



普通高等教育“十三五”计算机类规划教材

# 软件工程基础 与实例分析

◎ 张剑飞 主编

*Software Engineering Foundation  
and Case Analysis*

第2版



机械工业出版社  
CHINA MACHINE PRESS

普通高等教育“十三五”计算机类规划教材

# 软件工程基础与实例分析

第2版

主 编 张剑飞  
副主编 周 凤 邓春伟  
参 编 石 磊 高 辉  
主 审 刘兴丽

机械工业出版社

本书从实用的角度出发,系统地介绍了软件工程基础知识,包括传统的软件工程和面向对象的软件工程两大部分。在传统的软件工程部分,按照软件生存周期的顺序介绍各个阶段的任务、过程、方法、工具和文档编写规范;在面向对象的软件工程部分,介绍了面向对象的分析与设计方法以及统一建模语言(UML)的相关知识,同时配有开发实例和软件文档模板。

本书理论与实践相结合,内容循序渐进、深入浅出、通俗易懂、侧重应用。

本书可作为高等院校计算机、通信工程、电子信息工程、自动化等相关专业软件工程课程的教材,还可供软件工程师、软件项目管理者 and 应用软件开发人员参考。

本书配有免费电子课件,欢迎选用本书作为教材的教师登录 [www.cmpedu.com](http://www.cmpedu.com) 注册下载。

### 图书在版编目(CIP)数据

软件工程基础与实例分析/张剑飞主编. —2版. —北京:机械工业出版社, 2018.11

普通高等教育“十三五”计算机类规划教材

ISBN 978-7-111-61079-3

I. ①软… II. ①张… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2018)第224051号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑:路乙达 责任编辑:路乙达 刘丽敏

责任校对:李杉 封面设计:张静

责任印制:张博

三河市宏达印刷有限公司印刷

2019年1月第2版第1次印刷

184mm×260mm·15.75印张·378千字

标准书号:ISBN 978-7-111-61079-3

定价:39.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

服务咨询热线:010-88379833

读者购书热线:010-88379649

封面无防伪标均为盗版

网络服务

机工官网:[www.cmpbook.com](http://www.cmpbook.com)

机工官博:[weibo.com/cmp1952](http://weibo.com/cmp1952)

教育服务网:[www.cmpedu.com](http://www.cmpedu.com)

金书网:[www.golden-book.com](http://www.golden-book.com)

# 前 言

软件工程学是指导软件生产、维护的一门工程科学，从20世纪60年代起迅速发展，现已成为计算机科学中一个重要分支，它的研究范围非常广泛，包括技术、方法、工具和管理等许多方面。

本书从实用角度出发，系统地介绍了软件工程基础知识。在传统的软件工程部分，按照软件生存周期的顺序，介绍了各个阶段的任务、过程、方法、工具和文档编写规范。在面向对象的软件工程部分，介绍了面向对象的分析与设计方法以及统一建模语言（UML）的相关知识及实例。

本书尽量用实例来解释概念，用案例来演绎方法和原理，并选择典型的软件工程开发实例进行剖析，使读者能够在牢固掌握理论知识的同时，思考并尝试解决实际问题。

本书文字通俗易懂、概念清晰、深入浅出、实例丰富、实用性强，可作为高等学校计算机、通信工程、电子信息工程、自动化等相关专业软件工程课程的教材，还可供软件工程师、软件项目管理者 and 软件开发人员参考。

本书的第1~3章由绥化学院周凤编写，第4~7章由哈尔滨石油学院邓春伟编写，第8~9章由中南林业科技大学石磊编写，第10~12、14~15章由黑龙江科技大学张剑飞编写，第13章由黑龙江科技大学高辉编写。黑龙江科技大学刘兴丽担任本书主审。

