

中图分类号：TP311

论文编号：10006SY0308248

北京航空航天大学
硕士学位论文

基于 RUP 的中小型信息系统敏捷开发过程研究

作者姓名	韩小汀
学科专业	管理科学与工程
指导教师	王强 副教授
培养院系	经济管理学院

**The Research of Agile Software Development Process
Applied in Small-Sized Information System
Based on Rational Unified Process**

A Dissertation Submitted for the Degree of Master

Candidate: Han Xiaoting

Supervisor: Pro. Wang Qiang

School of Economics & Management

Beihang University, Beijing, China

中图分类号: TP311

论文编号: 10006SY0308248

硕 士 学 位 论 文

基于 RUP 的中小型信息系统 敏捷开发过程研究

作者姓名 韩小汀

申请学位级别 管理学硕士

指导教师姓名 王强

职 称 副教授

学科专业 管理科学与工程

研究方向 信息管理与信息系统

学习时间自 2003 年 9 月 15 日 起至 2006 年 3 月 10 日止

论文提交日期 2006 年 2 月 27 日 论文答辩日期 2006 年 3 月 7 日

学位授予单位 北京航空航天大学 学位授予日期 年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期： 年 月 日

指导教师签名：_____ 日期： 年 月 日

摘 要

经过多年的努力，中国软件业经历了一个从无到有、从弱到强的发展历程，并且逐步向产业化、规模化发展。而开发中小型信息系统是许多 IT 公司或者团队经常面临的任务，这样的系统具有开发团队规模小、人员配备不齐全、开发周期短、投资少、业务规范化程度低、用户需求变更迅速、项目本身风险小等特点，如果按照严格规范的大型信息系统开发方法的过程进行开发和管理，多人力、高成本、长周期、多文档、严规范等，在中小型信息系统开发中反而成为制约系统开发成功的一些重要影响因素。

RUP 是一套由 Rational 公司设计的软件开发模型。它吸收了多种开发模型的优点，具有很好的可操作性和实用性，被众多的开发组织认同和应用。但是由于 RUP 的内容比较复杂，各部分联系紧密，不易剪裁，所以很少有 RUP 在中小型信息系统开发中应用成功的案例。因此本文提出了结合敏捷开发方法，对 RUP 进行剪裁和改进，研究基于 RUP 的适合中小型信息系统开发的过程框架。

本文首先研究了 Rational 统一过程和敏捷开发，对比它们各自的优缺点。接着，根据中小型信息系统开发的特点，提出了将 RUP 和敏捷开发的思想及实践结合在一起，以适应中小型信息系统开发的思想。然后，本文基于敏捷开发方法对 RUP 进行剪裁和改进，提出了适合中小型信息系统开发的过程框架 ARUP，本过程框架中定义了 4 个核心过程工作流程：建模、实施、测试和部署，3 个核心支持工作流程：项目管理、配置与变更管理和环境，并且定义了 4 个迭代的开发阶段：先启、精化、构建、产品化。这个过程可以帮助开发团队更快的开发更高质量的软件。采用 ARUP 开发过程，成功开发了海峡两岸渔业合作应用系统，以此验证了 ARUP 过程框架的有效性。目前，该系统已经投入运行。

关键词：Rational 统一过程，敏捷软件开发，中小型信息系统

Abstract

The Chinese software industry got the substantial development through years of development. But the small-sized information system often bothers many IT corporations or teams because of their own features such as small team and few people, short time, little money and changeable requirements. It won't be suitable if we use the traditional process which has strict specifications to develop the small-sized information system.

Rational Unified Process (brief name RUP) is software engineering model development by Rational Corporation. RUP has been received by many development teams because it is easy to handle and practical. But because RUP is contacting in the trivial and each process in contents is too close to tailor; there are few examples of applying RUP to the small-sized information system. So the thesis suggest that people can using the thoughts of Agile Software Development to tailor and improve RUP and studies a process frame which can be applied in small-sized information system based on RUP.

Firstly the thesis studies Rational Unified Process and Agile Software Development, and focuses on their advantages and disadvantages. Then the thesis points out that people can combine RUP and the agile methods in order to adapt to the features of small-sized information system. Using the thoughts and experiences of Agile Software Development to tailor and improve RUP, a process named ARUP which fits the development of small-sized information system is given. ARUP defines seven core workflows which are model, implementation, test, deployment, project management, configuration management and environment, and also defines four phases which are inception, elaboration, construction and transition. The ARUP process can help teams develop software of high quality in shorter time. Using ARUP, we succeed to develop a real information system in order to prove the availability of the process. The system has been in use for a long time till now.

Key words: Rational Unified Process, Agile Software Development, Small-Sized Information System

目 录

第一章 绪论	1
1.1 论文的背景	1
1.2 研究现状	2
1.3 主要研究内容和论文结构安排	3
第二章 中小型信息系统开发的特点、问题及对策	5
2.1 中国软件行业发展现状	5
2.2 中小型信息系统的特点	5
2.3 中小型信息系统开发过程中存在的问题及对策	6
2.3.1 在项目开始前不能进行有效的估算	6
2.3.2 没有良好的风险管理	6
2.3.3 项目没有确实可行的项目计划	7
2.3.4 对可能发生变更的项没有有效的配置和变更管理	7
2.3.5 人力资源不足	7
2.3.6 整个开发过程没有有效的质量保证	8
2.3.7 忽略单元测试	8
第三章 RUP 开发过程	10
3.1 RUP 概述	10
3.2 RUP 最佳过程实践	11
3.2.1 迭代地开发软件	11
3.2.2 管理需求	12
3.2.3 基于构件的体系结构	12
3.2.4 可视化软件建模	12
3.2.5 不断的验证软件质量	12
3.2.6 控制软件变更	12
3.3 RUP 开发模型	13
3.3.1 静态结构：工作流程	14
3.3.2 动态结构：迭代式开发	16
3.4 RUP 的特点	18
3.4.1 用例驱动的过程	18
3.4.2 以架构为中心的过程	18
3.4.3 迭代增量开发	19
3.5 RUP 的优点及其存在的问题	20

