

高等学校教材·软件工程

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

高级软件开发过程

—— Rational统一过程、敏捷过程与微软过程

金敏 周翔 编著



清华大学出版社

高等学校教材·软件工程

高级软件开发过程

——Rational 统一过程、敏捷过程与微软过程

金 敏 周 翔 编著

清华大学出版社
北 京

内 容 简 介

本书探讨了 Rational 统一过程、敏捷过程与微软过程三种目前软件界最具影响力的软件开发过程各自的优势与不足,帮助软件开发企业从中进行选择。

本书首先分析了瀑布模型、螺旋模型等传统的软件生命周期模型在解决现代软件产业困境方面存在的局限性,提出走出现代软件产业困境需要如 Rational 统一过程、敏捷过程与微软过程的“软件过程模式”。接着围绕软件过程模式中的生命周期、人员、方法与产品四大要素及相互关系,分别剖析了三种软件过程的原则规范、思想本质、存在的优点和缺陷及适用的软件项目类型。阐述过程注重深入浅出、理论联系实际,既有抽象的思想分析,又有生动的实际案例对原则的内涵及运用方法进行诠释;既有成功的项目实践经验介绍,又有失败的项目教训总结。本书最后在综合吸收三种软件过程的优点、克服其缺点的基础上,提出一种新的集成过程模式,并指出:该模式也不是一种放之四海皆准的软件过程模式,任何软件项目都应根据项目的具体需求和环境对选用的软件过程模式进行适当地裁剪或扩充,寻求最适于该项目的、专用的最佳软件过程。

本书可作为高等院校计算机及相关专业的研究生和本科高年级学生的教材,也可作为软件开发人员、项目经理、企业领导选择和实施 Rational 统一过程、敏捷过程与微软过程的快速指南和简明参考资料。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

高级软件开发过程——Rational 统一过程、敏捷过程与微软过程/金敏,周翔编著. —北京:清华大学出版社,2005.9

(高等学校教材·软件工程)

ISBN 7-302-11543-5

I. 高… II. ①金… ②周… III. 软件开发-高等学校-教材 IV. TP311.52

中国版本图书馆 CIP 数据核字(2005)第 089736 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
<http://www.tup.com.cn> 邮 编:100084
社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:丁 岭

文稿编辑:付弘宇

印 装 者:北京市清华园胶印厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:8.25 字数:192千字

版 次:2005年9月第1版 2005年9月第1次印刷

书 号:ISBN 7-302-11543-5/TP·7554

印 数:1~3000

定 价:12.00元

改革开放以来，特别是党的十五大以来，我国教育事业取得了举世瞩目的辉煌成就，高等教育实现了历史性的跨越，已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上，高等教育规模取得如此快速的发展，创造了世界教育发展史上的奇迹。当前，教育工作既面临着千载难逢的良好机遇，同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾，是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月，教育部下发了《关于加强高等学校本科教学工作，提高教学质量的若干意见》，提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月，教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件，指出“高等学校教学质量和教学改革工程”是教育部正在制订的《2003—2007年教育振兴行动计划》的重要组成部分，精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间（2003—2007年）建设1500门国家级精品课程，利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放，以实现优质教学资源共享，提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代

表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

清华大学出版社经过近 20 年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材经过 20 多年的精雕细刻,形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
E-mail: dingl@tup.tsinghua.edu.cn

软件项目开发曾被喻为“野马”。目前在软件界，最有能力驾驭这匹“野马”的三种软件开发过程是 Rational 统一过程、敏捷过程与微软过程。如果开发人员、项目经理、企业领导希望在自己的软件项目或企业组织中引入软件开发过程，究竟应该选择哪一种？这三种软件开发过程究竟孰优孰劣？本书即探讨了这些问题。

本书主要内容和结构安排

第 1 章为绪论。分析了瀑布模型、螺旋模型等经典的软件生命周期模型在解决现代软件产业困境方面存在的局限性，提出走出现代软件产业困境需要诸如 Rational 统一过程、敏捷过程与微软过程这类“软件过程模式”。