在本书编写过程中，参考了大量的相关资料，同时得到了各方面有关专家的大力支持和帮助，在此一并感谢。由于时间仓促，水平有限，书中难免有不足之处，敬请读者不吝赐教。

编 者

# 目 录

## 前言

<b>第 1 章 软件工程概述</b> .....	1
1.1 软件 .....	1
1.1.1 软件的概念及特点 .....	1
1.1.2 软件分类 .....	2
1.1.3 软件危机的原因及解决途径 .....	3
1.2 软件工程概念 .....	4
1.2.1 软件工程的定义和内容 .....	4
1.2.2 软件工程的基本原理 .....	4
1.3 软件生存周期 .....	5
1.4 常用软件开发过程模型 .....	6
1.4.1 瀑布模型 .....	6
1.4.2 快速原型模型 .....	7
1.4.3 螺旋模型 .....	8
1.4.4 喷泉模型 .....	9
1.5 软件开发方法简述 .....	9
1.5.1 面向数据流的结构化方法 .....	9
1.5.2 面向数据结构的 Jackson 方法 .....	10
1.5.3 面向对象的方法 .....	10
1.6 软件文档 .....	11
1.6.1 软件文档在软件开发中的地位和作用 .....	11
1.6.2 软件文档的种类及写作要求 .....	12
小结 .....	14
习题 1 .....	14
<b>第 2 章 可行性研究</b> .....	15
2.1 问题定义 .....	15
2.2 可行性研究的任务 .....	15
2.3 可行性研究的过程 .....	16
2.4 可行性研究阶段使用的工具 .....	17
2.4.1 系统流程图 .....	17
2.4.2 数据流图 .....	19
2.4.3 数据字典 .....	24
2.5 成本效益分析 .....	26
2.6 网上招聘系统可行性研究报告 .....	28

小结	31
习题 2	31
<b>第 3 章 需求分析</b>	<b>32</b>
3.1 需求分析的任务	32
3.2 需求分析的过程	33
3.3 需求分析阶段使用的工具	35
3.3.1 实体关系图	35
3.3.2 数据规范化	36
3.3.3 层次框图	37
3.3.4 Warnier 图	37
3.3.5 描述算法的 IPO 图	38
3.4 网上招聘系统需求规格说明书	38
小结	42
习题 3	43
<b>第 4 章 概要设计</b>	<b>44</b>
4.1 软件设计的目标和任务	44
4.1.1 软件设计的目标	44
4.1.2 软件设计的任务	44
4.2 概要设计的过程	45
4.3 软件设计的原理	47
4.3.1 模块化	47
4.3.2 抽象	48
4.3.3 信息隐蔽	49
4.3.4 模块独立	50
4.4 启发规则	52
4.4.1 改进软件结构提高模块独立性	52
4.4.2 模块规模适中	52
4.4.3 适当控制深度、宽度、扇出、扇入	52
4.4.4 模块的作用域应该在控制域之内	53
4.4.5 力争降低模块接口的复杂程度	53
4.4.6 设计单入口单出口的模块	53
4.4.7 模块功能可预测	53
4.5 概要设计阶段使用的工具	53
4.5.1 层次图	53
4.5.2 HIPO 图	54
4.5.3 结构图	55
4.5.4 程序系统结构图	56
4.6 结构化设计方法	56
4.6.1 基本概念	56
4.6.2 系统结构图中的模块	57
4.6.3 结构化设计过程	58
4.6.4 变换分析	58
4.6.5 事务分析	61

4.6.6 混合结构分析 .....	62
4.7 网上招聘系统概要设计说明书 .....	62
小结 .....	66
习题4 .....	66
<b>第5章 详细设计</b> .....	<b>67</b>
5.1 详细设计的过程 .....	67
5.1.1 详细设计的基本任务 .....	67
5.1.2 详细设计方法 .....	68
5.2 详细设计阶段使用的工具 .....	68
5.2.1 程序流程图 .....	68
5.2.2 盒图 .....	70
5.2.3 问题分析图 .....	70
5.2.4 判定表与判定树 .....	71
5.2.5 过程设计语言 .....	72
5.3 面向数据结构的设计方法 .....	75
5.3.1 改进的 Jackson 图 .....	76
5.3.2 Jackson 方法 .....	76
5.4 网上招聘系统详细设计说明书 .....	79
小结 .....	84
习题5 .....	84
<b>第6章 编码</b> .....	<b>85</b>
6.1 选择开发语言 .....	85
6.1.1 程序设计语言分类及特点 .....	85
6.1.2 选择的标准 .....	87
6.2 软件编码的规范 .....	88
6.2.1 程序中的注释 .....	88
6.2.2 数据说明 .....	88
6.2.3 语句结构 .....	89
6.2.4 输入和输出 .....	90
6.3 网上招聘系统编码规范 .....	91
小结 .....	93
习题6 .....	93
<b>第7章 测试</b> .....	<b>94</b>
7.1 测试的目标和原则 .....	94
7.2 测试用例设计 .....	95
7.2.1 黑盒测试 .....	95
7.2.2 白盒测试 .....	98
7.3 测试的步骤 .....	100
7.3.1 单元测试 .....	100
7.3.2 集成测试 .....	102
7.3.3 确认测试 .....	104
7.3.4 系统测试 .....	105