3.5.1 RUP 的优点.....	20
3.5.2 RUP 本身存在的问题和缺陷.....	20
第四章 敏捷软件开发	23
4.1 敏捷软件开发的发展与现状	23
4.2 敏捷软件开发的基本原理	24
4.2.1 代码与文档.....	24
4.2.2 变化与计划.....	24
4.2.3 软件交付.....	25
4.3 敏捷软件开发的典型代表	25
4.3.1 极限编程.....	25
4.3.2 SCRUM.....	26
4.3.3 水晶系列方法.....	26
4.3.4 适应性软件开发方法.....	27
4.4 敏捷软件开发的主要特征	27
4.4.1 以人为本，重视交流和沟通.....	27
4.4.2 软件过程本身的可变化性.....	28
4.4.3 是方法论而不是具体的过程指导.....	28
4.4.4 形式上的轻量型和简洁性.....	28
4.5 敏捷软件开发的优点和局限性	29
4.5.1 敏捷软件开发的优点.....	29
4.5.2 敏捷软件开发的局限性.....	31
4.5.3 客观的看待敏捷开发过程.....	33
4.6 中小型信息系统开发应用敏捷软件开发面临的问题.....	34
第五章 基于 RUP 的中小型信息系统敏捷开发过程	35
5.1 如何在中小型信息系统开发中使用 RUP	35
5.1.1 RUP 如何与中小型信息系统的开发相结合	35
5.1.2 RUP 过程剪裁方法.....	36
5.2 基于 RUP 的中小型信息系统敏捷开发过程	40
5.2.1 基于 RUP 的中小型信息系统敏捷开发过程框架.....	40
5.2.2 核心工作流程.....	41
5.2.3 过程的四个阶段及其里程碑.....	53
5.3 ARUP 过程小结.....	58
第六章 案例研究	59
6.1 案例开发背景和目的	59

6.2 项目特征分析	60
6.3 第一次迭代开发	61
6.3.1 先启阶段.....	61
6.3.2 精化阶段.....	62
6.3.3 构建阶段.....	65
6.3.4 产品化阶段.....	69
6.4 第二次迭代开发	69
6.5 开发实例总结	69
结论	73
1 研究的成果	73
2 进一步研究的内容	74
参考文献	75
附录	77
附录 A 部分程序代码	77
附录 B 部分单元测试代码	81
附录 C 系统展示	82
攻读硕士期间取得的学术成果	84
致 谢	85

第一章 绪论

1.1 论文的背景

信息系统的开发至今已有 40 多年的历史，计算机把人们从繁重的信息处理（数据的收集、存储、加工等）中解放出来，大大地提高了工作效率，但是同时也让人们付出了极大的代价。以信息系统开发为例，很多项目无法继续而被中途取消；相当一部分项目严重超支；更多的项目有质量问题；只有极少数项目能保质保量地按期完成。

软件开发中的问题在中小型信息系统的开发中表现得更为突出。由于中小型信息系统具有团队规模小、开发时间短、成本低、规范化程度低、需求经常变更等特点，所以如何快速高效的开发高质量的信息系统，减小开发风险，对于中小型信息系统来说，是一个迫切需要解决的问题，也是多年来一直被研究的热点问题。

而要解决上述问题，就必须拥有一个合适的开发过程。系统开发过程可以定义为人们用来开发和维护软件以及相关产品（工程计划、设计文档、规章、检测事例及用户手册等）的一组活动、方法、实践及转换。行之有效的系统开发过程可以提高软件开发的生产效率、提高软件质量、降低成本并减少风险。

在系统开发领域，最早为人们所熟知的是结构化生命周期法和原型化方法，随着面向对象技术的发展，市场上涌现出动态软件开发方法（Dynamic Software Development Method, DSDM）、Open 过程、面向对象软件过程（Object-Oriented Software Process, OOSP）等开发过程。1995 年，由世界著名的面向对象技术专家 Grady Booch、Jim Rumbaugh 和 Ivar Jacobson 共同提出的统一建模语言（Unified Modeling Language, UML）得到了业内人士的高度认可，而与之配套的系统开发方法论 Rational 统一过程（Rational Unified Process, RUP）作为一种信息系统开发方法也得到了广泛关注。

近几年来，由于信息技术的不断发展与完善，越来越多的中小型团队参与到了信息系统的开发中，在软件开发领域出现了“敏捷开发”这一概念，以矫正各种开发方法中机械繁琐的过程，并对开发过程进行自主调整等特征，在软件业引起了广泛的研究兴趣。有一些方法声称自己是敏捷的，其中最著名的是 Kent Beck 等人所提出的极限编程（eXtreme Programming, XP）和 Ken Schwaber 等人提出的 SCRUM 方法，它们都试图找到一种轻型的开发过程，来满足当今软件发展的需求。

从上述各种方法和目前软件开发实践中，可以得到以下六点经验：迭代的开发过程，

管理需求，使用基于组件的架构，为软件可视化建模，验证软件质量，控制对软件的变更。Rational 统一过程很好地吸取了这些经验。

RUP 是 Rational 公司开发和维护的一个产品。其实质是一个软件工程化过程，是 Rational 公司为软件开发提供的完整解决方案，并且已经在世界上许多著名的公司中，诸如爱立信、惠普等，得到了很好的应用。