第 2~4 章围绕软件过程模式中的生命周期、人员、方法与产品四大要素及相互关系，在三个章节中分别剖析了 Rational 统一过程、敏捷过程与微软过程三种软件过程的原则规范、思想本质、三者对比存在的优缺点和相互关系，以及适用的软件项目范围。每种过程的阐述深入浅出、理论联系实际，既有抽象的原则思想分析，又有形象生动的实际案例，对原则内涵及运用方法进行诠释；既有成功的项目实践经验介绍，又有失败的教训总结。

第 5 章在综合比较吸收前三章给出的三种软件过程各自的优点、同时克服相应缺点的基础上，提出一种新的集成过程模式，同时指出：集成过程模式也不是一种放之四海皆准的软件过程模式，任何软件项目都应根据项目的具体需求和环境对选用的软件过程模式进行适当地裁剪或扩充，寻求适于该项目的、专用的最佳软件过程。

本书特色

相对其他软件开发过程方面的书籍，本书具有如下特色。

1. 先进性、前沿性和实用性

本书跟踪全球软件界最新的软件过程方法技术和实践经验，分析了三种软件开发过程——Rational 统一过程、敏捷过程与微软过程。这三种软件开发过程是目前在软

件学术界最具影响力，同时也是软件企业界目前最受欢迎、正在使用或正准备使用的三种软件过程。这三种软件开发过程所具有的先进性、前沿性和实用性与它们的创立者密切相关。其中：“Rational 统一过程”由专业化的软件过程产品提供商——Rational 公司提出并维护，该公司聚集了面向对象领域三位杰出的专家 Grady Booch、James Rumbaugh 和 Ivar Jacobson，他们也是面向对象开发的行业标准语言——UML（Unified Modeling Language，统一建模语言）的创立者，整个 Rational 统一过程（RUP）从最初的 Ericsson（爱立信）方法到最终的 RUP 的发展史实质就是一部 Rational 公司对现代软件开发中诸多最佳实践经验三十多年的跟踪捕获史；“敏捷过程”则是 2001 年 2 月由 17 个软件方法学家组织成立的敏捷软件开发联盟提出的软件开发准则，其目的是矫正某些官僚、烦琐的软件过程；“微软过程”则是由世界上最大、也是最成功的软件公司——微软公司根据自身三十多年软件开发的经验总结而成，微软自身的巨大成功说明了该过程的实用性与有效性。

2. 独特性

国内外目前已有一些专门独立论述这三种软件开发过程的书籍资料。然而，对于希望在自己的软件项目或企业组织中引入软件开发过程的开发人员、项目经理、企业领导而言，摆在他们面前的一个更现实的问题是：这三种软件开发过程究竟孰优孰劣？究竟应该选择哪一种？至今尚未有一本书籍将这三种软件开发过程综合在一起进行对比论述，并给出以上问题的答案。

本书填补了这一空白，从软件过程模式角度，纵横捭阖，对比分析了 Rational 统一过程、敏捷过程、微软过程三种软件过程的原则规范、思想本质、存在的优点和缺陷、三种软件过程的相互关系以及各自适用的软件项目范围。

3. 综合性

本书综合吸收了国内外同类书籍的优秀成果，主要参考书籍包括：

- 《统一软件开发过程》^[6] 该书由 Rational 公司面向对象领域的三位大师 Ivar Jacobson, Grady Booch, James Rumbaugh 合著而成。他们也是 UML 的创立者。
- 《敏捷软件开发：原则、模式与实践》^[8] 该书由 Robert C. Martin（美）著，并于 2003 年 6 月荣获美国《软件开发》杂志举办的第 13 届 Jolt 大奖。
- 《微软开发过程与案例》^[12] 该书由微软中国研究院的项目经理和开发人员根据在微软公司的亲身软件开发实践总结而成。该书在浙江大学作为试用教材，获得了极大的成功。

4. 理论的严谨性

本书首先分析了瀑布模型、螺旋模型等传统的软件生命周期模型在解决现代软件产业困境方面存在的局限性，在对最新的软件开发实践经验进行分类整理和加工提炼的基础上，提出一个新的有关软件过程的概念——软件过程模式。相对软件生命周期模型，