7.4 常用测试工具及特点	105
7.5 软件测试阶段文档写作规范	107
7.5.1 测试文档的类型	107
7.5.2 软件测试过程文档	107
7.6 网上招聘系统客户端测试文档	110
7.6.1 测试计划文档	110
7.6.2 测试设计文档	112
小结	115
习题7	115
<b>第8章 维护</b>	116
8.1 软件维护的概念及特点	116
8.2 软件的可维护性	118
8.3 软件维护的步骤	119
8.4 软件维护过程文档写作规范	121
8.5 用户手册的主要内容及写作要求	121
8.6 网上招聘系统维护文档	122
小结	123
习题8	123
<b>第9章 面向对象的基本概念及UML</b>	124
9.1 传统方法学与面向对象方法比较	124
9.2 面向对象的基本概念	125
9.3 UML概述	127
9.3.1 UML的主要特点	127
9.3.2 UML的应用领域	127
9.4 UML的构成	128
9.5 UML的视图	128
9.6 UML的模型元素	129
9.6.1 事物	129
9.6.2 关系	132
9.7 UML的基本准则和图形表示	132
9.7.1 UML的基本准则	132
9.7.2 UML的图形表示	133
小结	142
习题9	142
<b>第10章 面向对象分析</b>	143
10.1 需求分析与用例建模	143
10.2 建立对象类静态模型	145
10.3 建立对象类动态模型	146
10.3.1 交互模型建模	146
10.3.2 状态模型建模	147
10.4 系统体系结构建模	148
10.4.1 软件系统体系结构模型	148



10.4.2	硬件系统体系结构模型 .....	148
10.4.3	组件图建模 .....	149
10.4.4	配置图建模 .....	150
小结	.....	150
习题 10	.....	150
<b>第 11 章</b>	<b>面向对象设计</b> .....	<b>151</b>
11.1	面向对象设计准则 .....	151
11.2	启发式原则 .....	152
11.3	系统分解 .....	153
11.4	设计问题域子系统 .....	155
11.5	设计人机交互子系统 .....	156
11.6	设计任务管理子系统 .....	158
11.7	设计数据库管理子系统 .....	160
11.8	设计类中的服务 .....	161
11.9	设计关联 .....	162
11.10	设计优化 .....	164
小结	.....	166
习题 11	.....	166
<b>第 12 章</b>	<b>面向对象实例 1——银行系统的分析与设计</b> .....	<b>167</b>
12.1	系统需求 .....	167
12.2	创建用例模型 .....	167
12.2.1	识别参与者 .....	168
12.2.2	识别用例 .....	168
12.2.3	用例的事件流描述 .....	169
12.3	对象类静态模型 .....	177
12.3.1	定义系统对象类 .....	177
12.3.2	定义用户界面类 .....	182
12.3.3	建立类图 .....	184
12.3.4	建立数据库模型 .....	185
12.4	对象类动态模型 .....	186
12.5	系统体系结构建模 .....	194
小结	.....	195
<b>第 13 章</b>	<b>面向对象实例 2——俄罗斯方块分析与设计</b> .....	<b>196</b>
13.1	系统需求 .....	196
13.2	面向对象分析 .....	196
13.2.1	建立功能模型 .....	196
13.2.2	建立动态模型 .....	198
13.2.3	建立对象模型 .....	198
13.2.4	界面设计 .....	200
13.3	面向对象设计 .....	201
13.3.1	系统架构设计 .....	201
13.3.2	模型层设计 .....	202