RUP 作为一种迭代式的开发过程，吸收了多种开发模型的长处，具有很多优点：每一次迭代过程都选择风险最大的 Use Case 执行，因此风险在迭代过程中不断地被发现、被消灭，显著降低了项目的风险系数；RUP 与统一建模语言良好集成，具有多种 CASE 工具的支持，在迭代的开发过程、需求管理、基于组件的体系结构、可视化软件建模、验证软件质量以及控制软件变更等方面，都为软件开发团队提供了开发指南、文档模板和开发工具，提高了团队的生产力；易于管理需求的变化，并能够及时修正以前工作中的失误，这对于信息系统的开发具有重要的意义。

本文给出的基于 RUP 的软件开发过程，是一种适合于中小型信息系统的敏捷软件开发过程。它按照 RUP 的精髓思想来把 RUP 定制成一种敏捷的开发过程，吸取 RUP 中设计与文档的特点，又遵守以 XP 为代表的敏捷方法中快速开发、重构等原则，既适合于现在中小型信息系统开发的实际情况，又不必使开发者陷入过分专注于设计而导致开发进度缓慢的困境，最终达到快速、和谐的系统开发，满足用户需求的目的。

1.2 研究现状

现代的软件技术、软件开发工具比 10 年前的先进许多，而且几乎所有的开发人员都学习过分析、设计、编程、测试等知识与技能，可是如今绝大多数软件项目依然面临着质量低下、进度延误、费用超支这些老问题。人们逐渐意识到，由于企业管理软件过程的能力比较弱，常常导致项目处于混乱状态。过程混乱使得新技术、新工具的优势难以体现。经典的软件工程不是不好，而是不够用。

我们把拥有大量工件和复杂控制的软件开发方法称为重型（Heavy Weight）方法，相对的，我们称工件较少的方法为轻型（Light Weight）方法。以 XP 为代表的敏捷软件开发方法就属于轻型方法，它也是目前广为关注的方法之一。它汲取众多轻量级方法的精华，更加强调对变化的适应和对人性的关注。它以代码为中心，并提供了一项承诺：花费最少的过程开销得到最大的生产力。敏捷软件开发中的许多技术值得在恰当的情况下考虑和采用。

但是，在传统的观念中，我们认为重型方法要比轻型安全许多。由于 RUP 所提供的大量不同级别的细节，许多专家认为 RUP 属于重型方法，并不适合小而快速的项目。但业内也有很多专家通过研究认为，由于 RUP 是一种框架结构，可以用不同的方式使用，如“瀑布”式开发方式或者敏捷方式。既可以把 RUP 调节得轻捷灵便，也可以把它弄成繁文缛节。这取决于如何在开发环境中对 RUP 进行裁剪和应用。通过对 RUP 进行裁剪，RUP 完全可以很好的适应小项目的需要。各个组织可根据自身的实际情况，以及项目规模对 RUP 进行裁剪和修改，以制定出合乎需要的软件工程过程。对于如何使 RUP 变得敏捷，近几年来也有许多人提出了很多观点。

著名学者 Michael Hirsch 认为，RUP 可以适应小项目的需要，并指出，在小项目中成功应用 RUP 的一个关键是仔细选择合适的工件子集并保持这些工件非常简明，剔除不需要的形式主义的工件。Michael Hirsch 所指的小项目是开发人员为 3 到 10 个，开发周期为 3 到 10 个月的项目。Michael Hirsch 根据丰富的项目经验提出了一个应用于小项目的裁剪后的 RUP 的活动集和工件集，他将这一活动和工件集合称为小型项目的 RUP 的建议剩余结构^[1]。

Craig Larman 极力主张以敏捷型方式来使用 RUP。在《UML 和模式应用》一书中，他提出了一个过程，就是基于这种“轻型”RUP 的思想。他的观点是：目前如此众多的努力以推进敏捷型方法，只不过是接受能被视为 RUP 的主流 OO 开发方法而已^[2]。

另一种对待 RUP 的策略是 Robert Martin 的 dX 过程。dX 过程是对 RUP 的成功剪裁，它是一个完全符合 RUP 的过程，而又碰巧与 XP 十分相似。dX 是特别适合于那些不得不用 RUP 而又想用 XP 的开发团队。由于 dX 既是 XP 又同时是 RUP，它可作为以敏捷方式运用 RUP 的一个极好的例子^[3]。

目前，在国内外的业界中，有学者指出，单独采用 RUP 或敏捷开发来开发小型项目都有其不充分的地方，有必要将两者结合起来得到一个恰当级别的过程来开发小型项目。但是，他们都没有给出结合两种开发过程而得到的具体的开发过程和方法。

1.3 主要研究内容和论文结构安排

本文给出了一种适合于中小型信息系统开发的软件过程。它是一种经过剪裁的 Rational 统一开发过程，并在过程中，引入了极限编程等若干敏捷开发方法的思想。本文首先总结了中小型信息系统开发团队规模小、人员配备不齐全、系统开发周期短、系统开发投资少、业务规范化程度低、用户需求变更迅速、项目本身风险小等特点，提出

了如果按照严格规范的大型信息系统开发方法的过程进行开发和管理，多人力、高成本、长周期、多文档、严规范等在中小型信息系统开发中反而成为制约系统开发成功的一些重要影响因素。然后，本文着重总结了 RUP 和各种敏捷开发方法的优缺点，和如何结合敏捷开发的思想与实践对 RUP 模型进行剪裁和改造使之适合中小型信息系统的开发，并得到了一个适合中小型信息系统开发的过程框架 ARUP。然后，本文详细描述了 ARUP 过程中的七个核心工作流程和 4 个迭代的开发阶段。最后，以一个实际的中小型信息系统开发作为应用实例，演示了如何利用这一过程并验证这一过程的有效性。