软件过程模式更全面、深刻、细致地反映了软件过程中的各个层面和各个环节，是对软件生命周期模型的补充和发展。

接下来，本书从软件过程模式的四要素及相互关系的角度分析三种典型的软件过程模式——Rational 统一过程、敏捷过程与微软过程，使读者能条分缕析、纲举目张地迅速而准确地把握这些软件过程的思想本质、原则规范以及各自的优缺点和适用范围，最终能在自己的软件开发项目中对各种软件过程模式进行选择、取舍、结合、定制，得到适于该项目的、专用的最佳软件过程。

5. 理论紧密联系实际

本书对各种软件过程模式进行阐述时，既有抽象的理论原则分析，同时又引用和列举了大量生动的实际案例，对每种理论的原则内涵及运用方法进行诠释。这些案例既有成功的项目实践经验，又有失败的教训总结，特别在最后一章，从集成软件过程模式的角度汇总了软件业中常见的错误实践和规避策略。

6. 融合了编者多年在企业进行软件开发的心得

软件是一门实践性的学科，正如俗语所说，“纸上得来终觉浅”。本书是编者对自身多年在不同规模的企业中参与或大或小的项目实践的感悟。

7. 教学与参考价值

本书论述深入浅出，理论联系实际，且每章后均附有习题，因此适合作为高等院校计算机及相关专业研究生和本科高年级学生的教材，也可作为一些高级进修班的培训教材。参考学时为 32 学时。本书亦可作为软件开发人员、项目经理、软件企业领导的案头参考书籍。

本书作为教材的试用效果

本书的原稿作为讲义，在全国 35 所示范性软件学院之一——湖南大学软件学院的脱产和在职研究生班进行了试讲，学生们反响热烈，尤其是对于那些有过几年甚至十几年工作经历的在职研究生（他们当中有些已经是大中型项目的项目经理和技术主管）。他们认为本课程内容针对性很强，一方面解决了很多在以往工作中困扰他们及其所在项目组的实际难题；另一方面使他们能从实践层面上升到理论高度，从而系统、全面地接受软件开发过程的各种思想、原则、概念、方法、技巧的培训，而这些正是他们选择重新回到学校、进行进一步深造所希望实现的目标。

致谢与其他

全书由金敏主编，第 1 章和第 5 章的部分内容由周翔编写。

本书在编写过程中参考了不少相关文献资料，书中引用的文字和图片均已清楚注明

作者姓名、作品名称、文献出处并加入参考文献。这些引用是本书内容阐述所需，绝无侵权意图，特此申明。同时，对所引用的这些文献资料的作者或编者深表谢意。

在本书的出版过程中，得到了湖南大学软件学院谢冬青教授的热情支持，他在百忙中仔细阅读了全部书稿，提出了宝贵意见，在此表示衷心感谢。

本书在编写过程中力求精益求精，但由于编者水平有限，书中难免存在缺点、错误或有争议的问题，恳请读者批评指正并与我们探讨交流。

编者
2005年4月

第 1 章 绪论	1
1.1 现代软件产业的困境	1
1.1.1 困境中的现代软件产业	1
1.1.2 陷入困境的根源	2
1.2 软件生命周期模型及其局限性	3
1.2.1 困境中的消极态度	3
1.2.2 困境中的积极探索	4
1.2.3 软件过程	4
1.2.4 软件生命周期模型及其局限性	5
1.3 走出困境——软件过程模式	8
1.3.1 软件过程模式定义	8
1.3.2 几种典型的软件过程模式	8
1.3.3 提出软件过程模式的意义	10
1.4 本书的主要内容和意义	11
习题 1	12
第 2 章 软件过程模式 1——Rational 统一过程	13
2.1 Rational 统一过程概述	13
2.1.1 什么是 Rational 统一过程	13
2.1.2 Rational 统一过程的发展史	14
2.1.3 Rational 统一过程与最佳实践	15
2.1.4 Rational 统一过程应用概况	16
2.2 Rational 统一过程术语	17
2.3 Rational 统一过程规范	19
2.3.1 Rational 统一过程的生命周期——二维结构生命周期	19
2.3.2 Rational 统一过程的人员——角色及其活动	25
2.3.3 Rational 统一过程的方法——方法与工具	26
2.3.4 Rational 统一过程的产品——工件	31