13.3.3 视图层设计 .....	204
13.3.4 控制层设计 .....	205
小结 .....	206
<b>第14章 传统软件工程实例1——教学管理系统分析与设计</b> .....	<b>207</b>
14.1 可行性研究 .....	207
14.2 系统需求 .....	211
14.3 系统设计 .....	216
14.4 系统实现 .....	221
小结 .....	221
<b>第15章 传统软件工程实例2——高校学生档案管理系统分析与设计</b> .....	<b>222</b>
15.1 系统需求 .....	222
15.2 业务流程分析 .....	222
15.2.1 档案存档管理流程 .....	222
15.2.2 档案转递管理流程 .....	223
15.2.3 档案借阅管理流程 .....	223
15.2.4 档案报表统计流程 .....	225
15.2.5 档案查询管理流程 .....	225
15.2.6 咨询服务管理流程 .....	226
15.2.7 系统管理流程 .....	227
15.3 功能分析 .....	227
15.3.1 角色分析 .....	227
15.3.2 数据流分析 .....	227
15.4 数据分析 .....	232
15.4.1 实体关系分析 .....	232
15.4.2 主要数据流 .....	233
15.5 数据库设计 .....	235
小结 .....	238
<b>参考文献</b> .....	<b>239</b>

# 第 1 章 软件工程概述

## 本章要点

本章主要介绍软件和软件工程的基础知识、比较常用的软件开发模型、典型软件开发方法以及软件文档写作等方面内容。

## 教学目标

了解软件的特点、软件的分类、软件危机产生的原因及解决途径，掌握软件工程的基本原理、软件生存周期的意义以及软件生存周期各阶段的基本任务，掌握瀑布模型、快速原型模型、螺旋模型、喷泉模型等常用的软件开发模型和面向数据流、面向数据结构、面向对象的软件开发方法，认识软件文档的作用、种类和写作要求。

## 1.1 软件

### 1.1.1 软件的概念及特点

#### 1. 软件的概念

“软件”这一名词在 20 世纪 60 年代初从国外引进，当时人们无法说清它的具体含义，也无法解释英文单词“software”，于是有人把它翻译成“软件”或“软制品”，现在应该统一称其为软件。早期，人们认为软件就是源程序。随着人们对软件及其特性的更深层的研究，认为软件不仅包括程序，还应包含其他相关内容。目前，对软件通俗的解释为：

软件 = 程序 + 数据 + 文档

其中，程序是按照事先设计的功能和性能要求执行的指令序列；数据是程序运行的基础和操作的对象；文档是有关程序开发、维护和使用的各种图文材料。

#### 2. 软件的特点

- 1) 软件是一种抽象的逻辑实体。人们无法看到其具体形态，只能通过观察、分析、思考、判断等方式去了解它的特性功能。
- 2) 软件是一种通过人们智力活动，把知识与技术转化为信息的一种产品，是在研制、开发中被创造出来的。
- 3) 软件需要维护。主要是因为软件的生存周期中，为了使它能够适应硬件、软件环境的变化以及用户新的要求，必须进行多次修改（维护）。
- 4) 软件的开发和运行受到计算机硬件、操作系统的限制。
- 5) 软件开发至今尚未摆脱手工开发方式。很多软件仍然是“定制”的，这使得软件的

开发效率受到很大限制。

6) 软件的开发是一个复杂的过程。

7) 软件的成本较高。软件开发需要投入大量的、高强度的脑力劳动, 成本较高。

## 1.1.2 软件的分类

为了便于人们根据不同的应用要求选择相应的软件, 也鉴于不同类型的工程对象, 对其进行开发和维护有着不同的要求和处理方法, 因此从不同角度出发对软件进行分类, 更加符合实际情况。

### 1. 基于软件功能的划分

#### (1) 系统软件

系统软件是与计算机硬件紧密配合以使计算机的硬件部分与相关软件及数据协调、高效工作的软件, 例如操作系统、数据库管理系统、设备驱动程序以及通信处理程序等。

#### (2) 支持软件

支持软件是协助用户开发软件的工具性软件, 包括帮助程序人员开发软件产品的工具和帮助管理人员控制开发进程的工具。

#### (3) 应用软件

应用软件是在特定领域内开发、为特定目的服务的一类软件, 其中商业数据处理软件占有很大比例, 另外还有工程与科学计算软件、计算机辅助设计/计算机辅助制造 (CAD/CAM) 软件、系统仿真软件、智能产品嵌入软件 (如汽车油耗系统、仪表盘数字显示、刹车系统) 以及人工智能软件 (如专家系统、模式识别) 等。此外, 事务管理、办公自动化、中文信息处理、计算机辅助教学 (CAI) 等方面的软件也在迅速发展之中。

