21世纪高等学校规划教材

Textbook Series of 21st Century

软件工程

宋 雨 赵文清 编 著

中国电力出版社
http://jc.cepp.com.cn

21世纪高等学校规划教材
Textbook Series of 21st Century

软件工程

编 著 宋 雨 赵文清
主 审 边小凡

中国电力出版社
http://jc.cepp.com.cn

内 容 提 要

本书为21世纪高等学校规划教材。

全书分三篇：第一篇基础篇是软件工程的基本理论和技术，介绍了计算机系统的开发过程、软件及软件工程的概念，讲解了软件计划、软件需求分析、软件设计、软件编码、软件测试、软件维护和软件复用的具体内容。第二篇中级篇是软件工程专门技术，包括面向对象的软件工程、软件质量保证、软件的技术度量、软件工程经济学、软件开发工具与环境，可作为进一步选学内容，也可作为研究生课程的教学内容。第三篇高级篇可作为高级软件工程教学内容，是软件工程的深入研究领域，包括软件工程管理、软件过程管理、基于构件的软件工程（CBSE）、客户/服务器（C/S）软件工程以及柔性软件开发技术。三篇独立，可根据教学需要选择或组合。

本书主要作为高等院校软件工程、高级软件工程课程的本科和研究生教材，也可作为函授相关专业的教材，同时可供软件工程技术人员和管理人员阅读和参考。

图书在版编目（CIP）数据

软件工程/宋雨，赵文清编著. —北京：中国电力出版社，2007

21世纪高等学校规划教材

ISBN 978-7-5083-5173-5

Ⅰ.软… Ⅱ.①宋… ②赵… Ⅲ.软件工程—高等学校—教材 Ⅳ.TP311.5

中国版本图书馆CIP数据核字（2007）第004566号

中国电力出版社出版、发行

（北京三里河路6号 100044 http://jc.cepp.com.cn）

汇鑫印务有限公司印刷

各地新华书店经售

*

2007年2月第一版 2007年2月北京第一次印刷

787毫米×1092毫米 16开本 24印张 586千字

印数0001—3000册 定价**36.00**元

前 言

由于环境污染日益严重，使我们有了环保意识并重视环境工程。当前，软件系统已无处不在，软件作为信息社会人类离不开的工具正发挥越来越大的作用，然而带来的问题就像环境污染一样越来越严重，因而应像全民树立环保意识一样，我们应该具有软件工程意识、树立软件工程思想、重视软件工程教育。

在20世纪90年代初，中国计算机学会教育委员会及全国高等学校计算机教育研究会就把《软件工程》列为计算机学科中的专业必修课和主干课。之后，中国计算机科学与技术学科教程研究组又进一步将其列为核心课程，学时数为54+32，并建议在执行时实践学时数"安排为标定数的2倍"，此处即为64学时，可见《软件工程》课程的地位和作用已十分重要。

为了适应软件工程教育日益重要的形势，根据中国电力出版社规划、依据软件工程课程体系及学科内容我们制定了软件工程编写大纲。在依据软件工程编写大纲编写本书时我们虽然参考了其他相关文献，但又不同于其他同类书籍，力求体现自己的特色、反映自己的水平。本书从结构和内容上也不同于其他软件工程教材，既有成熟的内容，也有发展的内容，还有相当一部分内容是作者多年从事软件工程科研和教学的结晶。