2.3.5 Rational 统一过程的生命周期、人员、方法、产品四要素间的关系	36
2.4 Rational 统一过程的特点	36
2.4.1 Rational 统一过程的优点	36
2.4.2 Rational 统一过程的缺陷	37
2.5 Rational 统一过程的实施策略	38
2.6 Rational 统一过程案例——渐进交付的系统	39
2.6.1 渐进交付的案例描述	39
2.6.2 渐进交付的案例点评	41
2.7 本章小结	42
习题 2	42
第 3 章 软件过程模式 2——敏捷过程	44
3.1 敏捷过程概述	44
3.2 敏捷过程的价值观与原则	45
3.2.1 敏捷过程的基本价值观与原则	45
3.2.2 极限编程	50
3.2.3 其他敏捷过程	54
3.3 敏捷过程的特点——与 RUP 比较	63
3.3.1 敏捷过程的生命周期	63
3.3.2 敏捷过程的人员	63
3.3.3 敏捷过程的方法	64
3.3.4 敏捷过程的产品	66
3.3.5 敏捷过程的生命周期、人员、方法与产品四要素的关系	66
3.4 敏捷过程的实施策略	67
3.5 敏捷过程案例——对需求变化的两种态度	67
3.5.1 需求变化的案例描述	67
3.5.2 需求变化的案例点评	69
3.6 本章小结	70
习题 3	70
第 4 章 软件过程模式 3——微软过程	72
4.1 微软过程概述	72
4.2 微软过程术语	73
4.3 微软过程基本原则	73
4.3.1 微软过程的过程原则	74
4.3.2 微软过程的组队原则	77
4.4 微软过程的特点——与 AP、RUP 比较	80
4.4.1 微软过程的生命周期	80

4.4.2	微软过程的人员	82
4.4.3	微软过程的方法	88
4.4.4	微软过程的产品	91
4.4.5	微软过程的生命周期、人员、方法、产品四要素间的关系	91
4.5	微软过程的实施策略	93
4.6	微软过程案例——IE 浏览器经典战役	94
4.6.1	背景介绍——浏览器市场之争	94
4.6.2	IE 项目的生命周期	94
4.6.3	IE 项目的人员及组织	96
4.6.4	IE 项目的方法	97
4.6.5	IE 项目的产品	100
4.6.6	IE 项目的生命周期、人员、方法与产品四要素间的关系	100
4.7	本章小结	100
	习题 4	101
第 5 章	最佳软件过程模式	102
5.1	RUP、AP 与 MP 三者的集成	102
5.1.1	集成过程模式的生命周期	102
5.1.2	集成过程模式的人员	104
5.1.3	集成过程模式的方法	107
5.1.4	集成过程模式的产品	112
5.1.5	集成过程模式的生命周期、人员、方法与产品四要素间的关系	113
5.2	集成软件过程模式案例	113
5.2.1	高压变电站监控系统开发项目案例	113
5.2.2	热电厂再生废水处理自控系统集成项目案例	115
5.3	寻求最佳软件过程模式	117
	习题 5	117
	参考文献	118

绪 论

1946年,世界上第一台电子计算机诞生在美国宾夕法尼亚大学的摩尔学院,由此拉开了计算机软件的发展史。从宏观角度而言,计算机软件的发展主要经历了以下三个阶段^[1]。

(1) 第一阶段——程序设计阶段

20世纪60年代以前还没有软件开发的说法,那时只有程序设计的概念,最多在写出程序后配有程序结构说明和使用说明。经典的程序设计方法为“程序设计=数据结构+算法”。

(2) 第二阶段——软件工程阶段

20世纪70年代以来,人们认识到软件的工作不能仅限于编写程序,软件开发工作在程序编写之前和之后还有很多重要的工作不能忽略,例如需求分析、测试、维护等等。在总结“软件危机”教训后,人们认识到并建立了软件工程的思想。软件工程摒弃了认为只有充满编程技巧的程序才能高水平地发挥个人才能的观念,强调程序的可读性、可理解性、可测试性和易修改性等工程化的原则。

