

21世纪

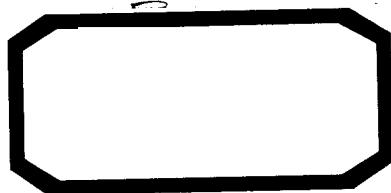
高等院校计算机系列教材

软件系统 分析与设计

殷建民 主 编
刘东瑰 赵良军 欧阳默 副主编



中国水利水电出版社
www.waterpub.com.cn



21 世纪高等院校计算机系列教材

软件系统分析与设计

殷建民 主 编

刘东瑰 赵良军 欧阳默 副主编

中国水利水电出版社

内 容 提 要

本书简明扼要地介绍了软件工程与项目管理的基础知识,着重讨论了软件系统分析与设计的方法、过程、工具与文档,并给出了一个完整的软件系统分析与设计案例。

本书强调理论与实际的结合,强调系统分析与设计技术的实用化、工具化与文档化,适合作为高校计算机学院、软件学院、软件职业技术学院计算机科学与技术、计算机应用、计算机软件和软件工程等相关专业本科、专科“软件系统分析与设计”课程的参考教材,也可作为软件开发人员的自修教材与工具书,还可作为全国计算机技术与软件专业技术资格(水平)考试系统设计师和系统分析师专业的辅导教材。

本书配有电子教案,读者可以从中国水利水电出版社网站免费下载,网址为:
<http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

软件系统分析与设计 / 殷建民主编. —北京: 中国水利水电出版社, 2008

(21 世纪高等院校计算机系列教材)

ISBN 978-7-5084-5381-1

I. 软… II. 殷… III. ①软件工程—系统分析—高等学校—教材②软件设计—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2008) 第 035680 号

书 名 作 者 出 版 发 行	软件系统分析与设计 殷建民 主 编 刘东瑰 赵良军 欧阳默 副主编 中国水利水电出版社(北京市三里河路6号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
经 售	
排 版 刷 格 次 印 数 定 价	北京万水电子信息有限公司 北京诚顺达印刷有限公司 787mm×1092mm 16 开本 15.75 印张 382 千字 2008 年 4 月第 1 版 2008 年 4 月第 1 次印刷 0001—4000 册 25.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

本书依据高校计算机学院、软件学院、软件职业技术学院计算机科学与技术、计算机应用、计算机软件和软件工程等相关专业本科、专科“软件系统分析与设计”课程和“软件工程”课程教学大纲中的相关内容以及全国计算机技术与软件专业技术资格（水平）考试系统设计师和系统分析师专业考试大纲中的相关知识点，简单介绍了软件工程与项目管理的基础知识，着重讨论了软件系统分析与设计的方法、过程、工具与文档，并给出了一个完整的软件系统分析与设计案例。

本书共分四部分。第1~4章为基础知识篇，第5~7章为系统分析篇，第8~12章为系统设计篇，第13章为实战案例篇。

第1章：软件工程基础知识，包括软件工程知识体系、软件生存周期与软件开发模型、软件质量模型与软件质量管理、软件配置管理和软件过程管理等内容。

第2章：项目管理基础知识，包括项目与项目管理、项目管理过程与过程组以及项目管理知识体系等内容。

第3章：软件开发技术，包括软件开发平台、中间件技术和构件技术等内容。

第4章：软件项目规划，包括项目策划、项目可行性分析、项目论证、评估与立项以及项目开发计划等内容。

第5章：系统分析方法学，包括系统需求分析与软件需求分析、结构化分析、原型化方法和面向对象的分析等内容。

第6章：统一建模语言，包括统一建模语言概述、用例图、类图和对象图、交互图、状态图、活动图、构件图和部署图等内容。

第7章：系统分析文档，包括系统/子系统需求规格说明、接口需求规格说明和软件需求规格说明等内容。

第8章：系统设计基础，包括系统设计概述、软件设计概述、设计原则和设计视图等内容。

第9章：系统设计方法，包括结构化设计、面向数据结构的设计、面向对象的设计和 design 模式等内容。

第10章：数据库设计，包括数据建模、数据规范化和数据库设计过程等内容。

第11章：用户界面设计，包括基本概念、界面设计技术、界面设计评估和界面设计案例等内容。

第12章：系统设计文档，包括系统/子系统（结构）设计说明、接口设计说明、软件（结构）设计说明和数据库设计说明等内容。

第13章：电力负荷管理系统的分析与设计，包括应用描述、过程与方法、系统分析和系统设计等内容。

软件系统分析与设计是一门实践性很强的工程学科，本书强调理论与实际的结合，强调分析与设计技术的实用化、工具化与文档化。

本书中提到的关于软件系统分析与设计技术的所有国家标准，编者均进行了仔细核实，证明是当前有效的最新版本的国家标准。例如，本书引用最新的《GB/T 8567-2006 计算机软件文档编制规范》取代了原来的《GB/T 8567-1988 计算机软件产品开发文件编制指南》。