论文的结构安排如下：

第一章为绪论。介绍了当前软件开发的背景以及开发中小型系统面临的问题，由此给出了一个适用于中小型信息系统的软件开发过程的需求和意义。

第二章为中小型信息系统开发的特点、问题及现状，总结了中小型信息系统的开发特点，开发过程中存在的问题，并针对这些问题提出了相应的解决方案。

第三章为 RUP 开发过程。主要介绍了 RUP 开发过程及其优缺点。

第四章为敏捷软件开发。介绍了敏捷软件开发的发展及现状，研究了 XP、SCRUM、ASD 等适合中小型信息系统的敏捷开发过程，对其进行比较并总结归纳其优缺点。

第五章为基于 RUP 的中小型信息系统敏捷开发过程。结合中小型信息系统的特点，给出了一种基于 RUP 的敏捷软件开发过程框架，详细论述了该过程的工作流程和迭代阶段，以及应用该过程时应注意的问题。

第六章为案例研究。利用本文提出的软件开发过程开发了一个中小型信息系统，主要论述如何利用这个软件开发过程来实现应用系统的开发，并验证这一过程的有效性。

最后是论文总结。对全文进行了总结，并提出进一步研究的方向。

第二章 中小型信息系统开发的特点、问题及对策

2.1 中国软件行业发展现状

经过 20 年的艰难发展，中国软件业走过了一个从无到有、从弱到强、从单个企业到产业集群的发展历程，已达到 150 亿人民币的产业规模，国内登记的软件企业也达到 1 万家左右，软件产品已超过了 1 万种。据软件行业协会统计，2003 年营收超过 1 亿人民币的企业已达到 214 家。但一个事实不容忽视：90% 以上的中国软件企业仅为 50 人以下的规模，绝大多数处于各自为战、手工作坊式的生产状态^[4]。

占中国软件行业绝大部分的还是中小规模的企业，这些中小企业普遍存在规模小、人员流动性大、资本规模和人力规模较小、管理混乱的问题。中小型软件企业必须从天才式的自由开发和作坊式制作中解脱出来，步入团队化、规范化、标准化的新阶段。

中小型企业对于高投入、高风险和长周期的高端软件产品基本没有开发的能力，一般从事的都是较小规模的项目的开发。中小型企业要想走规范化、标准化之路就必须加强针对中小规模的项目的管理。

项目管理和企业的生产率及产品质量有密切的关系。不仅大型项目需要有效的项目管理，中小型项目同样不可缺少。中小型的软件公司只有实行软件项目管理，才能生存和发展，才能向大中型软件公司迈进，促进软件产业的发展。

2.2 中小型信息系统的特点

中国软件业占多数的是中小型的软件开发企业，而他们大多从事的是中小型信息系统的开发。

那什么是中小型信息系统呢？

根据国家标准 GB8567-88，软件的规模可以按汇编代码量分为 4 级：

1. 小规模软件：源程序行数小于 5 000 的软件。
2. 中规模软件：源程序行数为 10 000-50 000 的软件。
3. 大规模软件：源程序行数为 100 000-500 000 的软件。
4. 特大规模软件：源程序行数大于 500 000 的软件。

根据实际情况，中小型信息系统一般都是定制型软件产品，这种软件产品一般都有专门和特定用户群，在开发开始以前已经通过签订合同确定了软件产品要实现的功能和

有明确的交付日期。众所周知，与传统的生产产业相比，软件产品的开发不需要太多的物质资源，主要的资源是大量拥有一定技能的人力资源；生产产品主要是程序代码和技术文件。软件项目的管理主要是对开发过程中的控制，与传统产业的管理有很大的不同。在软件开发项目管理中，中小型项目和大型项目的管理也各有不同。

中小型信息系统的开发一般有以下特点：

1. 项目功能比较简单、单一。
2. 项目人员数比较少。
3. 项目工期比较短。
4. 项目预算比较少。

5. 工期紧，一般是进行定制软件的开发，在项目开始之前就已经和客户确定了产品发布的日期和后期维护等事项，而这一日期往往是在不准确的估算的基础上建立的。

众所周知，软件开发是需要团队合作的工作，好的软件是开发团队集体智慧和精诚合作的产物。人的因素对于软件开发的成败占较大的比例。从管理的角度来考虑，参与开发的人员越多越不利与有效的组织和管理。中小型的开发团队规模比较小，人员结构不复杂。在开发过程中，项目组成员的沟通比较容易，团队成员比较少易于管理，所以中小型开发团队通常具有较高的生产效率。

中小型信息系统开发相对与大型系统来说在管理上应该是更容易的，但是由于中小型项目的特点给它的管理在带来优势的同时也形成了一定的不足。

2.3 中小型信息系统开发过程中存在的问题及对策

2.3.1 在项目开始前不能进行有效的项目估算

项目的估算是根据项目的需求规格说明书或其他需求文档的需求定义，通过对需求进行分解和估算人员掌握的经验数据，导出项目轮廓和项目的估算规模、人力数、和成本及时间进度等控制项目的要素的估算值。从而为制定项目计划和调整项目计划提供依据，是避免开发风险和制定合理的项目计划的有效保证。但是一些中小型项目的开发组织在开发前很少进行认真的估算^[5]。这就需要项目经理意识到项目估算的重要性，在开发之前对项目进行认真有效的估算。