全书分3篇，较全面地介绍了软件工程基本原理、方法和技术。

第一篇基础篇是软件工程的基本理论和技术，共8章，可作为软件工程课程选用，也可供初次接触该学科的读者作为入门学习用书。第一章从计算机系统的总体开发过程出发，介绍了软件、软件工程的概念以及软件生存期的11种模型；第二章讲解了在软件计划阶段的主要工作内容，包括确定软件范围、确定软件开发中的资源、估算软件成本（介绍了具体模型和技术）和安排软件开发进度；第三章是软件需求分析，介绍了结构化需求分析方法、原型化需求分析方法以及面向对象方法；第四章是软件设计，介绍了软件设计的有关概念，讲解了结构化设计方法、面向数据结构的设计方法以及面向对象的设计方法，简要介绍了其他新的设计方法，本章还讲解了软件设计工具及程序复杂性的度量等相关内容；第五章介绍了程序设计语言的分类、编码风格以及面向对象的编程语言；第六章讲解了软件测试的步骤、策略、方法，面向对象的测试以及软件可靠性；第七章介绍了软件维护的概念，讲解了软件的可维护性含义、提高可维护性的方法以及软件再工程方法；第八章是软件复用，介绍了软件复用概念、领域工程、可复用构件的建造和复用以及面向对象的软件复用技术。

第二篇中级篇是软件工程专门技术，共5章。第九章是面向对象的软件工程，系统讲解了面向对象的基本概念、面向对象建模及统一建模语言UML、面向对象的软件需求分析及设计过程和方法，给出了2个实例；第十章讲述了软件质量保证的有关内容，介绍了ISO9000质量标准，给出了软件质量预测模型及应用；第十一章是软件的技术度量，包括软件分析模型的度量、软件设计模型的度量、源代码度量、软件测试的度量、软件维护活动的度量以及面向对象的度量；第十二章系统介绍和讨论了Barry W. Boehm提出的COCOMO

模型；第十三章介绍了软件开发工具与环境。这5章基本是独立的，既可作为软件工程课程的进一步选学内容，也可作为研究生软件工程课程的教学内容。

第三篇高级篇也选了5章内容，是软件工程的深入研究领域。第十四章是软件工程管理，讨论了软件工程文化、现代人件、软件风险分析与管理、软件配置管理、软件工程标准化以及软件的知识产权保护等课题，本章还介绍了金尊和先生所提出的软件工程36计；第十五章讲解了由SEI提出的软件能力成熟度模型（CMM）以及个体软件过程（PSP）；第十六章讲述了基于构件的软件工程（CBSE），包括CBSE基本概念、可复用构件的分类与查询、基于构件的软件重用成熟度以及构件技术的应用；第十七章介绍了目前较流行的客户/服务器（C/S）软件工程的内容，包括C/S系统结构、C/S系统分析、C/S系统设计及C/S系统的测试；第十八章介绍了柔性软件开发技术。

每一章后都有小结，便于读者复习、归纳和总结。三篇独立，可根据教学需要选择或组合，如软件工程课程可选基础篇或基础篇＋中级篇、高级软件工程课程可选高级篇或中级篇＋高级篇。

本书第一篇和第三篇由华北电力大学计算机系宋雨教授编写，第二篇由赵文清副教授编写，王晓辉老师编写了第十二章和第十三章，研究生李小青编写了第十八章、王莉莉编写了第十五章。电子系阎力芳老师帮助整理了大量书稿，研究生王莉莉、金美玉、高伟、李小青、吴双和田华协助录入书稿。宋雨教授对全书进行了统一修改与定稿。

在中国电力出版社领导和有关编辑同志的关怀下，经过作者一年半的努力，本书终于完成了。在本书编写过程中，华北电力大学计算机学院博士生导师朱永利教授，计算机系王保义教授、程晓荣教授、郑顾平副教授，中国电力出版社编辑给与了大力支持；河北大学计算中心边小凡教授认真审阅了书稿，提出了宝贵意见。在此一并表示衷心的感谢！

本书参考了大量的文献，均已在书后列出，并在此向所有参考文献的作者表示衷心的感谢！

因为有了矛，所以有了对付它的盾。矛在发展，越来越锋利，试图所向披靡，盾也在发展，越来越结实，试图阻挡一切矛的进攻。针对软件危机，出现了对付它的软件工程。然而，就像矛和盾在此消彼长一样，软件危机并未完全摆脱，软件工程还在不断进步和发展，因而我们的书也只能反映当前“盾”的情况。

