普通高等教育“十一五”国家级规划教材

新版

21世纪

高职高专系列教材

软件工程

第2版

◎王宜贵　主编

◎万建成　眭碧霞　主审

提供电子教案增值服务

机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十一五”国家级规划教材

21世纪高职高专系列教材

软 件 工 程

第2版

王宜贵 主编
李晓方 刘 伟 尹 辉 参编
万建成 眭碧霞 主审

机械工业出版社

本书系统地介绍了软件工程的基本概念、软件开发方法、软件开发工具和软件项目管理。其中，第1章概要介绍软件工程；第2～7章按生命周期模型详细介绍制定计划、需求分析、软件设计、程序编码、软件检验和软件维护各个阶段的相关概念和工作内容，重点介绍了结构化方法和面向对象方法；第8章介绍软件开发工具和环境；第9章介绍软件项目管理；第10章是一个文档实例最后介绍了统一建模语言（UML）。

本书可供高职高专计算机专业及其相关专业师生使用。

图书在版编目（CIP）数据

软件工程/王宜贵主编. —2版. —北京：机械工业出版社，2008.1
普通高等教育“十一五”国家级规划教材.21世纪高职高专系列教材
ISBN 978-7-111-10789-7

Ⅰ.软… Ⅱ.王… Ⅲ.软件工程-高等学校：技术学校-教材
Ⅳ.TP311.5

中国版本图书馆CIP数据核字（2007）第136238号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
责任编辑：王 颖 祝 伟 责任校对：申春香
封面设计：雷明顿 责任印制：李 妍
北京中兴印刷有限公司印刷
2008年1月第2版第1次印刷
184mm×260mm · 15.5印张 · 379千字
17 001—22 000册
标准书号：ISBN 978-7-111-10789-7
定价：23.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379739
封面无防伪标均为盗版

21世纪高职高专计算机专业系列教材
编委会成员名单

出 版 说 明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位，促进学生技能的培养，以及教材内容要紧密结合生产实际，并注意及时跟踪先进技术的发展等指导精神，机械工业出版社组织全国近60所高等职业院校的骨干教师对在2001年出版的“面向21世纪高职高专系列教材”进行了全面的修订和增补，并更名为“21世纪高职高专系列教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师，针对相关专业的课程设置，融合教学中的实践经验，同时吸收高等职业教育改革的成果而编写完成的，具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中，本系列教材获得了较高的评价，并有多个品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中，除了保持原有特色外，针对课程的不同性质采取了不同的优化措施。其中，核心基础课的教材在保持扎实的理论基础的同时，增加实训和习题；实践性较强的课程强调理论与实训紧密结合；涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时，根据实际教学的需要对部分课程进行了整合。

归纳起来，本系列教材具有以下特点：

（1）围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。

（2）合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度，强调专业技术应用能力的训练，适当增加实训环节。

（3）符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁，多用图表来表达信息；增加相关技术在生产中的应用实例，引导学生主动学习。

（4）教材内容紧随技术和经济的发展而更新，及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念，并积极支持新专业的教材建设。

（5）注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合，提高教学服务水平，为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快，加之我们的水平和经验有限，因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息，以利于我们今后不断提高教材的出版质量，为广大师生提供更多、更适用的教材。

机械工业出版社

前　言

高等职业教育的目标是培养具有一定基础理论、专业知识水平和较强实践操作技能的技术应用型人才。本书内容强调针对性，理论上强调必需、够用，技术方法上强调实用、管用，编写本教材的指导思想就是基于以上这些要求。

软件工程是一门指导软件开发和维护的工程学科，研究如何用工程化的方式有效地管理软件开发、以较低的成本按期开发出高质量软件。本书较全面地介绍了软件工程的基本概念、软件开发方法、软件开发工具和软件项目管理，详细介绍了结构化方法和面向对象方法的实施。本书主要内容包括软件工程概述、制定计划、需求分析、软件设计、程序编码、软件检验、软件维护、软件开发工具和环境、软件项目管理、软件开发文档实例和 UML。