(3) 第三阶段——软件过程阶段

从20世纪90年代开始,人们更加强调软件开发的效率、软件的质量以及与软件开发相关的管理工作,建立了“软件过程”的概念。软件过程不仅包括软件开发过程,还包括了支持性、管理性过程。

以上发展历程表明,通过实践、总结、再实践、再总结……人们对软件这门实践学科的理解正朝着更全面、更系统、更深刻的方向发展。

1.1 现代软件产业的困境

1.1.1 困境中的现代软件产业

当今,全球市场变幻莫测,用户需求日趋复杂,IT技术日新月异。软件企业组织在不断变化的市场和技术环境中能否取得成功,关键在于企业组织是否能在市场许可的期

限和有限资源条件下不断推出满足用户需求的产品。

然而，现代软件产业的总体情况并不理想。下面先来看一个真实的案例^[14]。

Square Cal 3.0 版本计划在 2.0 版本上市后的 10 个月内发布。项目经理 Mickey 和上司 Kim 讨论后决定：他们将为项目组成员提供私人办公室、最新型的计算机以及免费的碳酸饮料，并且要求开发者在前 8 个月按照预先设计好的接口各自开发，8 个月之后进行可视化锁定，在最后 2 个月内完成系统集成。这是一个完美的计划。于是项目组成员各自做着自己的工作。随着可视化锁定日期的来临，他们开始进行代码集成。他们在可视化锁定最终截止日期前一天的下午两点开始工作，但很快发现程序不能编译通过，更不用说运行了。代码在编译时有数十个错误，而似乎每处理一个错误就会产生十个以上的新错误。他们一直干到午夜也没有结果，只好决定第二天再说。但测试发现问题的速度远比开发人员解决问题的速度快，解决系统某一部分的错误经常会导致其他部分的新问题。项目超期，项目组成员在巨大的压力下工作，士气逐渐低落。最后整个软件开发过程用了 15 个月时间，即超过了项目计划时间的 50%，公司错过了最佳的产品发布日期。产品发布后，用户对 Square Cal 3.0 版本反映冷淡，在几个月的时间内其市场份额从第二位下降到第四位。

再看一组统计数据。图 1-1^[12]是根据最近一次项目调查得到的某公司所有软件项目的完成情况。从图中可以看出，出现问题的项目和中途取消的项目所占比例远高于顺利完成的项目。实际上，图 1-1 代表了整个现代软件产业的现状，很多软件项目最终不能交付，或者最终交付的软件项目进度发生延期、成本超出预算，而且运行经常不可靠。因此，诸如“软件开发的滑铁卢”（Software Runaways）^[2]、“死亡之旅”（Death March）^[3]之类关于软件失败项目的报道不时见诸于媒体报端也就不足为奇。

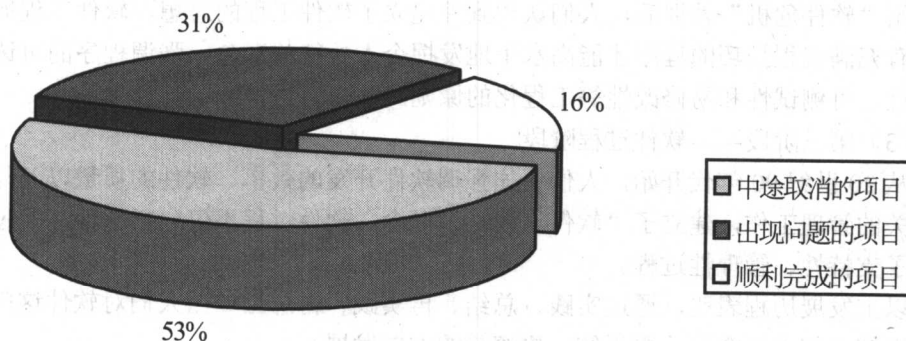


图 1-1 项目完成情况图

1.1.2 陷入困境的根源

众所周知，现代计算机和因特网的性能在逐年增强，用户对运行于计算机和因特网上的软件的功能和性能的渴望也随之不断增加，用户希望得到越来越复杂的软件系统以满足其需求；与此同时，市场的激烈竞争迫使现代软件企业必须更快地生产出用户需要