为使读者了解目前软件技术与软件行业的发展方向以及国家各类科技计划对软件项目的支持重点，本书还简明扼要地讲述了“国家中长期科学和技术发展规划纲要（2006—2020 年）”、“国家‘十一五’科学技术发展规划”、“信息产业科技发展‘十一五’规划和 2020 年中长期规划纲要”和 2008 年度电子信息产业发展基金项目指南中与软件相关的优先主题、重大专项、前沿技术和研发项目。

本书由殷建民任主编，刘东瑰、赵良军、欧阳默任副主编。其中，第 1、2、4、5、6、7、8、12 章由殷建民编写，第 9、11 章由刘东瑰编写，第 3、10 章由赵良军编写，第 13 章由欧阳默编写。

本书主编为具有 25 年计算机软件开发经验的技术专家，主持完成过国家电子发展基金项目、国家技术创新基金项目和教育部、国家语委科研项目，曾获国务院特殊津贴、国家科学技术进步奖、钱伟长中文信息科学技术奖和王选新闻科学技术一等奖。其他几位编者也都具有多年软件项目开发的实战经验。

在本书编写过程中，编者参考了许多相关的文献和书籍，在此对这些参考文献的作者表示衷心感谢。同时感谢中国水利水电出版社在本书出版过程中所给予的支持和帮助。

由于编者水平有限，且本书涉及的内容很广，书中难免存在错漏和不妥之处，恳请各位专家和广大读者批评指正，我们将十分感激。

编者

2008 年 2 月

目 录

前言

第 1 章 软件工程基础知识	1
1.1 软件工程知识体系	1
1.2 软件生存周期与软件开发模型	2
1.2.1 软件生存周期	2
1.2.2 软件开发模型	4
1.3 软件质量模型与软件质量管理	7
1.3.1 软件质量模型	7
1.3.2 软件质量管理	10
1.4 软件配置管理	12
1.4.1 软件配置项与基线	12
1.4.2 软件配置管理过程	13
1.5 软件过程管理	15
1.5.1 软件能力成熟度模型	15
1.5.2 软件过程与软件能力成熟度评估	16
1.5.3 软件过程改进	17
1.6 小结	18
第 2 章 项目管理基础知识	20
2.1 项目与项目管理	20
2.1.1 项目	20
2.1.2 项目管理	22
2.2 项目管理过程与过程组	23
2.2.1 过程与过程组	23
2.2.2 项目管理过程的交互作用	25
2.2.3 项目管理过程的裁剪	26
2.3 项目管理知识体系	27
2.4 小结	28
第 3 章 软件开发技术	29
3.1 软件开发平台	29
3.1.1 Microsoft .NET 平台	29
3.1.2 J2EE 平台	31

3.1.3	Microsoft .NET 与 J2EE 的异同.....	35
3.2	中间件技术	38
3.2.1	中间件简介	38
3.2.2	消息代理中间件	40
3.2.3	面向数据库的中间件	43
3.3	构件技术	44
3.3.1	构件库	45
3.3.2	构件模型	46
3.3.3	构件的属性与特点	46
3.3.4	构件与中间件	47
3.4	小结	49
第 4 章	软件项目规划	50
4.1	项目策划	50
4.1.1	从政策导向中寻找项目机会	50
4.1.2	从市场需求中寻找项目机会	54
4.1.3	从技术发展中寻找项目机会	54
4.1.4	从特定事件中寻找项目机会	55
4.2	项目可行性分析	55
4.2.1	技术可行性分析	55
4.2.2	项目投资及效益分析	57
4.3	项目论证、评估与立项	60
4.3.1	项目论证与评估的基本概念	60
4.3.2	项目可行性报告的真实性评估	60
4.3.3	项目可行性报告的客观性评估	62
4.3.4	项目评估报告	63
4.3.5	项目立项	63
4.4	项目开发计划	63
4.5	小结	67
第 5 章	系统分析方法学	68
5.1	系统需求分析与软件需求分析	68
5.2	结构化方法学	70
5.3	原型化方法	73
5.3.1	原型化方法与结构化方法的比较	73
5.3.2	原型生命周期及其策略	75
5.4	面向对象的分析	80
5.4.1	面向对象方法学概述	80
5.4.2	面向对象的分析	82