本书自第 1 版出版以来，得到了广大读者的厚爱，许多读者提出了宝贵意见，编者在此表示衷心感谢。为了适应软件工程技术的发展，在总结近几年来教学实践经验并结合当前高等职业教育特点的基础上，对第 1 版作了较大的修改和补充。改版后，本书具有如下特点：

1）内容更加合理。着重从实用角度讲述软件工程基本概念、原理和方法，既注重结构的完整性，又注重针对高职高专学生的实用性。

2）结构更加严谨。将结构化方法与面向对象方法结合在一起讨论需求分析和设计方法，保持软件工程概念的完整性、一贯性。

3）增加了面向对象方法的内容。针对当前软件开发的实际需要，增加了面向对象方法在软件开发各个阶段的应用。

4）采用较为完整的实例。在篇幅允许的情况下以较完整的实例介绍了软件开发方法，使读者对开发方法更容易理解和应用。

5）强调文档编写。将软件工程各阶段文档编写的规范置于相关章节之后，并给出了一个较为完整的文档实例，便于读者在开发软件时参考。

6）加强了课后练习。在每章后都配有精选的适量习题，便于读者对内容的学习和理解。

本书由王宜贵、李晓方、刘伟、尹辉共同编写，全书由王宜贵统稿。万建成教授和眭碧霞教授对本书做了全面、仔细的审定，提出了宝贵的修改意见，在此表示衷心的感谢。

在本书编写过程中，参阅了大量的文献资料，也得到许多老师的大力支持和帮助，在此一并表示衷心感谢。由于时间仓促，编者水平有限，书中缺点和错误之处难免，恳请读者批评指正。

为了配合本书教学，机械工业出版社为读者提供了电子教案，读者可从 www. cmpedu. com 上下载。

编　者

目　录

第1章　软件工程概述

1.1　软件的概念

1.1.1　软件的含义

软件在计算机系统中的比重越来越大，而且这种趋势还在增加。人们感到传统的软件生产方式已不再适应发展的需要，因此提出把工程学的基本原理和方法引进到软件生产中，把软件生产分成几个阶段，每个阶段都有严格管理和质量检验，研制了软件设计和生产的方法和工具，并用书面文件作为共同遵循的依据。这时软件的含义就成了文档加程序。

软件是计算机系统中与硬件相互依存的部分，包括程序及其相关文档。程序是计算机任务的处理对象和处理规则的描述；文档是为了理解程序所需的阐述性资料。

1.1.2　软件的特点

软件与硬件相比主要有以下不同：

（1）表现形式不同

硬件是有形有色、看得见摸得着的，而软件是无形无色、看不见摸不着的。软件正确与否，是好是坏，要到程序在机器上运行才能知道，这给设计、生产和管理带来许多困难，生产成本也相当高。

（2）生产方式不同

软件是人的智力的高度发挥，不同于传统意义上的硬件制造，它没有明显的制造过程，因此对软件的质量控制必须立足于软件开发方面。

（3）维护不同

硬件是有损耗的，它会产生磨损和老化使故障率增加甚至损坏，解决的办法是换上一个相同的备件。而软件不存在磨损和老化问题，但存在退化问题。因为在软件的生存期中，它一直处于改变（维护）状态，随着某些缺陷的改变，很可能带来一些新的缺陷，使软件的故障率增加。软件不像硬件一样存在备用件，它的维护要比硬件复杂得多。

1.1.3　软件的分类

（1）按软件的功能分类

① 系统软件　能与计算机硬件紧密配合，使计算机系统的各个部件、相关的软件和数据协调、高效地工作的软件。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 支撑软件　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以及帮助管理人员控制开发进程的工具。

③ 应用软件　在特定领域内开发，为特定目的服务的软件。

（2）按软件的规模分类

按开发软件所需人力、时间以及完成的源程序行数，可分为6种不同规模的软件，如表1-1所示。

表1-1　软件规模的分类

类　别	参加人员数/人	研制期限	源程序长度/行
微型	1	1～4周	500
小型	1	1～6月	$(1\sim2)\times10^3$
中型	2～5	1～2年	$(5\sim50)\times10^3$
大型	5～20	2～3年	$(50\sim100)\times10^3$
超大型	100～1000	4～5年	1×10^6
极大型	2000～5000	5～10年	$(1\sim10)\times10^6$

