Rose和统一建模语言的开发者）的说法，好像一个在线的指导者，它可以为所有方面和层次的程序开发提供指导方针、模板以及事例支持。RUP和类似的产品，例如，面向对象的软件过程（OOSP）以及OPEN Process都是理解性的软件工程工具——把开发中面向过程的方面（例如，定义的阶段、技术和实践）和其他开发的组件（例如，文档、模型、手册以及代码等）整合在一个统一的框架内。

RUP吸收了许多已经在商业上得到证明的软件开发的最佳实践经验，它适用于范围广泛的项目和组织。RUP为每个项目组成员提供了必要的准则、模板和工具指导，使整个开发团队可以充分利用这些最佳工程实践经验，这些最佳工程实践经验为开发队伍提供了很优秀的模板。RUP的实践经验如下：

（1）迭代式开发

在软件开发的早期阶段就想完全、准确地捕获用户的需求几乎是不可能的。实际上，我们经常遇到的问题是需求在整个软件开发工程中经常改变。迭代式开发允许在每次迭代过程中需求可能有变化，通过不断细化来加深对问题的理解。迭代式开发不仅可以降低项目风险，而且每个迭代过程以可执行版本结束，可以鼓舞开发人员。

（2）管理需求

确定系统的需求是一个连续的过程，开发人员在开发系统之前不可能完全详细地说明一个系统的真正需求。RUP描述了如何提取、组织系统的功能和约束条件并将其文档化，用例和脚本的使用已被证明是捕获功能性需求的有效方法。

（3）基于组件的体系结构

组件使重用成为可能，系统可以由组件组成。基于独立的、可替换的、模块化组件的体系结构有助于软件管理复杂性，提高重用率。RUP描述了如何设计一个有弹性的、能适应变化的、易于理解的、有助于重用的软件体系结构。

（4）可视化建模

RUP往往和UML联系在一起，对软件系统建立可视化模型，并帮助人们提供管理软件复杂度的能力。即RUP告诉我们如何可视化地对软件系统建模，获取有关体系结构关于组件的结构和行为信息。

（5）验证软件质量

在RUP中软件质量评估不再是事后进行或单独小组进行的分离活动，而是内建于过程中的所有活动，这样可以及早发现软件中的缺陷。

（6）控制软件变更

迭代式开发中如果没有严格的控制和协调，整个软件开发过程很快就会陷入混乱之中，RUP描述了如何控制、跟踪、监控、修改以确保成功的迭代开发。RUP通过软件开发过程中的制品，隔离来自其他工作空间的变更，以此为每个开发人员建立安全的工作空间。

步骤2．了解RUP的特点

RUP软件开发生命周期是一个二维的软件开发模型（即RUP过程可以用二维结构来描述）。横轴通过时间组织，是过程展开的生命周期特征，体现开发过程的动态结构，用来描述它的术语

主要包括周期（Cycle）、阶段（Phase）、迭代（Iteration）和里程碑（Milestone）；纵轴以内容来组织自然的逻辑活动，体现开发过程的静态结构，用来描述它的术语主要包括活动（Activity）、产物（Artifact）、工作者（Worker）和工作流（Workflow），如图 1-5 所示。

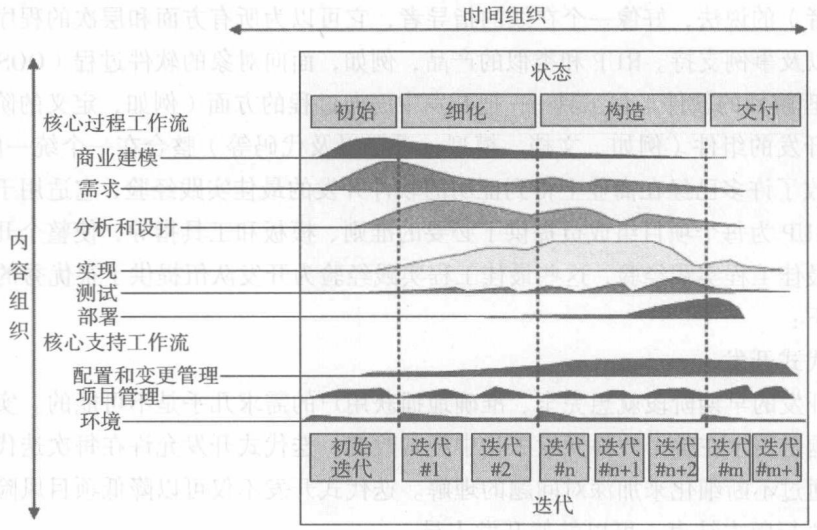


图 1-5 Rational 统一过程

步骤 3. 了解开发过程的时间轴

软件开发过程是使软件从概念到成品所遵循的一系列阶段，RUP 作为一种软件开发过程包含以下 4 个阶段：初始阶段（Inception）、细化阶段（Elaboration）、构造阶段（Construction）和交付阶段（Transition）。每个阶段结束于一个主要的里程碑（Major Milestone）；每个阶段本质上是两个里程碑之间的时间跨度。在每个阶段的结尾执行一次评估以确定这个阶段的目标是否已经达到。如果评估结果令人满意，可以允许项目进入下一个阶段。