5.5	小结	84
第 6 章	统一建模语言	85
6.1	概述	85
6.2	用例图	86
6.3	类图和对象图	89
6.4	交互图	92
6.5	状态图	93
6.6	活动图	94
6.7	构件图	96
6.8	部署图	96
6.9	小结	97
第 7 章	系统分析文档	98
7.1	系统/子系统需求规格说明	98
7.2	接口需求规格说明	104
7.3	软件需求规格说明	107
7.4	小结	113
第 8 章	系统设计基础	115
8.1	系统设计概述	115
8.1.1	系统级设计决策	115
8.1.2	系统架构设计	115
8.1.3	运行设计	117
8.1.4	系统出错处理设计	117
8.1.5	系统维护设计	117
8.2	软件设计概述	118
8.2.1	软件级设计决策	118
8.2.2	软件架构设计	118
8.2.3	软件详细设计	120
8.3	设计原则	120
8.3.1	组件化	120
8.3.2	抽象	121
8.3.3	内聚与耦合	121
8.3.4	封装与信息隐蔽	122
8.3.5	启发式规则	123
8.4	设计视图	123
8.4.1	架构视图	123
8.4.2	行为视图	124
8.5	小结	124

第 9 章 系统设计方法	125
9.1 结构化设计	125
9.1.1 概述	125
9.1.2 系统结构图	125
9.1.3 系统结构图分类	127
9.2 面向数据结构的设计	129
9.2.1 概述	129
9.2.2 Jackson 图	130
9.2.3 Jackson 方法	131
9.3 面向对象的设计	132
9.3.1 概述	132
9.3.2 面向对象设计技术	133
9.3.3 面向对象设计的设计过程	135
9.4 设计模式	137
9.4.1 概述	137
9.4.2 设计模式基本组成	137
9.4.3 设计模式分类	139
9.4.4 如何使用设计模式	143
9.5 小结	143
第 10 章 数据库设计	144
10.1 数据建模	144
10.1.1 数据模型分类	145
10.1.2 实体—联系模型	146
10.1.3 数据模型	147
10.2 数据规范化	151
10.2.1 数据规范化的几个基本概念	152
10.2.2 范式	153
10.3 数据库设计过程	154
10.3.1 数据库的需求分析	154
10.3.2 数据库的概念设计	156
10.3.3 数据库的逻辑设计	158
10.3.4 数据库的物理设计	159
10.4 小结	159
第 11 章 用户界面设计	160
11.1 基本概念	160
11.1.1 设计目标	160
11.1.2 设计原则	163

11.1.3 设计过程.....	164
11.2 设计技术	165
11.2.1 界面设计分析技术.....	165
11.2.2 界面设计方法.....	169
11.3 设计评估	175
11.4 设计案例	177
11.4.1 用户界面示例.....	177
11.4.2 界面案例分析.....	179
11.5 小结	180
第 12 章 系统设计文档	181
12.1 系统/子系统设计（结构设计）说明	181
12.2 接口设计说明	186
12.3 软件（结构）设计说明	188
12.4 数据库设计说明	193
12.5 小结	198
第 13 章 电力负荷管理系统的分析与设计	199
13.1 应用描述	199
13.1.1 系统概述	199
13.1.2 系统功能需求	202
13.1.3 系统非功能需求及约束条件.....	204
13.1.4 系统层次结构	204
13.1.5 系统典型配置	205
13.2 过程与方法	206
13.2.1 开发过程	206
13.2.2 需求分析	210
13.2.3 系统设计	213
13.3 系统分析	215
13.3.1 用例模型	216
13.3.2 动态模型	222
13.3.3 静态模型	224
13.3.4 候选架构	226
13.3.5 补充规约	227
13.4 系统设计	228
13.4.1 动态模型	228
13.4.2 静态模型	229
13.4.3 系统体系架构	233
13.4.4 用户界面设计	234