2.3.2 没有良好的风险管理

风险管理是项目开发过程中的重要活动，其目标是对项目过程中，可能对项目产生

重大影响的各种因素进行科学的预见，并加以动态管理。从而可以减少或杜绝这些风险因素给项目造成的损失。项目风险类型有很多种，大多数的中小项目的项目经理所认识的风险大多局限在商业风险（销售问题等）中，对风险的理解很片面。有的时候虽然预见到风险可是由于客观条件的制约也不能很好的制定解决和防范风险的策略。

而 RUP 提供了一套很好的针对风险管理的策略，它提出，不论是在项目开始之前，还是在每个迭代开始之前，都要对项目的风险进行重新的估算。

2.3.3 项目没有确实可行的项目计划

软件工程知识发展到今天，很多开发团队都已经认识到了它的重要性。大多都按照开发需要的步骤有序的按照开发流程进行开发。但是却由于各种原因忽视或不能有效的进行项目管理。往往由于项目较小又没有进行合理的项目估算，便很草率地制定一个开发日程表，没有认真地估计项目难度，结果实际完成时间与估计完成时间往往有较大差别。这种情况在中小型项目的开发活动中非常普遍。有的制定了比较完善的和合理的开发计划，但是变更在软件开发过程中是不可避免的，需求、设计的种种不可控制的变更发生时不能很好的调整项目计划也同样造成了计划不可靠性，不能真正的在整个开发过程进行有效的项目管理。由于不能有效的进行项目管理进而不能保证各个周期的按时完成，导致工期的延误或为赶工期不合理的加大工作量使整个开发团队疲惫不堪。这就需要用一个自适应的项目过程去对项目进行管理，也就是说，先对项目制定一个粗粒度的开发计划，然后根据开发的实际情况，及时的调整已经制定的开发计划。

2.3.4 对可能发生变更的项没有有效的配置和变更管理

在软件开发的过程中变更在所难免，而且可能在各个阶段发生。变更并不可怕，但是要保证每次变更都是被记录和可在可控的情况下进行的。每次变更都会造成相应的工作量的消耗。不可控的和随意的变更往往会造成工作流程之间的多次往复，极大的浪费了工作量，同时对正常的开发活动造成了延误。由于一些变更得不到有效控制甚至造成后续工作的变更不及时给系统造成了错误。可以通过有效的变更管理来减少发生变更所带来的损失，在这一点上，RUP 也提供了很好的示范。

2.3.5 人力资源不足

在软件开发不需要使用大量的物质资源，人力资源是软件开发中最重要的资源。人力资源不足又成为困扰各个开发组织特别是进行中小型项目开发的开发组织的问题。

1. 高水平开发人员的缺少。缺少有开发经验丰富、技术水平比较高的人员。高水

平人员的缺少造成在开发前期不能或很难对项目进行有效的风险估算和项目时间、成本、工作量等问题的估算。

2. 开发人员少。开发人员少，意味着不同人员的程序之间交互、接口相对少一些。通常是同样的几个人从头到尾负责一个项目。往往是几个人简单讨论一下最基本的数据结构、函数接口便分头去做自己的工作了，缺少比较正式的文档。这都为项目的其它阶段的开发留下了很大的隐患。并且中小型的开发组织人员流动性比较大，如果没有确定正式的文档可能会造成一旦有人中途退出开发组织，其他人加入时，新来的人难以理解前人做好的工作，只能自己从头来，极大的浪费人力资源，甚至延误工期^[6]。

3. 领域专家的缺失。在一些特殊行业的应用软件的开发过程中，行业的业务规则可能很烦琐、生僻、专业化强，使开发人员对用户的需求难于理解，很难开发出充分满足客户需要的产品。这时往往需要开发组织中有熟悉和了解相应行业的领域专家加入开发小组。但是由于人力成本的问题，在中小型的开发组织中，往往都很难配备该角色。

人力资源不足基本上是每个中小型信息系统开发过程中都会遇到的难题，在无法增加人员的基础上，就需要提高现有开发人员的勇气和能力，而以 XP 为代表的敏捷开发，强调“以人为本”去进行软件开发，在这一点上做出了很好的示范。

2.3.6 整个开发过程没有有效的质量保证

大多数小型软件公司没有专门的质量保证人员和质量保证规程，只是把软件质量保证简单理解为测试的过程，这是远远不够的。与传统工业一样，软件产品的质量也是开发组织不可忽视的问题。软件质量保证的目的是以独立审查的方式监控软件生产任务的执行。质量保证监督的不但是开发产品，更重要的是保证整个开发过程的质量。质量保证的重点在于项目开发过程的质量，通过对开发过程的质量监督达到提高产品质量的目的。

2.3.7 忽略单元测试

造成这一现象的原因是每个模块相对比较简单，但是为了测试一个模块需要建立一些测试环境，编写测试用例。为了避免麻烦，这一过程有时就被省略了。

殊不知，一旦直接进入系统测试，发现运行结果不正确后需要一步步查找。由于模块间的调用关系，可能查了很久才发现是某个模块的问题。这种方法一来效率比较低，大量的时间用在了将一个错误定位在模块上了。另外由于这种测试不完全，真正运行系统，当调用某模块时，可能大部分时候都是正常数据，极少出现边界情况，可能某些边

界情况容易被忽视，很久之后才被发现。在进行中小型信息系统开发的时候，可以借鉴“测试优先”的思想。也就是说，在编写代码之前，先编写应有的单元测试代码，生成测试用例，这样很容易消除一些隐患。问题被发现的越早对开发过程的影响就越小，发现的越晚，消除该问题消耗的工作量开销就越大。