### 2. 基于软件规模的划分

按软件开发所需要的人力、时间以及完成的源程序行数, 可划分为 6 种不同规模的软件, 即微型软件、小型软件、中型软件、大型软件、甚大型软件和极大型软件, 见表 1-1。

表 1-1 软件规模分类

类别	参加人员数	研制期限	产品规模 (源程序行数)
微型	1	1~4 周	500
小型	1	1~6 月	1000~2000
中型	2~5	1~2 年	5000~50 000
大型	5~20	2~3 年	50 000~100 000
甚大型	100~1000	4~5 年	1000 000 以上
极大型	2000~5000	5~10 年	10 000 000 以内

### 3. 基于软件工作方式的划分

#### (1) 实时处理软件

实时处理软件是在事件或数据产生时, 立即处理并回馈信号, 控制需要监测和控制的过程的软件, 主要包括数据采集、分析、输出 3 部分。其特点是对外界变化的反映及处理有严格的时间限定。

## (2) 分时软件

分时软件允许多个用户同时使用计算机。系统把处理机时间轮流分配给联机用户，但用户的感受是只有自己在使用计算机。

## (3) 交互式软件

交互式软件实现人机通信。这类软件接收用户给出的信息，但在时间上没有严格的限定。这种工作方式给予用户很大的灵活度。

## (4) 批处理软件

批处理软件是把一组输入作业或一批数据以成批处理的方式一次运行，按顺序逐个处理完的软件。

### 1.1.3 软件危机的原因及解决途径

软件危机指的是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。其实，这些问题并不仅仅是那些不能正常运行的软件才具有的毛病，几乎所有的软件都不同程度地存在或多或少的问题。1968年北大西洋公约组织（NATO）的计算机科学家在联邦德国召开的国际学术会议上第一次提出了“软件危机”（Software Crisis）这个名词。

概括来说，软件危机包含两方面问题：一方面是如何开发软件，以满足不断增长、日趋复杂的需求；另一方面是如何维护数量不断膨胀的软件产品。

#### 1. 产生软件危机的原因

软件危机产生的原因：一方面与软件本身的特点有关；另一方面与软件开发和维护的方法不正确有关。其根本原因，主要表现在以下方面。

##### (1) 忽视软件开发前期的需求分析

经验告诉我们，对用户要求没有准确完整的认识就匆忙着手编写程序，编码阶段开始得越早，完成整个工程的时间反而越长。

##### (2) 没有统一的、规范的方法论的指导

在开发过程中文件资料不齐全，忽视人与人之间的交流，缺乏有力的方法论的指导，加剧了软件产品的个性化，以致产生疏漏和错误。

##### (3) 忽视软件文档，造成开发效率低下

软件文档的编制在软件开发工作中占有突出的地位和相当的工作量。高效率、高质量地开发、管理和维护文档对于软件开发人员、软件维护人员以及用户都至关重要。

##### (4) 忽视测试阶段的工作，提交用户的软件质量差

事实上，对于任何软件来讲，错误是不可避免的。为了尽量减少提交给用户软件中的错误，就需要在测试阶段找出软件中存在的错误，并及时加以修改。

##### (5) 轻视软件的维护

在一个软件漫长的维护期中，必须注意改正软件使用中发现的每一个潜在的错误，以满足用户不断增长的需要。

#### 2. 解决软件危机的途径

为了更好地解决软件危机，既需要技术措施，还要有必要的组织管理措施。先进的开发方法和工具可以保证软件的质量。严密组织、严格管理和各类人员工作的协调一致都是必不可少的因素。经过不断实践和总结，得出一个结论：按工程化的原则和方法组织软件开发工