13.4.5 数据库接口设计	234
13.4.6 补充规约	236
13.5 小结	237
参考文献	239

第 1 章 软件工程基础知识

软件工程学研究如何有效地组织和管理软件开发的工程学科，又称软件工程、软件工程方法学。本章将介绍软件工程知识体系、软件生命周期与软件开发模型、软件质量模型与软件质量管理、软件配置管理和软件过程管理。

1.1 软件工程知识体系

“软件工程”概念的提出与 20 世纪 60 年代出现的“软件危机”有着密不可分的联系。正是为了摆脱软件危机的困扰，软件工程学作为一门研究软件开发与维护的普遍原理和技术的工程学科，从 20 世纪 60 年代末期开始迅速发展起来。

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题绝不仅仅是不能正常运行的软件才具有的，实际上，几乎所有软件都不同程度地存在这些问题。

自从强大的第三代计算机硬件问世以后，许多原来难以实现的计算机应用成为现实，同时对软件系统的需求数量和复杂度要求变得更高。而当时软件开发技术无法满足这一日益增长的需求，这就引发了软件危机。

软件危机的主要表现为：

- (1) 软件开发生产率提高的速度远远跟不上计算机迅速普及的趋势。软件需求的增长得不到满足，软件产品“供不应求”的现象使人类不能充分利用现代计算机硬件提供的巨大潜力。
- (2) 软件成本在计算机系统总成本中所占的比例逐年上升。
- (3) 不能正确估计软件开发产品的成本和进度，致使实际开发成本高出预算很多，而且超出预期的开发时间要求。
- (4) 软件开发人员和用户之间的信息交流往往很不充分，用户对“已完成的”软件系统不满意的现象经常发生。
- (5) 软件产品的质量不易保证。
- (6) 软件产品常常是不可维护的。
- (7) 软件产品的重用性差，同样的软件多次重复开发。
- (8) 软件通常没有适当的文档资料。

软件危机的产生一方面是由于软件开发本身的复杂性，另一方面与当时的手工作坊式软件开发模式有着密切关系。开发一个具有一定规模和复杂性的软件系统和编写一个简单的程序大不一样。其间的差别，借用 Booch 的比喻，如同建造一座大厦和搭一个狗窝的差别。大型的、复杂的软件系统的开发是一项工程，必须按工程学的方法组织软件的生产与管理，必须经过计划、分析、设计、编程、测试、维护等一系列的软件生命周期阶段。这是人们从软件危机中获得的最重要的教益，这一认识促使了软件工程学的诞生。

软件工程学就是研究如何有效地组织和管理软件开发的工程学科。IEEE 在 1983 年将软件工程定义为：软件工程是开发、运行、维护和修复软件的系统方法。

著名的软件工程专家 Boehm 于 1983 年提出了软件工程的 7 条基本原理：

- (1) 用分阶段的生命周期计划严格管理。
- (2) 坚持进行阶段评审。
- (3) 实行严格的产品控制。
- (4) 采用现代程序设计技术。
- (5) 结果应能清楚地审查。
- (6) 开发小组的人员应该少而精。
- (7) 承认不断改进软件工程实践的必要性。

软件工程方法学包含 3 个要素：方法、工具和过程。方法是指完成软件开发的各项任务的技术方法，工具是指为运用方法而提供的软件工程支撑环境，过程是指为获得高质量的软件所需要完成的一系列任务的框架。目前，软件工程学已成为计算机科学技术的一个重要分支。

根据 IEEE 计算机学会 (IEEE Computer Society) 职业实践委员会 (Professional Practices Committee) 发布的《软件工程知识体系指南(2004 版)》(Guide to the Software Engineering Body of Knowledge 2004 Version)，软件工程包括以下 10 个知识域：

- (1) 软件需求 (Software Requirements)。
- (2) 软件设计 (Software Design)。
- (3) 软件构造 (Software Construction)。
- (4) 软件测试 (Software Testing)。
- (5) 软件维护 (Software Maintenance)。
- (6) 软件配置管理 (Software Configuration Management)。
- (7) 软件工程管理 (Software Engineering Management)。
- (8) 软件工程过程 (Software Engineering Process)。
- (9) 软件工程工具和方法 (Software Engineering Tools and Methods)。
- (10) 软件质量 (Software Quality)。