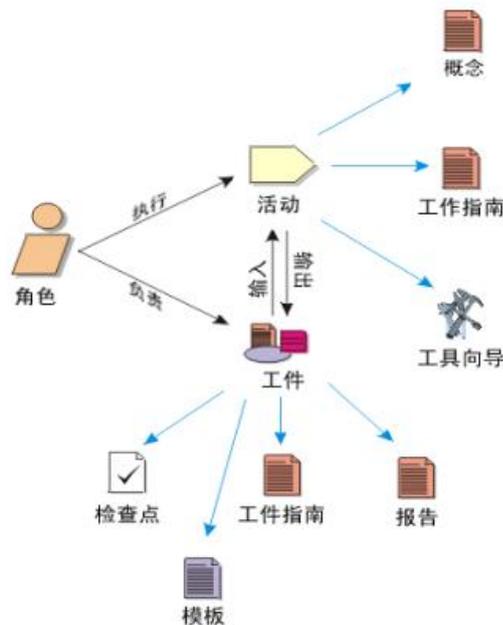
第三章 RUP 开发过程

3.1 RUP 概述

RUP 是由 Rational 软件公司开发的一种软件工程过程，主要由 Ivar Jacobson 的 The Objectory Approach 和 The Rational Approach 发展而来。它是文档化的软件工程产品，所有 RUP 的实施细节及方法导引均以 Web 文档的方式集成，由 Rational 公司开发、维护并销售，为各种软件开发组织提供了一种有效的分配与管理任务和职责的规范方法，保障开发组织能够在预定的进度和预示范围内开发出满足最终用户需要的高质量软件产品。RUP 又是一套软件工程方法的框架，各个组织可根据自身的实际情况，以及项目规模对 RUP 进行裁剪和修改，以制定出合乎需要的软件工程过程。

RUP 吸收了多种开发模型的优点，具有很好的可操作性和实用性并且和目前应用广泛的统一建模语言有着良好集成、有多种 CASE 工具进行支持、不断的升级与维护，迅速得到软件开发行业的广泛认同，越来越多的软件开发组织以它作为软件开发模型框架。

RUP 中定义了一些核心概念，如图 1。其中，角色（Worker）描述某个人或者一个小组的行为与职责，RUP 预先定义了很多角色。活动（Activity）是一个有明确目的的独立工作单元。工件（Artifact）是活动生成、创建或修改的一段信息。



RUP 可以用二维坐标来描述。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）；纵轴以内容来组织为自然的逻辑活动，体现开发过程的静态结构。如图 2 所示：

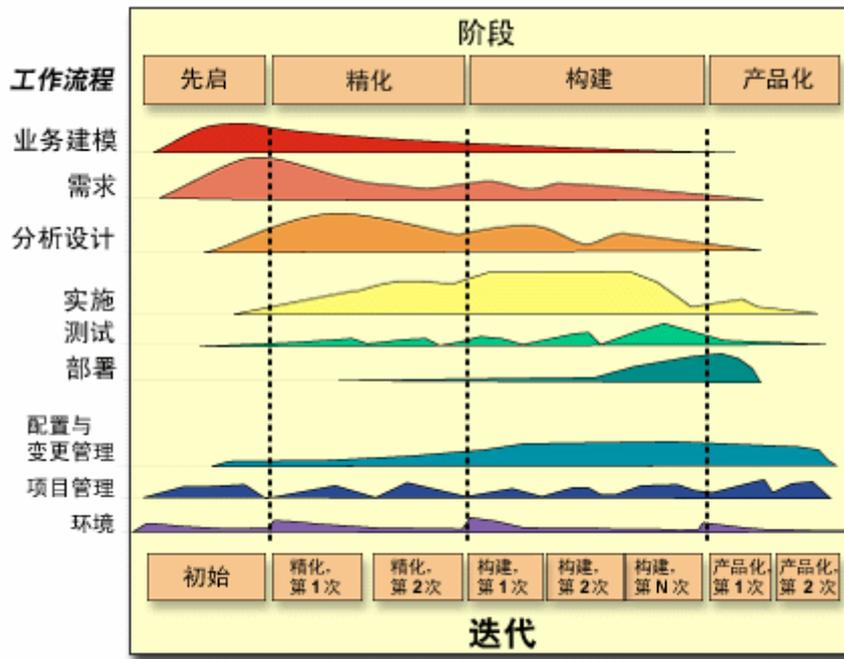


图 2 RUP 的二维开发模型^[8]

3.2 RUP 最佳过程实践

随着计算机应用的普及，软件作为一种应用产品在越来越多的领域、行业得到广泛的使用，软件产业得到了长足和飞速的发展。和其他传统性产业不同的是，软件产品的开发存在着更多的不确定性与不可控性，管理的难度更大。不同的软件开发项目在不同的方面有着不足和缺陷。RUP 针对一些在软件开发的过程中存在的共同的症状，结合大量的项目开发实践经验，分析更深层次的原因总结出了 6 项关键的最佳实践活动，来有效的解决这些制约软件开发过程管理和影响生产产品质量的根本原因。这 6 项最佳过程实践是：

3.2.1 迭代地开发软件

面对当今的复杂的软件系统，使用连续的开发方法，如：首先定义整个问题，设计完整的解决方案，编制软件并最终测试产品，是不可能的。需要一种能够通过一系列细化，若干个渐进的反复过程而生成有效解决方案的迭代方法。RUP 支持专注于处理在生命周期每个阶段中最高风险的迭代开发方法，极大地减少了项目的风险。迭代方法通过

可验证的方法来帮助减少风险——经常性的、可执行版本使最终用户不断的介入和反馈。因为每个迭代过程以可执行版本告终，开发队伍停留在产生结果上，频繁的状态检查帮助确保项目能按时进行。迭代化方法同样使得需求、特色、日程上战略性的变化更为容易^[9]。