（1）初始阶段

初始阶段的主要任务是建立软件系统的商业模型，需要考虑项目的效益，同时进行初步的需求分析。这一阶段需要与系统的用户或领域专家进行讨论。

初始阶段的目标是为系统建立商业案例并确定项目的边界。为了达到该目的，必须识别所有与系统交互的外部实体，在较高层次上定义交互的特性。本阶段具有非常重要的意义，在这个阶段中所关注的是整个项目进行中的业务和需求方面的主要风险。对于建立在原有系统基础上的开发项目来讲，初始阶段可能很短。初始阶段结束时是第一个重要的里程碑：生命周期目标里程碑。生命周期目标里程碑评价项目基本的生存能力。

（2）细化阶段

细化阶段的目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。为了达到该目的，必须在理解整个系统的基础上，对体系结构作出决策，包括其范围、主要功能和诸如性能等非功能需求。同时为项目建立支持环境，包括创建开发案例、创建模板、准则并准备工具。细化阶段结束时是第二个重要的里程碑：生命周期结构里程碑。生命周期结构里程碑为系统的结构建立了管理基准并使项目小组能够在构建阶段中进行衡量。此刻，要检验详细的系统目标和范围、结构的选择以及主要风险的解决方案。

（3）构造阶段

在构造阶段，所有剩余的构件和应用程序功能被开发并集成为产品，所有的功能被详细测试。从某种意义上说，构造阶段是一个制造过程，其重点放在管理资源及控制运作以优化成本、进度和质量。构造阶段结束时是第三个重要的里程碑：初始功能里程碑。初始功能里程碑决定了产品是否可以在测试环境中进行部署。此时，要确定软件、环境、用户是否可以开始系统的运作。

（4）交付阶段

交付阶段的重点是确保软件对最终用户是可用的。交付阶段可以跨越几次迭代，包括为发布做准备的产品测试、基于用户反馈的少量调整。在生命周期的这一点上，用户反馈主要集中在产品调整、设置、安装和可用性问题，所有主要的结构问题应该已经在项目生命周期的早期阶段被解决。在交付阶段的终点是第四个里程碑：产品发布里程碑。此时，要确定目标是否实现，是否应该开始另一个开发周期。在一些情况下这个里程碑可能与下一个周期的初始阶段的结束重合。

步骤4．了解迭代开发

在RUP中，迭代被定义为：迭代包括产生产品发布（稳定、可执行的产品版本）的全部开发活动和使用该发布产品所必需的所有其他外围元素。

RUP过程也是一个迭代递增的开发过程。使用迭代递增式的开发方式，不是在项目结束时一次性提交软件，而是分块逐次地开发和提交。每次迭代选择一些功能点，完成这些功能点，然后再选择别的功能点，如此循环迭代。实际上，涉及实际建模工作的过程存在于上述的每次迭代中。迭代式开发是项目成功的重要保证。RUP的迭代模型如图1-6所示。

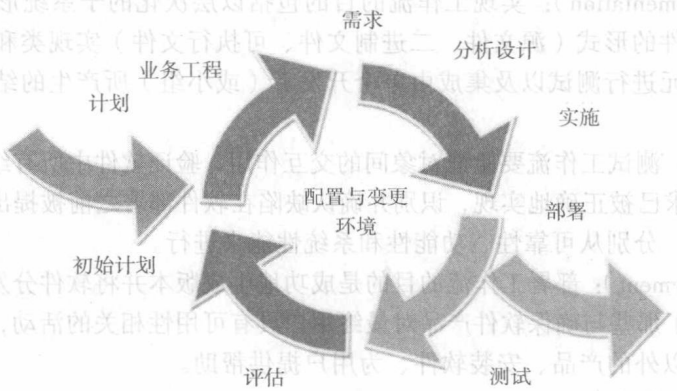


图1-6　RUP的迭代模型

RUP的每个阶段都可以进一步分解成为多次迭代。迭代是一次完整的开发循环，迭代产生可执行的产品版本，可执行产品是最终产品的一个子集，通过一次次迭代递增式地得到最终的系统。

与传统的瀑布式过程相比，迭代过程具有如下优点：

- 很早就开始降低风险。
- 容易控制变更。
- 可重用性更高。
- 更好的总体质量。
- 开发团队可以在开发中学习。

试读结束：需要全本请在线购买：　www.ertongbook.com