（3）按软件的工作方式分类

① 实时处理软件　在时间或数据产生时对其立即处理，并及时反馈信号以控制需要检测的过程的软件。实时处理软件主要包括数据采集、分析、输出三个部分。

② 分时软件　允许多个联机用户同时使用计算机的软件。

③ 交互式软件　实现人机通信的软件。

④ 批处理软件　把一组输入作业或一批数据以成批处理的方式一次运行，按顺序逐个处理的软件。

（4）按软件服务对象的范围分类

① 项目软件　也称定制软件，是为某个特定客户或少数客户开发的软件，如军用防空指挥系统软件、卫星控制系统软件等。

② 产品软件　直接提供给市场或是为众多用户服务的软件，如文字处理软件、财务处理软件、人事管理软件等。

1.2　软件工程的产生和概念

1.2.1　软件危机

由于新的电子元器件的出现，计算机硬件的功能和质量在不断地提高，价格却在大幅度地下降。同硬件投资相比，软件投资比例上升极快，但软件开发的生产率提高缓慢。

软件开发本质上是一个“思考”的过程。若没有统一的标准可遵循，开发人员可以按各自的爱好和习惯进行工作，以手工的方式进行，就会使软件开发工作很难管理，管理人员事前很难估计项目所需的经费和时间，技术人员在项目完成之前也难以预料系统是否成功。

人们在开发大型软件系统时遇到过很多困难。有的系统最终彻底失败了；有的系统虽然完成了但比原定计划迟了好几年，且经费大大超出了预算；有的系统未能完满地符合用户当初的期望；有的系统无法进行修改维护。失败的系统往往无可挽回，除非从头再来，但由于时间和经费的限制，这又是不可能的。

IBM公司在开发OS/306系统时这些矛盾非常突出。这一项目的工作量是5000人年，有

时1000人投入开发工作，共约100万条指令，结果却非常糟糕。该项目负责人Brooks沉痛地说："…像巨兽在泥潭中作垂死挣扎，挣扎得越猛，泥浆就沾得越多，最后没有一只野兽能逃脱淹没在泥潭中的命运，…程序设计就像是这样一个泥潭，…一批批程序员在泥潭中挣扎，…没人料到问题竟会这样棘手，…"。

人们发现研制软件系统需要投入大量的人力和物力，但系统的质量却难以保证，也就是说，软件开发所需的高成本同产品的低质量之间存在尖锐的矛盾。把软件开发和维护过程中遇到的一系列严重问题统称为软件危机，它有以下表现：

1）不能正确地估计软件开发成本和进度，致使实际开发成本高出预算很多，完成开发的期限一再拖延。

2）在开发的初期，软件需求不够明确，开发过程中又未能和用户及时交换意见，致使开发出的软件不能满足用户的需求，甚至无法使用。

3）开发过程没有统一、公认的方法和规范进行指导，且设计和实现过程的资料很不完整，使得软件很难维护。

4）未做好测试阶段的工作，提交给用户的软件质量差，在运行中暴露出大量问题。

5）开发效率低，远远满足不了社会发展的需要。

概括地说，软件危机是指计算机软件的开发和维护过程中遇到的一系列严重的问题，这些问题体现在如何开发软件以满足用户日益增长的需求和如何维护已有的数量庞大的软件两个方面。要解决软件危机需做好以下几个方面的工作：

1）加强软件开发过程的管理，做到组织有序，各类人员协同配合，共同保证工程项目完成，避免软件开发过程中个人单干的现象。

2）推广使用开发软件的成功技术与方法，并且不断探索更好的技术和方法；消除一些错误概念和做法。

3）开发和使用好的软件工具，支持软件开发的全过程，即建立软件工程支持环境。

总之，要从技术、管理两个方面入手，引入"软件工程"的概念。

1.2.2 软件工程的定义和内容

1. 软件工程的定义

1968年在北大西洋公约组织的学术会议上，软件工作者第一次提出"软件工程"这个词，讨论建立并使用正确的工程方法开发出成本低、可靠性高并能高效运转的软件，从而解决或缓解软件危机。