3.2.2 管理需求

RUP 描述了如何提取、组织和文档化需要的功能和限制；跟踪和文档化折衷方案和决策；捕获和进行商业需求交流。过程中用例和场景的使用被证明是捕获功能性需求的卓越方法，并确保由它们来驱动设计，实现和软件的测试，使最终系统更能满足最终用户的需要。它们给开发和发布系统提供了连续的和可跟踪的线索。

3.2.3 基于构件的体系结构

RUP 在全力以赴开发之前，关注于早期的开发和健壮可执行体系结构的基线。它描述了如何设计灵活的、可容纳修改的、直观便于理解的，并且促进有效软件重用的弹性结构。RUP 支持基于构件的软件开发。构件是实现清晰功能的模块、子系统。RUP 提供了使用新的及现有构件定义体系结构的系统化方法。它们被组装为良好定义的结构，或是特殊的、底层结构如 Internet、COBRA 和 COM 等的工业级重用构件^[10]。

3.2.4 可视化软件建模

开发过程显示了对软件如何可视化建模，捕获体系结构和构件的构架和行为。这允许隐藏细节和使用“图形构件块”来书写代码。可视化抽象帮助沟通软件的不同方面，观察各元素如何配合在一起，确保构件模块一致于代码，保持设计和实现的一致性，促进明确的沟通。Rational 软件公司创建的工业级标准 UML 是成功可视化软件建模的基础^[11]。

3.2.5 不断的验证软件质量

拙劣的应用程序性能和可靠性是戏剧性展示当今软件可接受性的特点。从而，质量应该基于可靠性、功能性、应用和系统性能，根据需求来进行验证。RUP 帮助计划、设计、实现、执行和评估这些测试类型。质量评估内建于过程、所有的活动，包括全体成员，使用客观的度量 and 标准，并且不是事后型的或单独小组进行的分离活动。

3.2.6 控制软件变更

管理变更的能力——确定每个修改是可接受的、能被跟踪的——在变更不可避免环

境中是必须的。开发过程描述了如何控制、跟踪和监控修改以确保成功的迭代开发。它同时指导如何通过隔离修改和控制整个软件产物（例如：模型、代码、文档等）的修改来为每个开发者建立安全的工作区。另外，它通过描述如何进行自动化集成和建立管理使小队如同单个单元来工作。

Rational 统一过程将这些最佳实践活动以一种适当的形式结合起来，使之适应广泛的项目开发和开发组织。作为一种软件开发过程，它提供了如何在开发组织中严格的分配任务和职责的方法。**RUP** 是一个适用于整个软件开发周期的软件开发过程。比较全面的涵盖了软件开发生命周期中各个阶段的活动。把各个最佳过程实践分析开来，用标示出的核心工作流程的方式表示各个阶段的相应关键活动，针对软件开发各个阶段可能遇到的问题来提出相应的参考解决方案，达到提高开发效率和产品质量的目的。

3.3 RUP 开发模型

RUP 与传统的开发过程模型最大的不同就是提出了二维开发模型。传统的瀑布式开发模型是一个单维的模型，开发工作划分为多个连续的阶段。在一个时间段内，只能作某一个阶段的工作比如，分析、设计或者实现。**RUP** 提出了以二维开发模型。生命周期根据时间和 **RUP** 的核心工作流程划分为二维空间。在所有工作流程都可以在不同阶段进行，只是工作量的大小存在差异。在一定的条件下，**RUP** 模型也可以退化为传统的开发模型。适应可能存在的各种开发环境的需要。

RUP 用时间来从开发组织对项目周期进行管理的角度来描述整个软件开发生命周期，可以进一步描述为周期、阶段，迭代，是 **RUP** 的动态组成部分。周期可以进一步划分为先启阶段（**Inception**）、精化阶段（**Elaboration**）、构建阶段（**Construction**）和产品化阶段（**Transition**）。每个阶段结束于一个主要的里程碑（**Major Milestones**）：每个阶段本质上是两个里程碑之间的时间跨度。在每个阶段的结尾按照事先制定的阶段目标和标准执行一次评估以确定这个阶段的目标是否已经满足。如果评估结果令人满意的话，可以允许项目进入下一个阶段，否则修改或重新执行该阶段直到到达标准为止。阶段可以进一步分解为迭代，一个迭代是一个完整的开发过程的循环，产生一个可执行的产品版本，是最终产品的一个子集，它增量式地发展，从一个迭代过程到另一个迭代过程到成为最终的系统。

RUP 用核心工作流程来从软件开发技术的角度描述整个软件开发生命周期，可以进一步描述为活动（**activities**）、工作流程（**workflow**），产品（**artifact**）、角色（**worker**），

是 RUP 的静态组成部分。RUP 中有 9 个核心工作流程,分为 6 个核心过程工作流程(Core Process Workflows)和 3 个核心支持工作流程(Core Supporting Workflows)。9 个核心工作流程在项目中轮流被使用,在每一次迭代中以不同的重点和强度重复。

3.3.1 静态结构: 工作流程

软件开发过程描述了什么时候,什么人,做什么事,以及怎样实现某一特定的目标。RUP 采用以下四个基本模型元素组织和构造系统开发过程:角色(the who)、活动(the how),工件(the what)、工作流程(the when)。

角色描述某个人或一个小组的活动与职责。一个开发人员可以同时是几个角色,一个角色也可以由多个开发人员共同承担。RUP 预先定义了很多角色,例如:业务流程分析师、架构师、数据库设计员、测试员、项目经理等等,并对每一个角色的工作和职责都作了详尽的说明。