尽管我们进行了认真的检查，但肯定还会有许多不足之处，恳请同行专家及读者提出宝贵意见，在此向您表示感谢！

作　者

2006年12月

目 录

第二篇　中　级　篇

第三篇　高　级　篇

第一篇 基础篇

第一章 概述

计算机是现代人都熟悉的一个词汇，也有人把它叫做电脑，还有人说“网络就是计算机”。从宏观上讲，一个计算机系统应该由3部分组成，即硬件（Hardware）、软件（Software）和人件（Peopleware），它们互相联系、互相依存，硬件和软件是“经济基础”，人件是“上层建筑”。软件的角色很特殊，有时起到“经济基础”的作用，有时又起到“上层建筑”的作用。这3部分都非常重要，缺一不可。

第一节 计算机系统的开发过程

计算机系统的开发过程是指从提出项目开始，经过分析、论证、设计、实施到交付使用的全过程，其开发流程如图1-1所示。

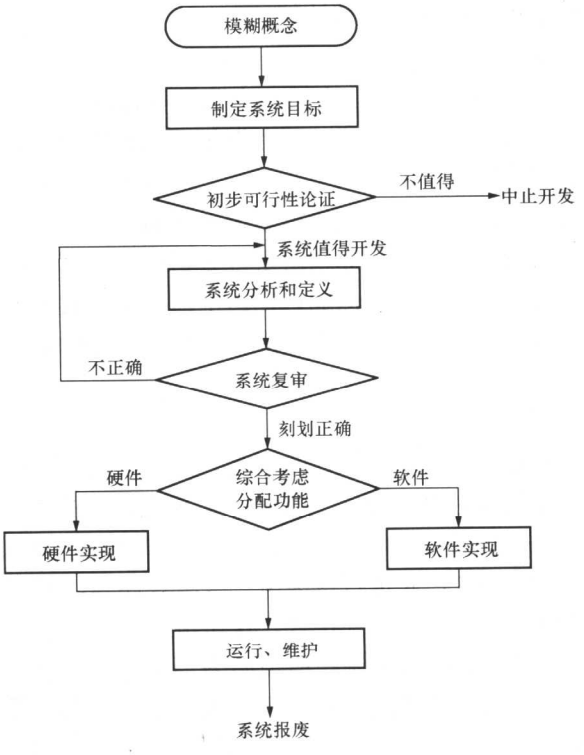

图1-1 计算机系统开发流程

从图1-1可以看出，最初对未来的计算机系统我们只能有一个模糊的概念，根据这个概念来制定系统的初始目标，主要应该明确建立计算机系统的目标是什么，适应于哪些用户

对象，期望的性能价格比是多少，开发周期多长，总的开发费用是多少等等。这些问题要尽量定量描述，对于无法定量描述的内容，要仔细研究和推敲，用严密精确的概念和术语去描述。

初步可行性论证主要从3个方面讨论该计算机系统是否值得开发：一是开发该计算机系统的先决条件是否具备，二是成功的可能性有多大，三是考虑是否有社会效益和经济效益、技术是否先进。完成了初步可行性论证工作后，要写出初步可行性论证报告。

接着进行的工作是系统分析和定义，这一步是关键性的一步，要从整体上寻求一个最优解，主要工作有4个方面。

1. 调查研究

通过查阅资料、上网、进行市场调查、走访用户等手段，明确要解决的问题和相关内容。

2. 确定系统的功能和性能

这是对系统总目标的一个求解过程，可以用由底向上的归纳法或自顶向下的演绎法。前者适合于应用系统的功能确定，可从应用对象的现状出发，分析实际业务活动，明确问题处理过程及处理过程中的每一操作，直至确定出系统总目标；后者适用于未知系统，一般从总目标出发逐步分解，直到功能不可分解为止。

