21世纪高职高专系列教材

软件工程

中国机械工业教育协会　组编

机械工业出版社
China Machine Press

软件工程

21世纪高职高专系列教材

软 件 工 程

中国机械工业教育协会　组编

主　编　天津大学　周志刚

副主编　天津大学　孙秀钰

南昌大学　林仲达

参　编　包头职业技术学院　张海黎

天津职业技术师范学院　王珍玲

主　审　大连理工大学　刘润彬

机 械 工 业 出 版 社

本书是根据高等职业技术教育理工科计算机类软件工程课教学要求编写的，全书共9章，内容包括绪论、软件定义及计划、软件需求分析、软件设计、软件编码、软件测试、软件运行及维护、软件项目管理技术、面向对象软件工程等。其中既给出了运用传统软件工程学的方法进行软件开发的详细过程与开发实例、软件开发文档的编写及实例、基本的软件项目管理技术，也介绍了最新软件工程方法的基本思想和开发过程。每章后均配有适量复习思考题，用以帮助学生加深对本章内容的消化和理解。附录中还提供了标准软件开发文档和实习题目供读者选用。

本书既可作为高等职业技术院校、高等专科院校、职工大学、业余大学、夜大学、函授大学、成人教育学院等大专层次的理工科计算机类软件工程课程的教材，也可作为广大自学者及软件技术人员的自学丛书。

图书在版编目（CIP）数据

软件工程 / 中国机械工业教育协会组编. —北京：机械工业出版社，2001.9
21世纪高职高专系列教材
ISBN 7-111-08408-X

I. 软… II. 中… 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（2001）第051733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:朱 华 封面设计:姚 毅
责任印制:郭景龙
北京铭成印刷有限公司印刷·新华书店北京发行所发行
2001年8月第1版第1次印刷
787mm×1092mm 1/16 · 11.25印张·275千字
0 001—4 000册
定价:18.00元

21世纪高职高专系列教材编委会名单

编委会主任 中国机械工业教育协会 郝广发

编委会副主任（单位按笔画排）

山东工程学院 仪垂杰
大连理工大学 唐志宏
天津大学 周志刚
甘肃工业大学 路文江
江苏理工大学 杨继昌
成都航空职业技术学院 陈玉华
机械工业出版社 陈瑞藻（常务）
沈阳工业大学 李荣德
河北工业大学 檀润华
武汉船舶职业技术学院 郭江平
金华职业技术学院 余党军

编委会委员（单位按笔画排）

广东白云职业技术学院 谢瀚华
山东省职业技术教育师资培训中心 邹培明
上海电机技术高等专科学校 徐余法
天津中德职业技术学院 李大卫
天津理工学院职业技术学院 沙洪均
日照职业技术学院 李连业
北方交通大学职业技术学院 佟立本
辽宁工学院职业技术学院 李居参
包头职业技术学院 郑 刚
北京科技大学职业技术学院 马德青
北京建设职工大学 常 莲
北京海淀走读大学 成运花
江苏理工大学 吴向阳
合肥联合大学 杨久志
同济大学 孙 章
机械工业出版社 李超群 余茂祚（常务）
沈阳建筑工程学院 王宝金
佳木斯大学职业技术学院 王跃国
河北工业大学 范顺成
哈尔滨理工大学工业技术学院 线恒录
洛阳大学 吴 锐
洛阳工学院职业技术学院 李德顺
南昌大学 肖玉梅
厦门大学 朱立秒
湖北工学院高等职业技术学院 吴振彪
彭城职业大学 陈嘉莉
燕山大学 刘德有

序

1999 年 6 月中共中央国务院召开第三次全国教育工作会议。作出了“关于深化教育改革，全面推进素质教育的决定”的重大决策。强调教育在综合国力的形成中处于基础地位，坚持实施科教兴国的战略。决定中明确提出要大力发展高等职业教育，培养一大批具有必备的理论知识和较强的实践能力、适应生产、建设、管理、服务第一线急需的高等技术应用性专门人才。为此，教育部召开了关于加强高职高专教学工作会议，进一步明确了高职高专是以培养技术应用性专门人才为根本任务，以适应社会需要为目标；以培养技术应用能力为主线设计学生的知识、 能力、素质结构缓和培养方案；以“应用”为主旨和特征来构造课程体系和教育内容体系；高职高专的专业设置要体现地区、行业经济和社会发展的需要，即用人的需要；教材可以“一纲多本”，形成 7 有特色的高职高专系列教材。