1.2 软件生存周期与软件开发模型

1.2.1 软件生存周期

软件生存周期 (Software Life Cycle)，也称软件生命周期，是人们在研究软件开发过程时所发现的一种规律性的事实。如同人的一生要经历婴儿期、少年期、青年期、老年期直至死亡这样一个全过程一样，一个软件产品也要经历计划、分析、设计、编程、测试、维护直至被淘汰这样一个全过程。软件的这一全过程被称为软件生存周期。

目前，软件生存周期各阶段的划分尚不统一，有的分得粗些，有的分得细些，所包含的实际内容也不完全相同。

(1) 1970 年, Boehm 提出了如图 1-1 所示的软件生存周期模型。

计划时期		开发时期					运行时期
问题定义	可行性研究	需求分析	总体设计	详细设计	编码	测试	维护

图 1-1 Boehm 定义的软件生存周期模型

(2) 1988 年制定和公布的国家标准《GB8566-1988 计算机软件开发规范》中将软件生存周期划分为如图 1-2 所示的 8 个阶段。

可行性研究与计划	需求分析	概要设计	详细设计	实现	组装测试	确认测试	使用和维护
----------	------	------	------	----	------	------	-------

图 1-2 GB8566-1988 定义的软件生存周期模型

(3) 20 世纪 90 年代初有了软件工程过程的概念之后, 1995 年制定和公布的国家标准《GB/T 8566-1995 信息技术 软件生存周期过程》定义了软件生存周期的 7 个主要过程, 如图 1-3 所示。

获取过程	供应过程	开发过程	运行过程	维护过程
管理过程				
支持过程				

图 1-3 GB/T 8566-1995 定义的软件生存周期过程模型

其中, “获取过程”和“供应过程”分别描述了软件的“获取者”(用户)和“供应者”(开发者)在开发之前的主要活动和任务, 而“管理过程”和“支持过程”则贯穿于整个软件生存周期。

(4) 1995 年国际标准化组织对软件生存周期过程做了调整, 公布了新的国际标准《ISO/IEC 12207 信息技术 软件生存周期过程》。2001 年 11 月, 国家质量监督检验检疫总局将其等同采用为国家标准《GB/T 8566-2001 信息技术 软件生存周期过程》。该标准全面、系统地阐述了软件生存周期的 3 组共 17 个过程活动和任务, 如表 1-1 所示。

表 1-1 GB/T 8566-2001 定义的软件生存周期过程模型

主要过程	获取过程、供应过程、开发过程、运行过程、维护过程
支持过程	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
组织过程	管理过程、基础设施过程、改进过程、培训过程

(5) 1999 年, Rational 软件公司的 3 位软件工程大师 Ivar Jacobson、Grady Booch 和 James Rumbaugh 联合编写了一部划时代的著作《统一软件开发过程》(The Unified Software Development Process, UP), 将他们多年研究所得的软件开发方法学融合在一起。该书清楚地说明了支持整个软件生存周期的统一软件开发过程是一个用例驱动的、以架构为中心的、迭代与增量的开发过程。统一软件开发过程是在重复一系列组成软件生存周期的循环, 每次循环都包括如表 1-2 所示的 4 个阶段和如表 1-3 所示的 5 种工作流。

表 1-2 UP 定义的软件生存周期模型的 4 个阶段

阶段	说明
初始阶段 (inception phase)	捕捉用例, 思考系统架构.....
细化阶段 (elaboration phase)	细化用例, 设计系统架构.....
构造阶段 (construction phase)	程序设计, 实现, α 测试.....
移交阶段 (transition phase)	β 测试.....

表 1-3 UP 定义的软件生存周期模型的 5 种工作流

工作流	说明
需求工作流 (requirements workflow)	捕捉需求, 使开发导向正确的系统
分析工作流 (analysis workflow)	生成一个有助于架构设计的需求描述
设计工作流 (design workflow)	建立系统设计模型
实现工作流 (implementation workflow)	实现系统
测试工作流 (test workflow)	验证实现的结果

软件生存周期中各阶段的划分标准不统一, 名称也不一致, 一般可将软件生存周期分为需求分析、概要设计 (系统设计)、详细设计 (组件设计)、编码、测试和维护等阶段。本书主要讲述软件的需求分析与设计。

1.2.2 软件开发模型

为了指导软件的开发, 可以用不同的方式将软件生命周期中的所有开发活动组织起来, 从而形成不同的软件开发模型。常见的开发模型有瀑布模型 (Waterfall Model)、快速原型模型 (Rapid Prototype Model)、演化模型 (Evolutionary Model)、增量模型 (Incremental Model)、螺旋模型 (Spiral Model) 和喷泉模型 (Water Fountain Model) 等。