活动是一个有明确目的的独立工作单元。工件是活动生成、创建或修改的一段信息。它是活动的输入同时又是它的输出结果。工件以多种形式存在,例如:模型(Model),源代码、可执行文件、文档等。

模型是从某一个角度对系统的完全描述。RUP 的很大一部分工作就是设计和维护一系列的模型,这其中有 Use Case Model, Business Model, Analysis Model, Design Model 等。所有的这些模型都以 UML 描述,因此它们是标准的并为多种 CASE 工具支持。RUP 并不鼓励写在字面上的文档,工件应尽可能地在 CASE 工具中创建和修改并为版本管理工具跟踪和维护,它们在整个软件开发周期中动态地增加和修改。当然也可以根据需要为模型生成报告(Reports),但它们是静态的,是某一时刻模型的快照不需要维护和修改。

工作流程描述了一个有意义的连续的活动序列,每个工作流程产生一些有价值的工件,并显示了角色之间的关系。RUP 主要提供两种组织工作流程的方式:核心工作流程(Core Workflow)和迭代工作流程(Iteration Workflow)。

核心工作流程从逻辑上把相关角色和活动划分为组,以描述 RUP 的逻辑组成部件。它们相当于模板一样,并不在开发过程中真正的执行。迭代工作流程是 RUP 的一个具体的实现过程,它们对核心工作流程进行裁剪,是核心工作流程的具体实现。每类工作流程都会同一个或多个模型打交道。

RUP 有九个核心的工作流程。以下简单描述这些工作流程的目的:

1. 业务建模 (Business Modeling)

业务建模工作流程描述了如何为新的目标组织开发一个构想，并基于这个构想在商业用例模型和商业对象模型中定义组织的过程，角色和责任。

2. 需求 (Requirements)

需求工作流程的目标是描述系统应该做什么，并使开发人员和用户就这一描述达成共识。为了达到该目标，要对需要的功能和约束进行提取、组织、文档化；最重要的是理解系统所解决问题的定义和范围。

3. 分析和设计 (Analysis & Design)

分析和设计工作流程将需求转化成未来系统的设计，为系统开发一个健壮的结构并调整设计使其与实现环境相匹配，优化其性能。分析设计的结果是一个设计模型和一个可选的分析模型。设计模型是源代码的抽象，由设计类和一些描述组成。设计类被组织成具有良好接口的设计包 (Package) 和设计子系统 (Subsystem)，而描述则体现了类的对象如何协同工作实现用例的功能。

设计活动以体系结构设计为中心，体系结构由若干结构视图来表达，结构视图是整个设计的抽象和简化，该视图中省略了一些细节，使重要的特点体现得更加清晰。体系结构不仅仅是良好设计模型的承载媒介，而且在系统的开发中能提高被创建模型的质量。

4. 实现 (Implementation)

实现工作流程的目的包括以层次化的子系统形式定义代码的组织结构；以组件的形式 (源文件、二进制文件、可执行文件) 实现类和对象；将开发出的组件作为单元进行测试以及集成由单个开发团队 (或小组) 所产生的结果，使其成为可执行的系统。

5. 测试 (Test)

测试工作流程要验证对象间的交互作用，验证软件中所有组件的正确集成，检验所有的需求已被正确的实现，识别并确认缺陷在软件部署之前被提出并处理。RUP 提出了迭代的方法，意味着在整个项目中进行测试，从而尽可能早地发现缺陷，从根本上降低了修改缺陷的成本。测试类似于三维模型，分别从可靠性、功能性和系统性能来进行^[12]。

6. 部署 (Deployment)

部署工作流程的目的是成功的生成版本并将软件分发给最终用户。部署工作流程描述了那些与确保软件工件对最终用户具有可用性相关的活动，包括：软件打包、生成软件本身以外的产品、安装软件、为用户提供帮助。在有些情况下，还可能包括计划和进

行 beta 测试版、移植现有的软件和数据以及正式验收。

7. 配置和变更管理（Configuration & Change Management）

配置和变更管理工作流程描绘了如何在多个成员组成的项目中控制大量的工件。配置和变更管理工作流程提供了准则来管理演化系统中的多个变体，跟踪软件创建过程中的版本。工作流程描述了如何管理并行开发、分布式开发、如何自动化创建工程。同时也阐述了对工件修改原因、时间、人员保持审计记录。

8. 项目管理（Project Management）

软件项目管理平衡各种可能产生冲突的目标，管理风险，克服各种约束并成功移交使用户满意的产品。其目标包括：为项目的管理提供框架，为计划、人员配备、执行和监控项目提供实用的准则，为管理风险提供框架等。

9. 环境（Environment）

环境工作流程的目的是向软件开发组织提供软件开发环境，包括过程和工具。环境工作流程集中于配置项目过程中所需要的活动，同样也支持开发项目规范的活动，提供了逐步的指导手册并介绍了如何在组织中实现过程。

核心工作流程并不是具体的实现，而核心工作流程中的某些活动有可能在软件开发周期中，一遍又一遍地在迭代工作流程中得以细化^[13]。

3.3.2 动态结构：迭代式开发

在时间维上，为了能够方便地管理软件开发过程，监控软件开发状态，RUP 把软件开发周期划分为 Cycles，每个 Cycle 生成一个产品的新的版本。每个 Cycle 都依次由四个连续的阶段组成，每个阶段都应完成确定的任务。

1. 先启阶段

先启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，先启阶段可能很短。

先启阶段结束时是第一个重要的里程碑：生命周期目标（Lifecycle Objective）里程碑。生命周期目标里程碑评价项目基本的生存能力。

2. 精化阶段

精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