“教学育人，教材先行”，教育离不开教材。为了贯彻中共中央国务院以及教育部关于高职高专人才培养目标及教材建设的总体要求，中国机械工业教育协会、机械工业出版社组织全国部分有高职高专教学经验的职业技术学院、普通高等 学校编写了这套《21 世纪高等职高专系列教材》。教材首批 80 余本（书目附书后）已陆续出版发行。

本套教材是根据高中毕业 3 年制（总学时 1600~1800）、兼顾 2 年制（总学时 1100~1200）的高职高专教学计划需要编写的。在内容上突出了基础理论知识的应用和实践能力的培养。基础理论课以应用为目的，以必要、够用为度，以讲清概念、强化应用为重点；专业课加强了针对性和实用性，强化了实践教学。为了扩大使用面，在内容的取舍上也考虑到电大、职大、夜大、函大等教育的教学、自学需要。

每类专业的教材在内容安排和体系上是有机联系、相互衔接的，但每本教材又有各自的独立性。因此各地区院校可根据自己的教学特点进行选择使用。

为了提高质量，真正编写出有显著特色的 21 世纪高职高专系列教材，组织编写队伍时，采取专门办高职的院校与办高职的普通高等院校相互协作编写并交叉审稿，以便实践教学和理论教学能相互渗透。

机械工业出版社是我国成立最早、规模最大的科技出版社之一，在教材编辑出版方面有雄厚的实力和丰富的经验，出版了一大批适用于全国研究生、大学本科、专科、中专、职工培训等各种层次的成套系列教材，在过内享有很高的声誉。我们相信这套教材也一定能成为具有我国特色的、适合 21 世纪高职高专教育特点的系列教材。

中国机械工业教育协会

前　言

本书是高等职业技术教育理工科计算机类软件工程课教学用书。

高职高专教育的培养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性人才。软件工程本身又是一门实用性很强的年轻学科，具有实践性的显著特点。编写本书时，我们正是据此出发，明确了指导思想：在内容的编排上强调针对性，理论上强调必需、够用，技术方法上强调实用、管用，突出高职高专教育特色。

教材的内容注重理论与实践技能相结合，以软件工程学的基本理论和基本概念为基础，结合大量实例重点介绍软件工程的基本方法及其应用。全书共9章，主要内容包括绪论、软件定义及计划、软件需求分析、软件设计、软件编码、软件测试、软件运行及维护、软件项目管理技术、面向对象软件工程等。其中既给出了运用传统软件工程学的方法进行软件开发的详细过程与开发实例、软件开发文档的编写及实例、基本的软件项目管理技术，也介绍了最新软件工程方法的基本思想和开发过程。每章后均配有适量复习思考题，用以帮助学生加深对本章内容的消化和理解。教材附录中还提供了标准软件开发文档和实习题目供读者参考和选用。

本书总课时为 48 学时，各院校及使用者可根据实际情况决定内容的取舍，部分内容可指定学生自学。

参加本书编写的单位及人员有：第 1 章 天津大学职业技术教育学院 周志刚；第 2、3、4 章 天津大学职业技术教育学院 孙秀钰；第 5、6 章 包头职业技术学院 张海黎；第 7、8 章 天津职业技术师范学院 王珍玲；第 9 章 南昌大学 林仲达；附录 天津大学职业技术教育学院 孙秀钰。

本书由周志刚主编，他提出了全书的总体构思及编写指导思想。孙秀钰、林仲达为副主编，他们对本书的编写提出了很多合理建议，使得本书的撰写能够顺利进行。由周志刚和孙秀钰进行了全书的统稿工作。