1. 瀑布模型

瀑布模型严格遵循软件生命周期各阶段的固定顺序: 计划、分析、设计、编程、测试和维护, 上一阶段完成后才能进入到下一阶段, 整个模型就像一个飞流直下的瀑布, 如图 1-4 所示。

瀑布模型有许多优点: 可强迫开发人员采用规范的方法; 严格规定了各阶段必须提交的文档; 要求每个阶段结束后, 都要进行严格的评审。

但瀑布模型过于理想化, 而且缺乏灵活性, 无法在开发过程中逐渐明确用户难以确切表达或一时难以想到的需求, 直到软件开发完成之后才发现与用户的需求有很大差距, 此时必须付出高额的代价才能纠正这一偏差。

2. 快速原型模型

快速原型是指快速建立起来的可以在计算机上运行的程序, 它所完成的功能往往是最终软件产品功能的一个子集。快速原型模型的第一步是快速建立一个能反映用户主要需求的软件原型, 让用户在计算机上使用它, 通过实际操作了解目标系统的概貌。开发人员按照用户提出

的意见快速地修改原型系统，然后再次请用户试用……，一旦用户认为这个原型系统确实能够满足他们的需求，开发人员便可据此书写软件需求说明，并根据这份文档开发出可以满足用户真实需求的软件产品。

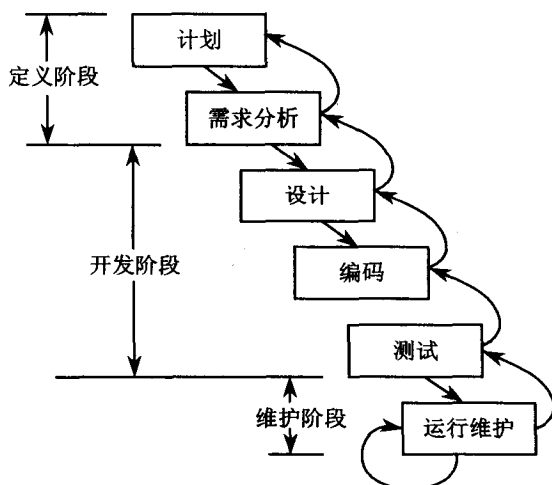


图 1-4 瀑布模型示意图

原型化方法基于这样一种客观事实：并非所有的需求在系统开发之前都能准确地说明和定义。因此，它不追求也不可能要求对需求的严格定义，而是采用了动态定义需求的方法。

具有广泛技能高水平的原型化人员是原型实施的重要保证。原型化人员应该是具有经验与才干、训练有素的专业人员。衡量原型化人员能力的重要标准是他是否能够从用户的模糊描述中快速获取实际的需求。

3. 演化模型

演化模型也是一种原型化开发方法，但与快速原型模型略有不同。在快速原型模型中，原型的用途是获知用户的真正需求，一旦需求确定了，原型即被抛弃。而演化模型的开发过程则是从初始模型逐步演化为最终软件产品的渐进过程。也就是说，快速原型模型是一种“抛弃式”的原型化方法，而演化模型则是一种“渐进式”的原型化方法。

4. 增量模型

增量模型是第三种原型化开发方法，但它既非“抛弃式”的，也非“渐进式”的，而是“递增式”的。增量模型把软件产品划分为一系列的增量构件，分别进行设计、编程、集成和测试。每个构件由多个相互作用的模块构成，并且能够完成特定的功能。如何将一个完整的软件产品分解成增量构件，因软件产品的特点和开发人员的习惯而异，但使用增量模型的软件体系结构必须是开放的，加入新构件的过程必须简单方便，新的增量构件不得破坏已经开发出来的产品。增量模型示意图如图 1-5 所示。

5. 螺旋模型

螺旋模型综合了瀑布模型和演化模型的优点，还增加了风险分析。螺旋模型包含 4 个方面的活动：制定计划、风险分析、实施工程、客户评估。这 4 项活动恰好可以放在一个直角坐

标系的 4 个象限，而开发过程恰好像一条螺旋线。采用螺旋模型时，软件开发沿着螺旋线自内向外旋转，每转一圈都要对风险进行识别和分析，并采取相应的对策。螺旋线第一圈的开始点可能是一个概念项目。从第二圈开始，一个新产品开发项目开始了，新产品的演化沿着螺旋线进行若干次迭代，一直运转到软件生命周期结束，如图 1-6 所示。