主要考虑的系统性能可包括信息的内容和长度、信息的输入/输出频率和系统的响应时间、存储能力、可靠性、可维护性、兼容性等。

3. 进行初步的方案设计

根据以上的工作可拟定3～5个候选方案，主要是考虑选择什么样的系统模式以及系统各元素如何分担功能，是采用工作站模式、服务器模式还是网络计算机或其他模式，可根据应用对象的特征和要求来确定。

把系统要实现的功能如何合理地分配给各系统元素是关系到能否获得最优性价比的一个重要因素，对于系统的功能要求，有些必须由硬件去实现，但有些既可由硬件也可由软件去实现。其参考原则是：首先要保证速度性能；在这个前提下，对于大批量生产的系统，尽可能用软件去实现功能，因为这样成本会大幅度降低；对于可靠性要求高的系统，功能则尽可能由硬件去实现。

4. 可行性分析

其任务就是从候选方案中选择一个最好的，此时要对候选方案进行评判和比较，主要考虑的因素包括方案是否能很容易地市场化、技术上是否先进（技术风险性如何）、制造成本如何、人件是否满足和合适、工具和环境如何以及专利方面的有关问题。在进行可行性分析时，可采用模糊综合评判的理论，先进行单因素一级综合评判，再根据权重定义进行二级综合评判。可行性分析结束后要提交可行性报告，这份报告是系统进一步发展的基本依据。

系统分析和定义结束后，要撰写系统规格说明书，这是一份很重要的文档，主要可包括引言、对每一功能的描述、实现各功能的系统元素的描述、各种约束（政治、经济、管理、环境、资源、进度、投资等）、成本预算及进度6个方面。除此以外，还要编写系统开发计划，这也是一份很重要的文档，可包括工程进程安排、各种人员需要情况、各种资源需要情况及经费估算等。

上述工作全部结束后要进行系统复审，具体做法是由开发人员和需求人员对系统规格说

明书中的各项逐项进行讨论，进行正确性检查，目的是要保证系统得到正确的刻划。

接下来的工作是分配功能，由硬件或软件具体实现，用硬件实现可用硬件工程的方法。硬件工程可分为三个阶段：计划和规格说明，设计和样机实现，生产、分配和现场服务。用软件工程的方法去开发软件是本书研究的内容。

图 1-1 仅是一个粗的流程图，每一项展开后都需要很多工作，即使在系统投入运行后，工作也没有结束，只有当系统报废后才能算结束了。

第二节 软件及其分类

软件一词源于程序，是 20 世纪 60 年代出现的，是一个发展的概念，是和硬件发展紧密相关的，不同时期有不同的解释。由于软件的功能、模块、开发方式在不断发生变化，因而对它的解释也在发展。目前，软件包括的内容和范围很广，一般认为执行指令的计算机程序以及与之相关的文档、数据、影视资料、方法、规则等都可算作软件。

软件一词具有 3 层含义：一是个体含义，指计算机系统中的程序、文档和数据；二是整体含义，指在特定计算机系统中所有上述个体含义下的软件的总称；三是学科含义，指在研究、开发、维护以及使用前述含义下的软件所涉及的理论、方法、技术所构成的学科。

软件的种类很多，随着复杂程度的增加，软件的分类也在发展变化，界限越来越不明显，按软件的作用，可分为以下 4 类。

1. 系统软件

系统软件由计算机生产厂家配置，服务于其他软件。如操作系统、汇编程序、编译程序、数据库管理系统、计算机通信及网络软件等。没有这些软件，计算机很难发挥功能，甚至无法工作。

2. 应用软件

应用软件是指在系统软件的基础上，为解决特定领域的问题而开发的软件。这类软件很多，如各种事务类软件，监视、分析和控制正在发生的现实世界时间的各种实时软件，各类科学和工程软件，用于工业、民用或军事上的各种功能的嵌入式软件，个人计算机软件，基于 web 的软件，人工智能软件等。