本书由大连理工大学刘润彬教授主审。他对本书的编写提纲以及全部书稿进行了审阅，并提出了很多的宝贵意见，在此表示衷心的感谢。

在本书的编写过程中还得到了天津大学教务处、天津大学职业技术职业教育学院李增武院长和有关同志的支持，在此一并表示感谢。

限于作者的水平，加之时间仓促，书中难免有缺点和不当之处，敬请专家、同仁及广大读者批评指正。

编　者

目　录

第1章 绪 论

计算机软件是计算机应用的灵魂。20 世纪 60 年代以后，随着软件产业的发展，计算机应用愈加广泛，几乎涉及到社会生活的各个方面。而对软件产业的形成和发展起着决定性的推动作用的，是 20 世纪 60 年代末期以来逐渐形成的一门新兴学科——软件工程。

本章主要介绍软件概念的产生与发展，软件危机及其解决，软件工程的概念、内容与目标，软件工程的基本方法、工具及过程模型等内容，使读者对软件工程的总体框架有概要的了解。

1.1 软件技术发展概述

1.1.1 软件概念及其产生

“软件”一词是 20 世纪 60 年代初从国外传入国内的。与“软件”相对应的一个词是“硬件”，在英文中这两个词分别是“software”和“hardware”，计算机系统即是由计算机软件系统与计算机硬件系统组成的。

众所周知，计算机技术的发展是从硬件开始的。1946 年第一台电子计算机 ENIAC 问世以后，用于算法的程序从硬件中分离出来，成为“软件”的雏形。在此后的 50 多年里，计算机的硬件技术发生了极大的变化，并日臻成熟。每次硬件技术的突破，都为软件技术的发展提供了更为广阔的空间，开拓了新的更广阔的应用领域。与此同时，软件的概念也在不断地变化和扩展。

目前，对于软件概念的一种公认的解释是：软件是计算机系统中与硬件相互依存的另一部分，包括程序、数据及相关文档的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列；数据是使程序正常操纵信息的数据结构；文档是与程序开发、维护和使用有关的图文材料。

1.1.2 软件发展的三个阶段

计算机软件的发展，主要经历了三个阶段。

1. 程序设计阶段(约为 20 世纪 50 年代至 60 年代) 这一阶段通用硬件已经相当普遍，但软件概念还没有正式出现。软件一般就是指为具体应用而编写的程序指令，规模很小，编写者和使用者往往是一个人。这一阶段的软件开发缺乏管理，也没有系统化的软件开发方法可以遵循，几乎没有文档资料保存下来。

2. 程序系统阶段(约为 20 世纪 60 年代至 70 年代) 这一阶段计算机经历了由电子管到晶体管，到集成电路的跃迁，为软件技术的发展提供了坚实的物质基础。此时的操作系统引入了多道程序、多用户分时和实时处理概念，程序开始系统化，出现了软件产品，用户开始使用购买的软件，软件规模和数量也因此急剧增加。但是，软件开发仍沿用早期个体化的开发方法，“软件作坊”成为软件开发的基本形式，由于软件技术的发展不能满足用户的需求，逐渐出现了“软件危机”。在这一阶段，软件的概念一般是指程序及其规格说明书。

3. 软件工程阶段(约为 20 世纪 70 年代至今) 随着超大规模集成电路技术的迅猛发展，

此时计算机的应用已经深入到社会生活中的各个领域，以软件产品化、系列化、工程化、标准化为标志的软件产业逐渐兴起，“软件作坊”演变成了软件工厂或公司，打破了软件开发的个体化特征，软件开发有了软件工程化的设计原则、方法和标准可以遵循。在该阶段内，软件危机经历了由加剧到逐渐得到解决的过程，并形成了目前对软件概念的公认解释，即包含程序、文档以及相关数据的集合。

目前，技术性的软件工程阶段正在逐渐向企业计算机化、社会信息化的计算机系统工程阶段过渡，以满足现代社会各种计算机应用对计算机软件的巨大需求。

1.1.3 软件危机及其解决

1. 软件危机的概念　在软件技术发展的第二阶段，随着计算机硬件技术的发展，硬件成本急剧下降；与此同时，计算机应用对于软件的需求急剧增加，软件的规模增大，数量膨胀，用户由于购买软件，产生了对于软件维护的需求。但是，这一时期仍然沿用早期个体化的软件开发方法，有些软件的个体化特征，使它根本无法维护。这样，在用户的需求和软件开发的实际状况之间形成了尖锐的矛盾，这就形成了20世纪60年代所谓的“软件危机”。