软件工程是一门指导软件开发和维护的工程学科，是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列完善的工程化原则。它应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则、方法来生产软件，以达到提高质量、降低成本的目的。

1993年IEEE为软件工程下的定义是：软件工程是将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护过程，即将工程化应用于软件中的方法的研究。

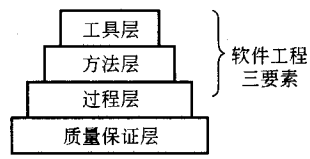

图1-1 软件工程层次图

2. 软件工程的内容

软件工程是一种层次化的技术，如图1-1所示。

与任何工程方法一样，软件工程以质量为关注焦点，全面质量管理及相关的现代管理理念是软件工程强有力的根基。

一般将方法、工具和过程称为软件工程的三要素。软件工程的基础是过程层。软件工程过程是为获得软件产品，在软件工具支持下由软件开发人员采用某种方法完成的一系列软件工程活动。过程层将方法和工具结合在一起，它定义了一组关键过程区域的框架，定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理以及软件开发各个阶段完成的里程碑。方法层提供了软件开发的各种方法，软件工程方法是指导研制软件的某种标准规范。工具层为软件工程方法提供了自动的或半自动的软件支撑环境。软件工程工具是指软件开发、维护和分析中使用的程序系统。使用软件工程工具可以有效地改善软件开发过程，提高软件开发效率，降低开发成本。

软件工程学是一门涉及内容广泛的学科，所依据的理论基础极为丰富，包括数学、计算机科学、经济学、工程学、管理学和心理学等，其研究的内容包括软件开发技术和软件管理技术。软件开发技术又包括软件开发方法、软件开发工具和软件工程环境，软件管理技术包括项目估算、项目计划、人员组织、进度控制、配置管理和软件度量等。

1.3 软件工程的目标和原理

1.3.1 软件工程的基本目标

从技术和管理上采取多项措施以后，组织实施软件工程项目的最终目的是保证项目成功，即达到以下基本目标：

- 付出较低的开发成本。
- 达到预期的软件功能。
- 取得较好的软件性能。
- 使软件易于移植。
- 需要较低的维护费用。
- 能按时完成开发工作，及时交付使用。

在实际项目开发中，使上述目标都达到理想的程度往往非常困难，而且上述目标很可能是相互冲突的。图 1-2 表明了软件工程目标之间存在的相互关系。有些目标是互补的，如高可靠性与易于维护之间、低开发成本与按时交付之间。有些目标是互斥的，如低开发成本与高可靠性之间、高性能与高可靠性之间。

以上几个目标是判断软件开发方法或管理方法优劣的衡量尺度。在一种新的开发方法提出之后，人们关心的是它对满足哪些目标比现有的方法更为有利。实际上，实施软件项目开发的过程就是在以上目标的冲突中取得一定程度平衡的过程。

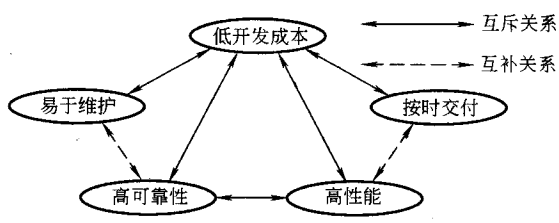

图 1-2 软件工程目标之间的关系

1.3.2 软件工程的基本原理

著名的软件工程专家 B. W. Boehm 综合了有关专家和学者的意见并总结了 TRW 公司多年的软件开发经验，于 1983 年提出了软件工程的 7 条基本原理。这 7 条原理被认为是保证软件质量和开发效率的原理的最小集合，又是相互独立、缺一不可、相对完备的最小集合。