3. 工具软件

工具软件用于辅助和支持开发及维护应用软件，以提高软件开发质量和生产率。如需求分析工具、设计工具、编码工具、测试工具、维护工具、管理工具等。

4. 可重用软件

可重用软件是指通过可重用构件来开发新的软件，目前，图形用户界面一般是使用可重用构件创建的，这些构件涉及图形窗口、下拉菜单和各种交互机制。建造界面所需要的数据结构和处理细节包含在一个由界面构件所组成的可重用构件库中。

若按功能对软件进行分类，则可划分为系统软件、支撑软件和应用软件 3 类；若按规模进行划分，可将软件分为 6 类，如表 1-1 所示；若按软件工作方式进行划分，则可分为实时处理软件、分时软件、交互式软件和批处理软件 4 类；若按软件服务对象的范围进行划分，则可分为项目软件和产品软件 2 类；也可按其他方式对软件进行划分，如按软件的使用频度进行划分、按软件失效的影响进行划分等。

表 1-1 按规模对软件的分类

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	0.5 千行
小型	1	1~6 月	1~2 千行
中型	2~5	1~2 年	5~50 千行
大型	5~20	2~3 年	50~100 千行
甚大型	100~1000	4~5 年	1 兆行（=1000 千行）
极大型	2000~5000	5~10 年	1~10 兆行

软件产品包括两类，通用软件产品和定制软件产品。它们的区别在于，通用软件产品的描述由开发者自己完成，而定制软件产品的描述通常由客户给出，开发者必须按客户要求进行开发。因而，各类数据库软件、字处理软件、绘图软件、工程管理软件等属于通用软件产品，而特定的业务处理系统、电子设备的控制系统、空中交通管制系统等属于定制软件产品。

第三节 软件工程的由来和发展

20 世纪 40 年代中期，美国宾夕法尼亚大学约翰·莫克莱（John Mauchly）和普雷斯帕·埃克特（J·Presper Eckert）研制出世界上第一台计算机 ENIAC。1947 年，数学家冯·诺依曼（J·Von Neumann）针对 ENIAC 提出了 EDVAC（Electronic Discrete Variable Automatic Computer）方案，在该方案中首次提出了“存储程序”的概念。1949 年，由英国皇家科学院院士莫里斯·威尔克斯制造的世界上第一台实现冯·诺依曼“存储程序”思想的计算机 EDSAC（Electronic Delay Storage Automatic Calculator）问世。从冯·诺依曼提出存储程序概念至今 60 年的时间中，软件发生了很大变化，其发展可用表 1-2 概括。

表 1-2 软 件 的 发 展

阶段划分	阶段名称	时间	对软件的解释	软件开发方法	决定软件质量的因素	硬件特征	软件特征及技术
第一阶段	程序设计阶段	20 世纪 50 年代	程序	个人	个人程序设计技术	价格昂贵，存储容量小，工作可靠性差	软件安全不受重视
第二阶段	程序系统阶段	20 世纪 60 年代	程序+说明	“软件作坊”式的小组	小组技术水平	降价，速度、存储容量及工作可靠性明显提高	多用户、实时、数据库、软件产品，软件技术的发展不能满足需要，出现了软件危机
第三阶段	软件工程阶段	20 世纪 70 年代、80 年代中期	程序、文档、数据	开发小组及大、中、小型软件开发机构	管理水平	向超高速、大容量、微型化发展	分布式系统、嵌入式软件出现；软件危机并未摆脱

续表

阶段划分	阶段名称	时间	对软件的解释	软件开发方法	决定软件质量的因素	硬件特征	软件特征及技术
第四阶段	现代软件工程阶段	20世纪80年代末至现在	程序、文档、数据、Web页、方法、规则	中间件技术、网络技术、构件技术、管理技术的应用使跨平台、跨区域开发成为可能，超大型软件工厂出现	管理水平	网络化发展迅速，大容量新型高速存储器问世，运算速度达每秒几万亿次至万万亿次的巨型计算机不断出现	强大的桌面系统、面向对象技术、专家系统、人工神经网络、并行计算、客户/服务器环境。软件危机仍然存在