的复杂软件。然而，现今大多数软件企业及其开发人员仍然沿用二三十年前的软件组织方式和开发方法来开发软件系统，这就是现代软件产业陷入困境的根源所在。

具体总结、归纳起来^{[2], [5], [12]}，大多数软件项目失败的根本原因主要有以下几个方面。

- 不完整、不现实的项目需求描述：需求分析过程中缺乏用户参与或与用户的交流过于模糊，常导致给出的需求描述不完整、不准确，需求项过多、需求项实现难度过高等。
- 对需求的变更束手无策：需求的变更往往是现实世界变化的反映，合理的需求变更在项目的某些阶段是允许的，这要求开发过程对需求变更具有应对能力。
- 脆弱的架构：表现为程序模块互不兼容，软件不易扩展、裁剪和移植。
- 采用不成熟的技术：表现为新技术不具有要求的功能、新技术存在局限性或者新技术是问题的错误解决方案。
- 测试的不充分性：未检测出需求、设计和实现三者之间的不一致。
- 拙劣的进度计划和评估：主要表现为对项目进度的评估过于乐观，正如霍夫斯塔特（Hofstadter）定律^[2]所指出的那样，“开发软件的时间总比想象的时间长，即使注意了霍夫斯塔特定律也是如此”。
- 缺乏资源：产品开发项目需要一定的投入，包括在经费、人员、场地、时间等资源方面的投入，尤其是在资深人员方面的投入。
- 不具备项目管理方法：包括对风险的预估和驾驭、对软件质量的度量等项目管理方法。
- 缺少管理层的支持：在软件企业中从事产品开发工作，必须获得企业高层管理者的支持，项目和产品本身也必须符合企业总体的发展规划和经营目标，否则项目就会夭折。

1.2 软件生命周期模型及其局限性

在具体探讨现代软件产业走出困境的解决方案时，有必要先回顾、总结一下现有软件工程领域的研究成果以及它们在解决现代软件产业面临的困境方面存在的局限性，以此作为提出走出困境的新的途径的思想基础。

1.2.1 困境中的消极态度

软件开发工作对软件开发人员来说通常不是一件很有趣的事情^[8]。项目经常在开始一段时间后遇到障碍，这时消极的开发人员开始指责别人。容易被指责的对象包括：愚蠢、粗暴的老板——他们的能力勉强够给自己系鞋带；在其他部门的纸堆里的呆子——他们总是要求过多的文档；愚蠢的用户——他们经常不知道自己想要什么，而在他们能够确切说明自己想要什么的时候，这些要求又总是没有意义的。当然，这些开发人员在指责别人的时候从不指责自己，他们相信自己是完美的。指责完之后该做些什么呢？有的倾尽全力，加班加点，为满足交给自己的、不切实际的要求做徒劳的努力，从而踏上

一条死亡之旅：有的则与项目完全脱离关系，做一些不相干的事，因为知道项目注定要失败，所以开始学习一些新的技术，至少可以用来充实自己的简历，以备项目组解散后被调换到另一个项目组或跳槽到另一家公司工作。不幸的是，上一个项目的历史在新项目过程中再一次被重演，新项目开始遇到障碍，他们又开始了新一轮的从指责到颓废，如此周而复始，恶性循环。

1.2.2 困境中的积极探索

正当消极的人在软件开发的泥潭中相互指责而越陷越深、难以自拔时，另一部分人开始了积极的反思与新的探求。难道真的全是别人的过错吗？自己有无责任？如何使项目走出困境？如何避免新的项目再次陷入泥潭？