（1）用分阶段的生命周期计划严格管理

把软件生命周期划分为若干个阶段，相应地制定可行的计划，严格按照计划对软件开发与维护进行管理。据统计，在不成功的软件项目中有一半左右是由于计划不周全造成的。

（2）坚持进行阶段评审

坚持在每个阶段结束前进行严格的评审，就可以尽早发现错误。错误发现越晚，纠正代价越大。

（3）实施严格的产品控制

在软件生命周期中改变需求是难免的，要依靠科学的产品控制技术来顺应用户提出的需求改变。为了保持软件各个配置成分的一致性，必须实施严格的产品控制。其中主要是实施基准配置管理。其基本思想是：对软件的各项修改建议，都必须按规定进行严格的评审和控制，获得批准后才能实施修改，同时要保证其他有关的各阶段的文档和代码作相应变动。

（4）采用现代程序设计技术

采用先进的程序设计技术既可以提高软件开发与维护的效率，又可以提高软件的质量。

（5）结果应能清楚地审查

为了更好地进行评价和管理软件开发小组的工作，应根据软件开发的总目标和完成期限，尽量明确地规定出软件开发小组的责任和产品标准，从而能清楚地审查所得到的结果。

（6）开发小组的成员应该少而精

不同素质的软件开发人员的开发效率可能相差几倍甚至几十倍，开发的软件的质量也相差很多。开发小组的人数不宜过多，因为随着人数的增多，各成员之间交流的通信开销会急剧增加，从而降低开发效率，并因误解等原因增加出错的概率。

（7）承认不断改进软件工程实践的必要性

遵循前 6 条基本原理就能较好地实现软件的工程化生产，但是仅有前 6 条原理并不能保证软件开发与维护的过程能赶上时代前进的步伐和技术的进步，因此要积极主动地采纳新的软件技术，不断总结经验。

1.4 软件生命周期和开发模型

1.4.1 软件生命周期

软件生命周期（Software Life Cycle）是指一个计算机软件从功能确定、设计，到开发成功投入使用，并在使用中不断地修改、增补和完善，直到停止该软件使用的全过程。它包括制定计划（可行性与计划研究）、需求分析、设计、程序编码（实现）、测试及运行维护 6 个阶段。

（1）制定计划

确定待开发软件系统的总目标，给出它的功能、性能、可靠性以及接口等方面的总体要求；完成该项软件任务的可行性研究，探讨解决问题的可能方案；制定完成开发任务的实施计划。

（2）需求分析

由系统分析人员对待开发软件提出的需求进行分析并给出详细的定义，确定该软件的各项功能、性能需求和设计约束，编写软件需求说明书及初步的用户手册。

（3）软件设计

系统设计人员和程序设计人员把已确定的软件需求转换成相应的软件设计，包括该软件的结构、模块的划分、功能的分配以及处理流程。若系统比较复杂，设计阶段应分解成概要设计阶段和详细设计阶段两个步骤，并编写概要设计说明书、详细设计说明书和测试计划初稿。

（4）程序编码

把软件设计的结果转换成计算机可以接受的程序代码，要完成源程序的编码、编译（或汇编）和排错调试得到无语法错误的程序清单。

（5）软件测试

在设计测试用例的基础上，按照软件测试计划对软件进行全面测试，给出软件测试报告，并排除检查出的错误，确保软件质量。

（6）运行和维护

软件在运行使用中要进行适当维护，包括纠正软件的错误、使软件能够适应环境的变化及扩充软件的功能。

1.4.2 软件开发模型

软件开发模型（也称为软件过程模型）是从软件项目需求定义开始直至软件经使用后废弃为止，跨越整个生命周期的系统开发、运行和维护所实施的全部过程、活动和任务的结构框架。软件开发模型能清晰、直观地表达软件开发全过程，明确规定要完成的主要活动和任务，用来作为软件项目工作的基础。它确立软件开发和演绎中各阶段的次序限制以及各阶段活动的准则，确立开发过程所遵守的规定和限制，便于各种活动的协调以及各种人员的有效通信。