从表1-2可以看出，在20世纪60年代由于软件的发展不能满足需求出现了软件危机，所谓软件危机是指在计算机软件开发和维护过程中所遇到的一系列问题，这些问题几乎存在于所有的软件中。

软件危机的主要表现为以下7个方面：

1. 软件开发成本和进度估计往往很不准确

实际成本比估计成本高出一个数量级，实际进度比预期进度拖延几个月，甚至几年的现象经常发生。

2. 软件系统不符合用户的实际需要

开发人员和用户之间的信息交流往往很不充分，在没有确切了解问题的情况下，或对用户需求还很模糊的情况下就开始编写程序，导致开发出的软件产品不符合用户的实际需要。

3. 软件经常出故障

一个著名的例子是美国IBM公司的OS/360，这是第一个功能较强的多道程序操作系统，参加这项研制工作的有IBM公司美国国内11个单位、欧洲6个单位，共计1000多人，耗费了5000人·年的工作量，但结果不尽人意，它的每个版本都是在前一版本的基础上找出1000个程序错误而修正的结果，项目总负责人费雷德里克·布鲁克斯（Frederick P·Brooks Jr.）在他所著的《人月神话》（The Mythical Man-Month）一书中生动地描述了研制中遇到的困难和问题，提出了很多人都深省的观点。

4. 软件常常不可维护

很多程序中的错误很难改正，让一个软件系统适应新的硬件环境，或在原有程序中增加一些新功能都非常困难。

5. 缺乏文档资料

在软件开发过程中应该有一套与之同步生成的文档资料，它们起着开发过程“里程碑”的作用，若事后再补或缺乏文档资料都会给开发及维护带来困难。

6. 软件成本在计算机总成本中所占比例逐年上升，维护费用增长迅速

目前，软件费用已经成为计算机系统的主要费用，其中维护费用占去了大部分。

7. 软件生产不能满足对软件的需求

虽然软件开发人员及软件生产率都在不断增长，但由于计算机的普及程度越来越高，应用领域越来越广，使得人们对软件的需求越来越大，导致生产与需求的差距不断加大。

软件危机的表现远不止这些，为了摆脱日益严重的危机，1968年在德国召开的NATO

学术会议上与会学者们首次提出了软件工程，这是软件开发走向工程化的标志，从此软件技术的发展及软件工程的研究有了长足的进步，并起着越来越大的作用。目前，软件工程已经成为一门大的学科，在它下面又出现了很多新的学科，需要研究的未知领域和课题还很多，但软件危机并未完全摆脱。

第四节 软件的生命周期

软件和任何有生命或无生命的事务一样，有它的生命周期（或称为生存周期），一个人的生命周期是从孕育、出生，经过乳幼期、少年期、青年期、壮年期、老年期直到死亡，各个阶段的投入、目标、活动显然都是不一样的。软件也是类似的，一个软件的生命周期(Software Life Cycle)，是指从制定软件计划开始，经过软件需求分析、软件设计、编码、软件测试、运行和维护直到软件报废的全过程。

软件的生命周期各阶段是从宏观上对软件的划分，当然也可以更粗地划分为计划、开发和维护3个阶段。与人的生命周期不同的是，软件的生命周期与开发模型有关，不同的开发模型可能对应着不同的生命周期。生命周期不同，则软件开发阶段的划分、评审次数及基线标准都有所不同。

1. 软件的计划阶段

确定待开发软件系统的工作范围，给出它的功能、性能、约束、接口和可靠性等方面的要求；预测开发该软件所需的人力、物力资源；对软件成本进行估算；对进度进行安排。软件计划阶段工作结束后，要交付项目开发计划，连同可行性研究报告一起交管理部门审查。