概括地说，软件危机是指“在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。”这些问题主要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。

2. 软件危机的表现

(1) 不能准确地估计软件开发的成本和进度。 由于缺乏软件开发经验和科学的理论指导，盲目制订开发计划，或根本没有制订可以遵循的开发计划，造成实际的开发成本可能比估计的要高得多。开发的进度可能要拖上几个月甚至几年，软件开发出来，可能就过时了，这种现象严重影响了开发组织的声誉。有时为了赶进度和节约成本，采取一些权宜之计，又会降低软件质量，引起用户的强烈不满。

(2) 用户对“已完成的”软件系统不满意。 由于开发人员与用户之间信息交流不充分，开发人员没有仔细了解用户的需求，没有确切地认识到所要解决的问题，就“仓促上阵”，“闭门造车”，在干了一阵以后，或开发完成后，才发现软件不满足用户的需求；或者是由于用户在开发初期不能明确地提出对软件的需求，直到开发后期才逐渐明确，这时为时已晚，软件修改已很困难，也会造成用户对“已完成的”软件系统不满意。

(3) 软件产品的质量不可靠。 由于测试过程的工作不够充分，又没有好的软件质量保证技术，导致提交给用户的软件质量较差。

(4) 软件的可维护性差。 其主要表现是：源程序存在的错误难以改正；程序的可移植性差，很难适应新的系统环境；不能根据用户的需求增加新功能；软件的可重用性差，大量的软件人员还在重复开发基本类似的软件等。

(5) 开发过程没有统一公认的方法和规范指导，又缺乏设计和实现过程完整的文档资料。

(6) 软件开发生产率的提高速度难以满足对软件需求的增长速度，软件产品“供不应求”。

(7) 由于软件成本居高不下，硬件的性价比迅速提高，软件成本在计算机系统总成本中所占的比率逐年上升。

以上只是列举了软件危机的主要表现，其实，与软件开发和维护有关的问题还有很多。

3. 软件危机产生的原因　在软件开发和维护过程中存在着这么严重的问题，其原因很多，归结起来主要有两个方面：一是与软件本身的特点有关，二是与软件开发与维护方法不

当有关。

(1) 软件的特点：软件与硬件不同，硬件是一种物理实体，而软件是一种逻辑实体，具有抽象性，它具有以下的一些基本特点：

1) 软件是在研制、开发过程中形成的，而不是传统意义上“制造”产生的。 软件在开发过程中，需要花大力气，一旦研制、开发成功后，大批量地生产几乎不花什么成本。在硬件的制造过程中可能会出现质量问题，而软件的制造只是简单的拷贝，质量问题几乎不存在。只是软件开发的进程难以衡量，质量难以评价，因此，管理和控制软件开发的过程相当困难。

2)软件不会被磨损，更不会被“用坏”，但存在着“退化”问题。 硬件使用时间久了，就会出现机械磨损、老化问题，其失效率呈现如图 1-1a 所示的 U 形曲线。软件的失效率曲线则如图 1-1b 所示，在软件的运行和使用期间，不存在磨损、老化问题，却存在着退化问题。如果软件运行过程中出现了错误，不是由于使用造成的，而可能是由开发时期引入的在测试阶段没有检测出来的问题造成的。这样，软件的维护需要多次改正或修改最初的设计，每次修改不可避免地会引入新的错误，使软件失效率升高，因此就出现了软件的退化问题，使得软件很难被维护。