最早的软件开发模型是 1970 年 W. Royce 提出的瀑布模型，它是软件工程的基础模型。而后随着软件工程学科的发展和软件开发的实践，人们提出了快速原型化模型、增量模型、螺旋模型、喷泉模型等。应该考虑项目和应用的性质、将要使用的方法和工具等，选择其中的一种模型。

1. 瀑布模型

瀑布模型规定了各项软件工程活动，包括制定开发计划、需求分析、设计、编码、测试和运行维护，并且规定了自上而下相互衔接的固定顺序，如同瀑布流水，逐级下落，如图 1-3 所示。每个阶段的工作都以上一个阶段工作的结果为依据，同时又是下一个阶段工作的前提，一个阶段的工作完成后才能开始下一个阶段的工作。

采用瀑布模型进行开发组织时，应指定软件开发规范或开发标准，其中要明确规定各个

开发阶段应交付的产品，这为严格控制软件开发项目的进度、最终按时交付产品以及保证软件产品质量创造了有利条件。

为了保障软件开发的正确性，每一阶段任务完成后，都必须对它的阶段性产品进行评审，确认之后再转入下一个阶段。如评审过程发现错误和疏漏，应该返回到前面的有关阶段修正错误，弥补漏洞，然后再重复前面的工作，直至该阶段的工作通过评审后再进入下一阶段。

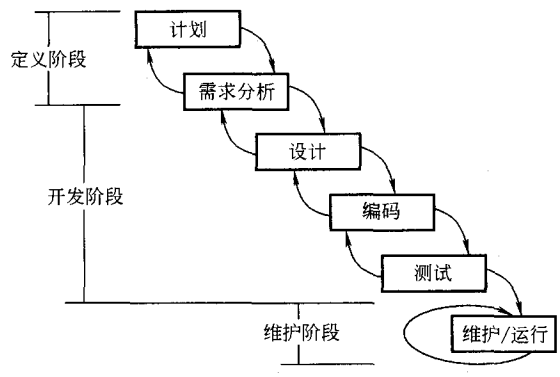

图 1-3　瀑布模型

瀑布模型在支持结构化软件开发、控制软件开发的复杂性、促进软件开发工程化等方面起着显著作用。与此同时，瀑布模型在大量软件开发实践中也逐渐暴露出它的缺点。其中最为突出的缺点是描写缺乏灵活性，无法通过开发活动澄清本来不够确切的软件需求，这些问题可能导致开发出的软件并不是用户真正需要的软件，无疑要进行返工或不得不在维护中纠正需求的偏差，为此必须付出高额的代价，给软件开发带来不必要的损失。并且，随着软件开发项目规模的日益庞大，该模型的不足所引发的问题显得更加严重。为弥补瀑布模型的不足，人们又提出了多种其他模型。

2. 快速原型模型

原型通常是指模拟某种产品的原始模型。在软件开发中，原型是软件的一个早期可运行的版本，它反映最终系统的重要特性。

按照瀑布模型开发软件，分析员要作出准确的需求分析，但往往由于用户不熟悉计算机技术，分析员对用户的业务了解不深，在需求分析阶段的用户需求往往是不完全和不准确的。

快速原型模型的基本思想是软件开发人员根据用户提出的主要需求快速开发一个原型，以便让用户看到软件系统的基本功能和性能，以判断哪些功能是符合需要的，哪些是需要改进的，进一步使需求精确、完全，并将原型改进、完善，如此反复，直到软件开发人员和用户都确认软件系统的需求并达成一致的理解为止。利用原型确定系统需求的过程如图 1-4 所示。

快速原型模型与瀑布模型相比，系统开发人员一开始就针对用户的要求给出一个实实在在的系统原型，然后与用户反复交流修改来确定用户的需求。

按运用的目的和方式不同，原型可分为废弃型、追加型（演化型）。用废弃型原型是先构造一个功能简单而且质量要求不高的模型系统，针对这个模型系统反复进行分析修改，形成比较好的设计思想，系统构造完成后原型被废弃。用追加型原型是先构造一个功能简单而且质量要求不高的模型系统，作为最终系统的核心，然后通过不断地扩充修改，逐步追加新要