2. 需求分析和定义阶段

理解用户的要求，对待开发软件的需求进行分析，给出详细定义，编写软件需求说明书(SRS)及初步的用户手册，提交管理部门评审，SRS是以后各阶段工作的基础。

3. 软件设计阶段

以SRS为基础，把确定的软件需求转换成相应的体系结构，对每个模块的具体工作进行描述，编写设计说明书，提交有关部门评审。

4. 编码阶段

将软件设计转换成计算机程序代码，编写测试用例，对模块进行测试。

5. 软件测试阶段

进行系统的单元测试、整体测试、有效性测试和系统测试，检验开发的软件是否完成了所要求的功能，整体性能、结构如何，是否满足用户的要求以及是否能与其他系统元素协同工作等。

6. 运行和维护阶段

软件系统在投入运行后，还会逐步暴露出错误，另外，用户可能也会不断提出一些新的要求。因此，系统需要不断地排错、修改、扩充功能，这些工作都是维护。维护工作量是很大的，一般占软件70%以上的工作量。有人做过统计，如果一次修改5~10个程序语句，则成功的可能性是50%，若一次修改40~50个语句，则成功的可能性只有20%，因此，维护也是很困难的。

软件生命周期的每个阶段都要产生一定规格的文件移交给下一阶段，使下一阶段在所提

供的文件基础上继续开展工作，在软件开发过程中主要应编制 14 种文件，它们是可行性研究报告、项目开发计划、软件需求说明书、数据要求说明书、概要设计说明书、详细设计说明书、数据库设计说明书、用户手册、操作手册、模块开发卷宗、测试计划、测试分析报告、开发进度月报以及项目开发总结报告，在编制时应遵循有关标准和规范。

第五节 软件生存期模型

类似于其他工程项目中安排各道工序，为了反映软件生命周期各种活动如何组织、各个阶段如何衔接，需要用软件生存期模型以图示的方式直观地表达出来。软件生存期模型是反映软件开发过程、开发活动和开发任务的结构框架。

一、瀑布模型

这是提出最早的，目前使用最广泛的软件开发模型，图 1 - 2 是瀑布模型的基本形式。从图中可以看出，整个开发活动如同瀑布流水，逐级下落，互相衔接，每个阶段的工作都以上一阶段的工作结果为基础，同时为下一阶段的工作提供依据。

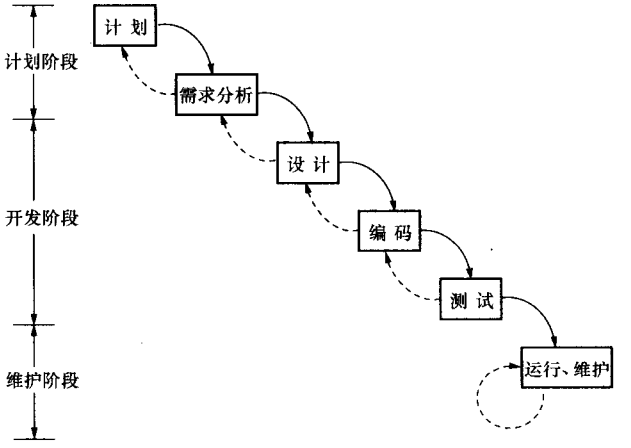

图 1 - 2 软件生存期的瀑布模型

在图 1 - 2 的基础上又提出了其他的变种，如前文所述，软件在投入运行后还要进行维护，维护活动也要经历软件生命周期的各个阶段，如何把维护和开发一并来表达，就构成了软件生存周期循环，如图 1 - 3 所示。

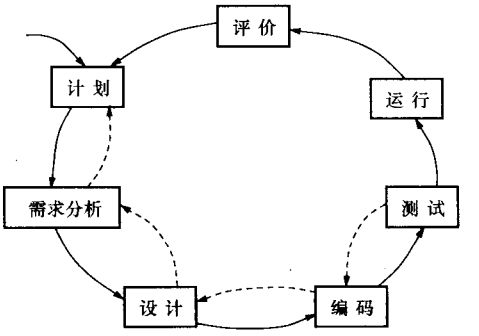

图 1 - 3 软件生存周期循环

由于软件在投入运行后要经常维护，可以把维护看作是软件的二次开发，为把开发活动和维护活动区别开来，又提出了 b 型软件生存周期模型，如图 1 - 4 所示。

瀑布模型支持结构化开发，为软件开发和维护提供了较为有效的管理模式，它对控制软件开发复杂度、制定开发计划、进行成本预算、组织阶段评审和文档控制等各项软件工程活动都较为有效，对保证软件质量具有较好的作