作是有效的，是摆脱软件危机的一个主要出路。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的学科。

## 1.2 软件工程概念

### 1.2.1 软件工程的定义和内容

#### 1. 软件工程的定义

1968年10月，北大西洋公约组织（NATO）科学委员会在联邦德国开会讨论软件可靠性与软件危机的问题，会议上，Fritz Bauer首次提出了“软件工程”的概念。后来，人们曾经多次给出了有关软件工程的定义。1993年IEEE为软件工程的定义是：软件工程是将系统化的、规范化的、可度量的途径应用于软件的开发、运行和维护的过程，即将工程化应用于软件的方法的研究。

#### 2. 软件 Engineering 的内容

软件工程是一种层次化的技术，如图1-1所示。和其他工程方法一样，软件工程是以质量为关注焦点，以相关的现代化管理为理念。

软件工程的基础是过程层。软件工程的过程是为获得软件产品，在软件工具支持下由软件人员完成的一系列软件工程的活动，贯穿于软件开发的各个环节。它定义了方法使用的顺序、要求交付的文档资料，是软件开发各个阶段完成的标志。

软件工程的方法为软件开发提供了“如何做”的技术，通常包括以下内容：与项目有关的计算和各种估算方法、需求分析、设计方法、编码、测试和维护等。

软件工程的工具为软件工程方法提供了自动的或半自动的软件支撑环境，辅助软件开发任务的完成。现有的工具覆盖了需求分析、系统建模、代码生成、程序调试和软件测试等多个方面，形成了集成化的软件工程开发环境计算机辅助软件工程（Computer-Aided Software Engineering, CASE），以便提高软件开发效率和软件质量，降低开发成本。

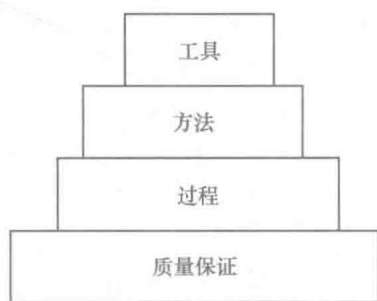


图 1-1 软件工程层次图

### 1.2.2 软件工程的基本原理

自从1968年提出“软件工程”这一术语以来，研究软件工程的专家学者们陆续提出了100多条关于软件工程的准则。美国著名的软件工程专家Boehm综合这些专家的意见，并总结了多年的开发软件的经验，于1983年提出了软件工程的7条基本原理。

- 1) 用分阶段的生存周期计划严格管理开发过程。
- 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) 明确地规定开发小组的责任和产品标准。

- 6) 开发小组的人员应少而精。
- 7) 承认不断改进软件工程实践的必要性。

上述7条基本原理,相互独立,缺一不可。实践过程中,可以对这些原理进行细化和再生,灵活运用这些原理指导软件开发。

### 1.3 软件生存周期

概括地说,软件生存周期就是从提出软件产品开始,直到该软件产品被淘汰的全过程。一般来说,软件生存周期包括计划、开发、运行三个时期,每一时期又可分为若干更小的阶段。本节介绍的软件生存周期分为软件系统的可行性研究、需求分析、软件设计(概要设计和详细设计)、编码、软件测试、运行与维护。它们之间的关系如图1-2所示。

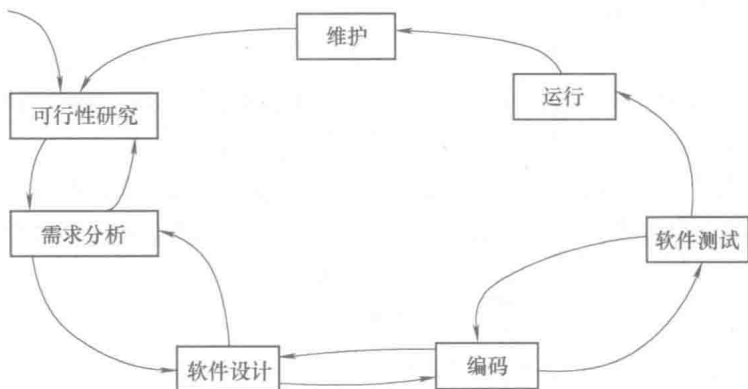


图 1-2 软件的生存周期

软件生存周期每个阶段的基本任务如下。

#### (1) 可行性研究

可行性研究阶段主要确定软件的开发目标及其可行性,给出其在功能、性能、可靠性以及接口等方面的要求。可行性研究由系统分析员和用户合作探讨,并且对可利用资源(计算机硬件、软件、人力等)、成本、可取得效益、开发的进度做出估量,制订任务实施计划,连同可行性报告提交给管理部门。