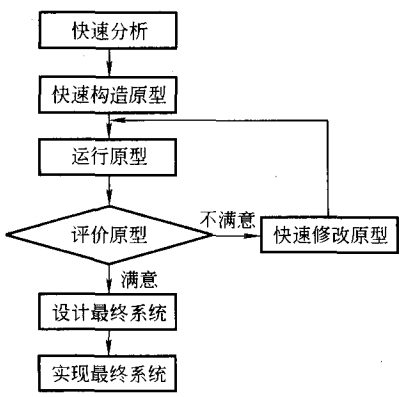

图 1-4　快速原型模型

求，发展成为最终系统。

因为是用较少的开支较快地建立原型，所以只能针对用户和开发人员最为关心的一部分性质，如用户界面、功能或性能。

3. 增量模型

前面介绍的瀑布模型和快速原型模型基本都属于线性顺序的开发模型。原型模型的迭代也只涉及软件开发的某些阶段，整个软件开发过程还是线性顺序的。但软件产品需求经常随着开发过程的向前推进会发生变化，这使得线性顺序的开发模型往往不合实际。

增量模型规定软件的开发过程是一次开发产品的一个部分。第一轮开发的是一个基本产品，也就是说只满足用户一些基本的需求，还有一些其他的需求（可能已经清楚，也可能不清楚）暂不实现。这时可以提交基本产品给用户使用和评价。根据用户的评价结果和提出的修改和补充意见，即可进行下一轮的计划和开发。这一过程不断重复，直到完成最终产品。增量模型如图 1-5 所示。

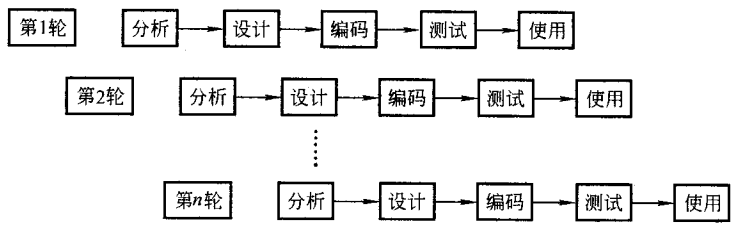

图 1-5　增量模型

增量模型在开发人员不是非常充裕或者规定交付期限很短的情况下十分有用。采用增量模型的优点是用户不需要等到最终产品就可以使用软件的部分功能，而且由于软件产品不是突然一次性提交给用户使用，用户比较容易接受。

采用增量模型的一个难点是后来开发的部分必须能够集成到先前已经开发的产品中而不破坏已开发的功能，这就要求开发的产品必须容易扩充。为了使所开发的产品的各个部分最后能有机地组合起来，应该有一个统一的开放的体系结构设计。

4. 螺旋模型

风险是软件开发不可忽略的潜在的不利因素，它可能在不同程度上损害到软件开发过程和软件产品的质量。要在造成危害之前及时对风险进行识别、分析，采取对策，以消除风险或减小风险损失。实践表明，项目规模越大，问题越复杂，资源成本和进度等因素的不确定性越大，项目的风险也就越大。

螺旋模型是将瀑布模型与增量模型结合起来，不仅体现了两个模型的优点，而且还增加了两个模型都忽略了的风险分析。螺旋模型如图 1-6 所示，它由 4 部分组成：制定计划、风险分析、实施开发、用户评价。

沿螺旋线自内向外每旋转一周，软件开发就前进一个层次，生成一个更为完善的新版本。系统模型越来越完整，直至最后得到一个用户满意的软件版本为止。

螺旋模型不仅保留了瀑布模型中系统地、按阶段逐步地进行软件开发的风格，而且引入了风险分析，使得开发人员和用户对每个演化曾出现的风险有所了解，继而作出应有的反应。如果项目风险较大，又未能及时发现，势必造成重大损失。用户始终参与软件开发，并对阶段性的软件产品提出评审意见，这对保证软件产品的质量十分有利。