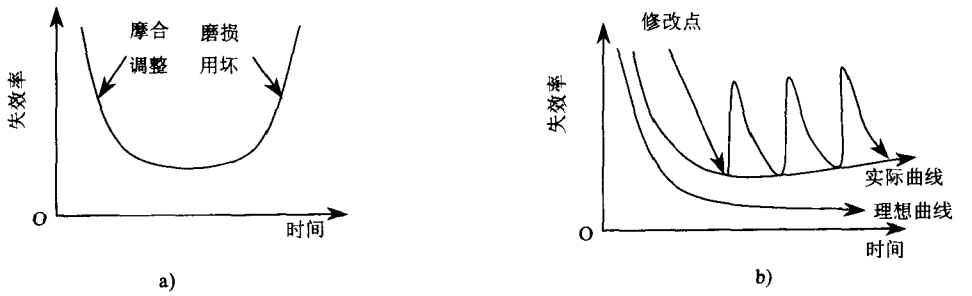

图 1-1 硬件和软件的失效率曲线

3) 软件的规模庞大，逻辑结构复杂。软件开发往往涉及到其他领域的专门知识，而且往往与社会、人的组织和管理因素相关，所以其开发和维护必然困难。

4) 多数软件是自定义的，而不是通过已有的构件组装的。 虽然近年来软件技术取得了不少进展，但是软件的开发至今没有完全摆脱手工的方式，很少能做到利用现成的部件组装成所需要的软件产品，因此，软件的可重用性差，开发的效率受到很大的限制。

5) 软件的开发、运行受到计算机系统的限制。 软件不能摆脱硬件单独存在。有的软件常常是为某个型号的计算机所专用的，有的软件依赖于某个操作系统，脱离了该操作系统就不能运行，这就出现了软件难以移植的问题。

以上我们所讨论软件的特点，是产生软件危机的客观原因。另一方面，在软件开发和维护过程中存在着一系列的错误认识和做法，是产生软件危机的主观原因，也是更为重要和可以克服的因素。

(2) 软件开发和维护过程中的错误认识和做法 在软件开发和维护过程中的错误认识和做法主要表现在忽视软件需求分析的重要性和轻视软件维护等方面。

1) 忽视软件需求分析和可行性研究工作，对客户的要求没有完整、准确地认识就匆忙着手编写程序。 实际上软件的需求分析和可行性研究工作，对于软件的成功开发是至关重要的，研究与实践都表明，一个软件错误修改的代价与该错误发生与修改滞后的时间关系极

大。但是，很多用户往往开始并不能准确地叙述他们的需求，软件开发人员只有通过大量的调研工作，与用户进行充分交流，才能真正全面、准确、具体地了解用户的要求，为以后的开发工作打好坚实的基础。

2) 轻视维护，认为维护工作是容易做的简单工作。软件人员经常认为程序编完交付使用就算完事。实际上，程序只是完整的软件产品的一个组成部分，一个软件产品必须由一个完整的配置组成，编写程序也只是软件开发过程中的一个阶段。有些软件产品寿命较长，在使用期间要改正错误，可能还要面临系统更新换代，还要经常扩充、修改满足用户不断变化的需求。这些维护工作是艰难而复杂的，费用也很高，约占总开发费用的 55% ~70%左右。

软件开发和维护过程中存在着的错误认识和做法的形成，可以归因于早期软件开发个体化的特征，软件人员没有掌握恰当的工程化方法，同时也缺乏先进的软件管理技术和规范的开发标准。

4. 软件危机的解决途径　了解了软件危机产生的原因，就可以找到解决它的办法。解决软件危机主要有以下三种途径。

(1) 采用工程化方法和工程途径来开发和维护软件。 软件开发不应只是个体化的劳动，而应该是由组织良好、管理严密、各类人员共同配合完成的一个工程项目，因此应该注意吸收和借鉴从事其他工程项目行之有效的科学原理和方法。

(2) 采用先进的技术、方法、工具开发和设计软件。 包括先进的生产管理技术、规范的开发方法和模型、各种提高开发效率的软件工具等。

(3) 采用必要的组织管理措施。

事实上，找到解决软件危机的途径就是软件工程形成的过程。软件工程正是在解决软件危机问题的过程中形成的一门综合技术与管理两个方面的新兴学科，并逐渐成为指导计算机软件开发、维护、管理的理论依据。

1.2　软件工程

1.2.1　软件工程的概念

“软件工程”一词，是 1968 年北大西洋公约组织在联邦德国召开的一次会议上为解决软件危机问题而首次提出的，从此逐渐形成了一门新兴的学科——软件工程。30 多年来，人们曾从不同的角度，给软件工程下过定义。Boehm 曾经将软件工程定义为：“运用现代科学技术知识设计并构造计算机程序及为开发、运行、维护这些程序所必需的相关文件资料。”IEEE 为软件工程下的定义是：“软件工程是开发、运行、维护和修复软件的系统方法。”