回顾图 1-1，尽管最终失败或出现问题的项目占 84%，但毕竟还有 16% 的成功项目，一个最明显的例证即微软公司^[13]。微软公司在 1975 年只有 3 名员工，营业额仅 16 000 美元；到 1989 年已经有 8000 名员工，营业额达 80 亿美元；而发展至 2000 年时员工已有 35 000 名，营业额达 240 亿美元，获利更高达 150 亿美元，成为世界上最大的软件公司。这一发展过程堪称世界软件业奇迹之首。除微软公司外，也不乏其他优秀的软件企业，如嵌入式软件产品与服务的提供商——风河（Wind River）公司就属于软件业中的后起之秀。那么，这些优秀的软件企业是如何组织开展软件产品开发的？通过深入挖掘这些企业软件开发项目成功的经验，同时深刻剖析不成功的软件开发项目失败的原因，人们总结出了很多软件开发实践经验，并决定将这些实践应用于新的软件开发项目中，以期以一种可预测的方式指导新的软件开发项目并获得成功，同过通过实践检验来不断完善这些经验。最终，这些“软件开发实践经验总结”不断演变，其体系化的研究成果表现为“软件过程”概念的诞生和一些“软件生命周期模型”的相继推出。

1.2.3 软件过程

1. 定义

定义 1-1 软件过程是从软件项目需求定义开始直至软件使用后被废弃为止，跨越整个软件生存期内的系统开发、运行和维护等全部活动及相关项的总和。

2. 内容^[1]

软件过程包括 5 个主要过程、8 个支持过程和 4 个组织过程。

- 5 个主要过程为获取过程、供应过程、开发过程、运行过程、维护过程。
- 8 个支持过程为文档编制过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审核过程、问题解决过程。
- 4 个组织过程为管理过程、基础设施过程、改进过程、培训过程。

3. 软件过程能力评估标准和改进方案

以下为三种最具影响力的软件过程能力评估标准和改进方案。

- CMM (Carnegie Mellon University, 能力成熟度模型): 1987 年由美国卡内基梅隆大学 (Carnegie Mellon University) 提出, 适用范围为国际贸易中的软件。CMM 将软件能力成熟度从低到高分为五级, 分别为初始级、可重复级、已定义级、已定量管理级、优化级, 软件企业可按这五级对其软件过程进行持续改进。
- ISO 9000: 1987 年由国际标准化组织 (ISO) 颁布, 适用范围不仅包括国际贸易中的软件, 同时也包括国际贸易中的硬件和服务。
- 6 σ : 六西格玛 (Six Sigma, 6 σ) 起源于制造业, 其本质是一种采用统计学技术的质量度量和管理方法。在 20 世纪 90 年代中期, 通用电气公司成功地将 6 σ 从一种质量度量管理方法提升为一种高度有效的企业过程设计、改造和优化的方法体系, 继而推广到各个行业; 其思想核心是将所有工作作为一种过程或流程, 采用量化的方法分析过程中影响质量的因素, 找出最关键的因素加以改进, 从而达到更高的客户满意度。

1.2.4 软件生命周期模型及其局限性

1. 定义

定义 1-2 软件生命周期模型是软件过程中全部活动的生命周期结构框架的一种形式化描述, 也称为软件生存期模型^[1]。

2. 种类

迄今为止, 已提出多种软件生命周期模型, 最典型的包括瀑布模型、演化模型、螺旋模型、喷泉模型等。

(1) 瀑布模型

如图 1-2 所示, 瀑布模型规定了软件生命周期各阶段的不同活动, 包括定义阶段的项目计划和需求分析, 开发阶段的设计、编码和测试, 维护阶段的运行维护。这些活动自上而下, 相互衔接, 呈线性图状, 如同瀑布流水, 逐级下落。

软件开发的实践证明, 瀑布模型可用于指导用户需求较明确、稳定的软件项目。

尽管瀑布模型的适用范围有很大的局限性, 但其思想精髓“线性化”是多种软件生命周期模型 (如演化模型、螺旋模型、喷泉模型等) 的基本细粒度元素, 对此应深刻领会并灵活运用。

(2) 演化模型

如图 1-3 所示, 演化模型包括两大步骤: 第一步进行试验开发, 得出产品“原型”, 其目标在于弄清软件需求并探索可行性; 第二步在原型基础上开发出较为满意的软件产品。因此演化模型又称为原型模型。