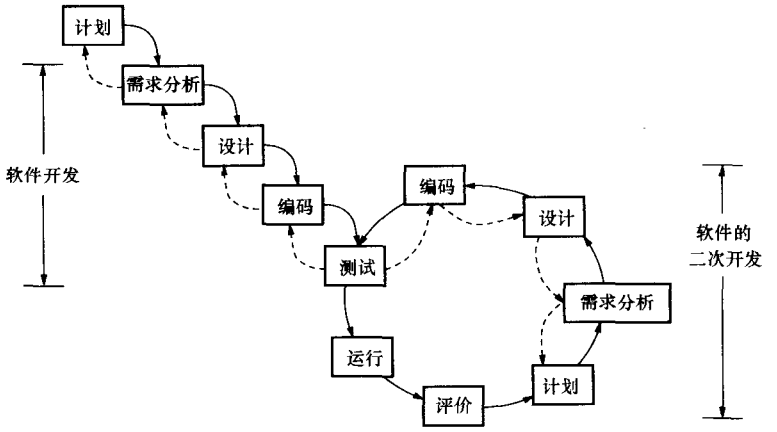

图 1-4 b 型软件生存周期模型

用。但它的突出缺点是缺乏灵活性，无法应付软件需求不明确、不准确的问题，特别是，由于各阶段工作次序固定，使前期工作中造成的差错越到后期影响越大，带来的损失也越大，而要想纠正它们所花费的代价也越高，而这又是不可避免的。

二、演化模型

演化模型也叫原型开发模型，主要是针对事先不能完整定义需求的软件开发。开发人员根据用户的需求，先开发核心系统（即原型），让用户试用，用户提出改进、精化及增强系统能力的需求。开发人员根据用户的反馈意见，实施开发的迭代过程。每一迭代过程均由需求、设计、编码、测试、集成等阶段组成，为整个系统增加一个可定义的、可管理的子集，如图 1-5 所示。如果在一次迭代中，有的需求不能满足用户的要求，可在下一次迭代中进行修正。

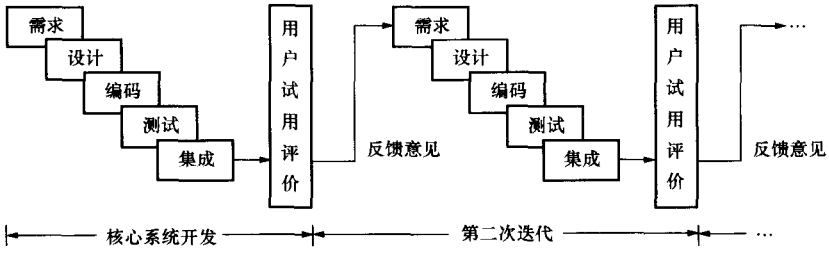

图 1-5 演化模型

演化模型也有多种形式，如丢弃型、样品型、渐增式演化型等，它的特点是突出一个“快”字，用户可以很快看到未来系统的“样品”，但它也存在很多问题。一方面，构造原型时很难考虑软件的整体质量和系统以后的可维护性问题。另一方面，为了尽快造出原型，开发人员常常使用不适当的开发环境、编程语言和效率不高的算法，而这些有可能成为系统集成的一部分。使用演化模型的关键在于开始时原则的确定，用户和开发人员双方必须达成一致，即建立原型主要是作为定义需求的一种机制，实际软件设计的重点在于如何提高软件质量和软件的可维护性。

三、螺旋模型

螺旋模型是将瀑布模型和演化模型相结合，加入了二者所忽略的风险分析所建立的一种