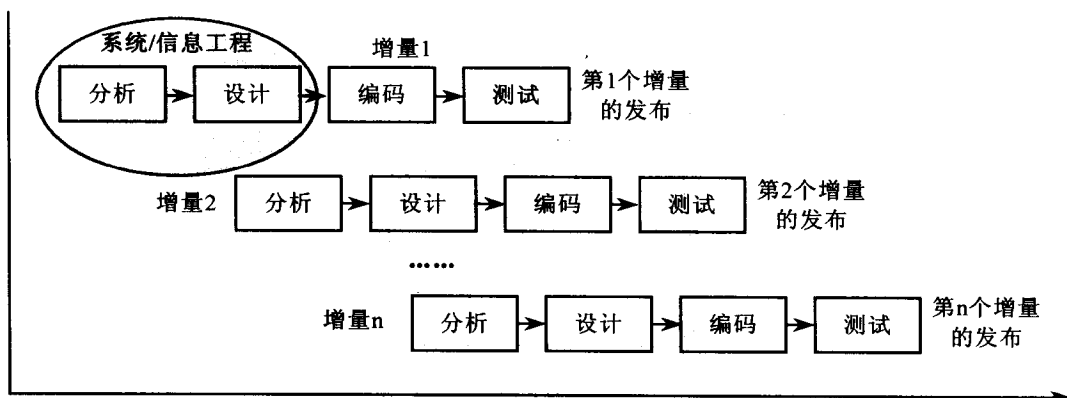


图 1-5 增量模型示意图

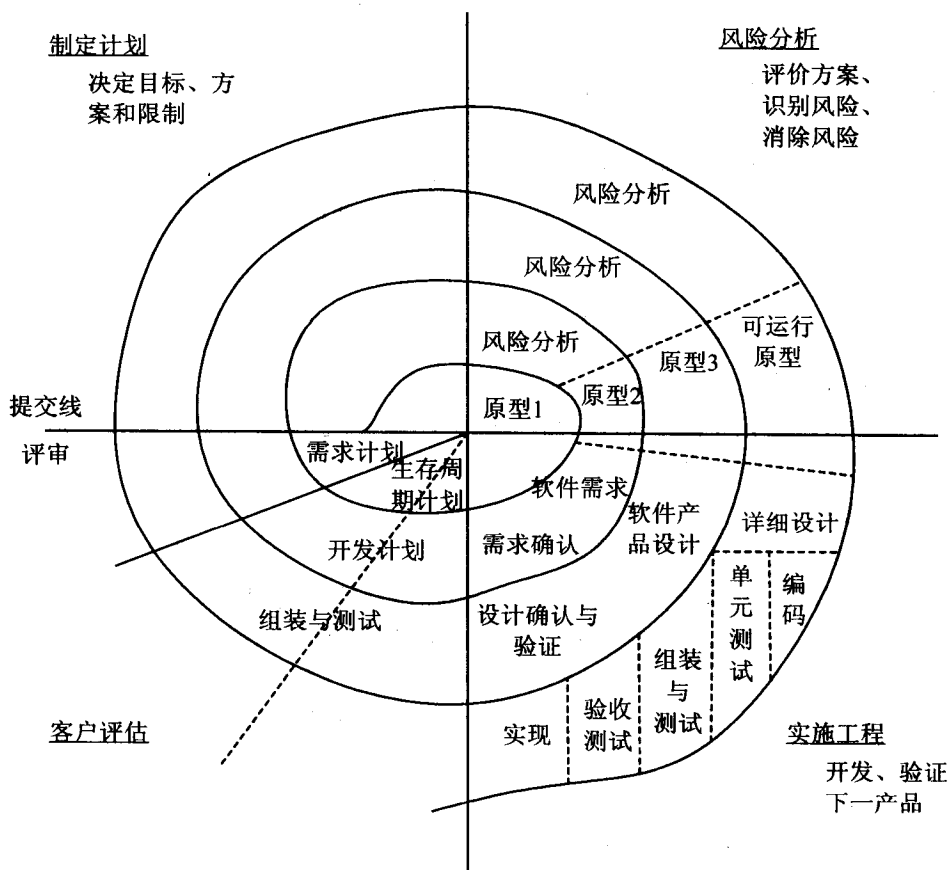


图 1-6 螺旋模型示意图