#### (2) 需求分析

需求分析主要解决待开发软件要“做什么”的问题,包括需求的获取、分析、规格说明、变更、验证、管理等一系列工作。软件开发人员与用户共同讨论决定,哪些需求是可以满足的,并且加以确切地描述,然后编写出软件需求说明书或系统功能说明书和初步的系统用户手册,提交给管理部门。

#### (3) 软件设计

软件设计主要解决待开发软件“怎么做”的问题,软件设计通常可分为概要设计和详细设计。概要设计的任务是设计软件系统的体系结构,也就是确定程序由哪些模块组成以及模块间的关系。详细设计是具体设计每个模块,确定实现模块功能所需要的算法和数据结构。

#### (4) 编码

编码是软件开发过程中的生产步骤。具体就是将软件转化为计算机代码,对其功能用某

一种特定的计算机语言进行描述。编写出的程序要求具有结构性，并且易读，重要的是要与设计要求一致。

#### (5) 软件测试

软件测试的目的是确认软件的质量，一方面是确认软件做了用户所期望的事情，另一方面是确认软件以正确的方式来做这些事情。首先进行单元测试，查找各个模块在内部功能结构上存在的问题，其次进行集成测试，查找模块间联合工作存在的问题，最后进行确认测试、系统测试，决定软件产品质量是否过关，能否交用户使用。

#### (6) 运行与维护

软件产品开发完成投入使用后可能运行若干年。在运行过程中可能因为各方面原因需要进行修改，硬件变更、操作系统升级、平台移植等问题都可能需要对软件进行维护。

## 1.4 常用软件开发过程模型

软件工程是基于问题求解的，要解决问题必须找到求解问题的策略。该策略包括软件的过程、方法、工具三个层次，被称为软件开发过程模型。该模型规定了把生存周期划分成哪些阶段及各个阶段的执行顺序。软件开发过程模型的选择依赖于项目和应用的性质、所采用的方法和工具、需要交付的产品和软件开发的控制。因此，选择开发过程模型对于项目的开发至关重要。

### 1.4.1 瀑布模型

瀑布模型是在1970年由W. Royce最早提出的软件开发模型，如图1-3所示。它将软件生存周期的各项活动规定为依固定顺序连接的若干阶段工作，这些工作之间的衔接关系是从上到下、不可逆转的，如同瀑布一样，因此称为瀑布模型。

每项开发活动均应具有下述特征：

- 1) 以上一项活动产生的工作对象作为输入。
- 2) 利用这一输入，实施本项活动应完成的内容。
- 3) 给出该项活动的工作结果，作为输出传给下一项活动。
- 4) 对实施该项活动的工作结果进行评审。若其工作得到确认，则继续进行下一项活动，否则返回前项，甚至更前项的活动进行返工。

瀑布模型自提出以来，一直是一种广泛采用的开发模型，但在长期的实践中，人们发现这种模型有如下一

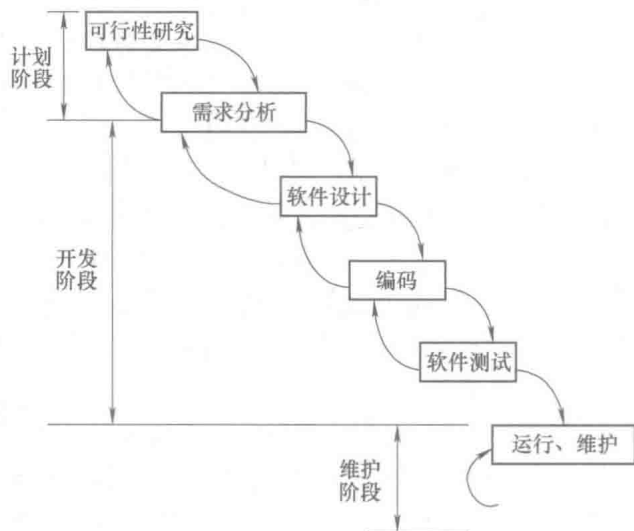


图 1-3 瀑布模型



些缺点:

1) 在项目开始阶段,开发人员和用户对需求的描述常常是不全面的。如果需求阶段未发现这些问题,就会影响到后面各阶段的工作。

2) 瀑布模型中各阶段所做的工作都是文档说明,而对一般用户来说,很难全面理解文字描述背后的软件产品。当用户在提出一些意见时,加大了系统修改难度。

3) 影响整个软件开发进度。例如在开发过程中,事先选择的技术或需求迅速发生变化,需要返回到前面的某个阶段,对前面的一系列内容进行修改。

总的来说,瀑布模型是一种应付需求变化能力较弱的开发模型。因此,很多在该模型基础上开发出来的软件产品不能真正满足用户需求。

## 1.4.2 快速原型模型

快速原型模型的第一步是建造一个快速原型,实现客户或未来的用户与系统之间的交互,用户或客户可以通过对原型的评价,进一步细化待开发软件的需求,由此通过逐步调整原型而进一步满足客户的要求,开发人员也可以确定客户的真正需求是什么;第二步则在第一步的基础上开发客户满意的软件产品。快速原型模型如图1-4所示。

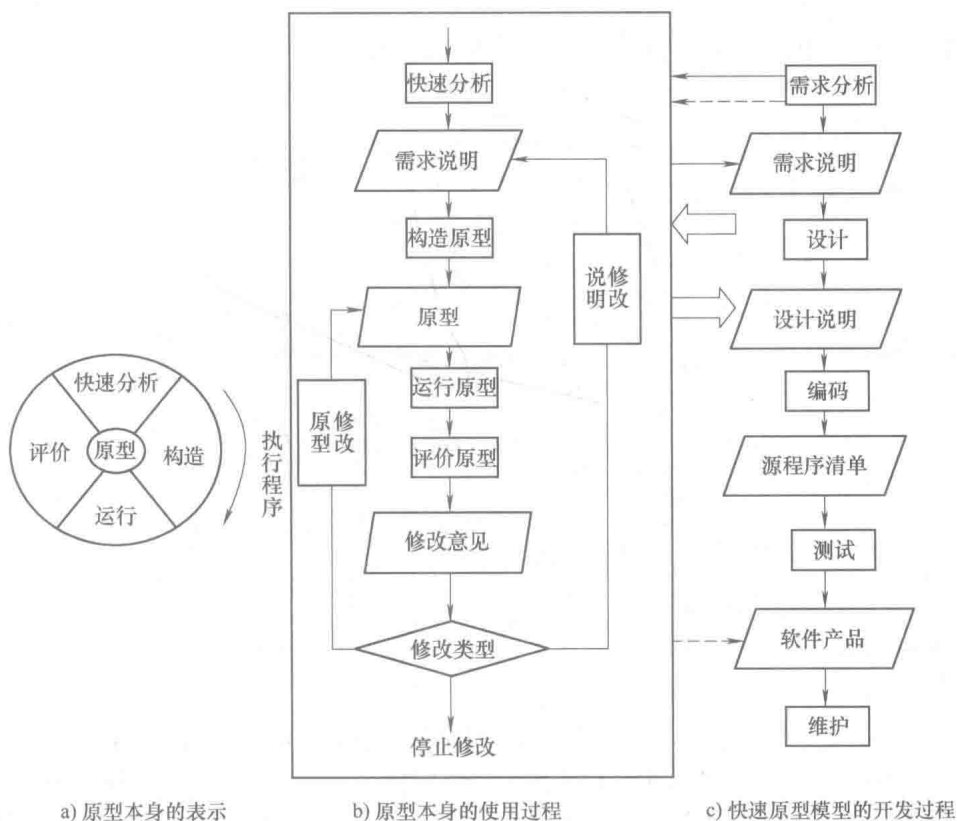


图 1-4 快速原型模型

显然,快速原型模型可以克服瀑布模型的缺点,减少由于软件需求不明确所带来的开发风险,事实证明具有显著的效果。