我们将软件工程定义为：“采用工程化的概念、原理和技术、方法来开发和维护软件，把经过时间考验证明是正确的管理技术和当前最好的技术方法结合起来指导计算机软件开发和维护的工程学科。”

1.2.2　软件工程的内容与目标

1. 软件工程的内容　软件工程的内容也就是软件工程所包含的要素，主要包括方法、工具和过程三个方面。

软件工程方法为软件开发提供了“如何做”的技术。它包括了多方面的任务，如项目的计划与估算、软件系统的需求分析、数据结构和系统总体结构的设计、算法过程的设计、编码、测试以及维护等。软件工程方法常采用某一种特殊的语言或图形的表达方法及一套质

量保证标准。

软件工程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。

软件工程的过程则是将软件工程的方法和工具综合起来以合理、及时地进行软件开发的软件工程活动及结构框架。过程定义了方法使用的顺序、要求交付的规范文档资料、为保证质量和协调变化所需要的管理及软件开发各个阶段的里程碑等。

2. 软件工程的目标　软件工程的根本目标是要消除软件危机。对一个具体的软件工程项目来讲，采用软件工程的技术和管理方法组织软件的开发，最终目标是希望获得软件项目的成功，实现软件产品的正确、可靠和高效率。具体地讲，软件工程的目标主要包括：

1) 付出较低的开发成本。

2) 达到要求的软件功能。

3) 取得较好的软件性能。

4) 开发的软件易于移植。

5) 需要较低的维护费用。

6) 有较高的开发效率，能按时完成开发工作，及时交付使用等等。

在实际的开发过程中，要想使以上的几个目标都能达到理想程度往往是很困难的，有些目标之间可能存在着冲突。因此，实施软件开发项目还要努力协调以上目标，突出重点，取得平衡。

1.2.3　软件工程的过程模型

软件工程过程是为获得软件产品，以某种开发方法为基础，在软件工具支持下由软件开发人员完成的一系列的软件工程活动及其结构框架。软件工程过程通常包含四种基本的过程活动：软件规格说明、软件开发、软件确认、软件演进。将这四种基本活动进行展开，并安排不同的衔接步骤和结构框架，可以得到不同的过程模型。构成不同的过程模型的基础是软件的生存周期。

1. 软件的生存周期(Software Life Cycle)　软件的生存周期是软件工程中最基本的概念。软件生存周期，指一个软件系统从目标提出到最终被淘汰的整个存在期。软件生存周期可以划分为若干阶段，每个阶段有相对独立的任务，有特定的方法和工具。

对软件的生存周期如何进行划分，不同的教材略有不同。在本书中，将软件的生存周期划分为以下几个阶段：

(1) 软件定义及计划：在软件定义及计划阶段应确定软件系统要解决问题的性质、任务，分析解决该问题的可行性，当问题具有可行解时，还要制订完成任务的计划。

(2) 需求分析：需求分析阶段的任务是确定为了满足用户的需要系统必须做什么。也就是对目标系统实现的功能和性能以及系统要求的运行环境等提出完整、准确、清晰、具体的要求，分析系统中的数据，并在分析的基础上得出用各种图形工具及简洁算法所描述的系统详细的逻辑模型。

(3) 软件设计：软件设计大体上可以分为总体设计(也称概要设计)和详细设计两部分。

总体设计的主要任务，一是概要地给出系统的实现方法，划分出组成系统的物理元素；二是确定系统的软件结构，即组成系统的各个模块及模块之间的关系。详细设计要对软件结构图的每个模块所采用的逻辑关系进行分析，设计出全部必要的过程细节，并给出清晰的描述，从而在编码阶段可以把这个描述直接翻译成用某种程序设计语言书写的程序。

(4) 编码：编码阶段的任务是选用适当的程序设计语言，将设计阶段得到的结果转换成用选定的语言所表示的“源程序清单”，写出的程序应是结构良好、清晰易读的。

(5) 软件测试：软件测试阶段的任务是要通过各种类型的测试，发现并改正软件中存在的错误，使软件达到预定的要求，能够交付用户使用。

(6) 软件运行及维护：软件运行及维护是软件生存周期的最后一个阶段，它的主要任务是在软件产品投入使用后，对其进行调整和修改，以改正在开发阶段产生在测试阶段未发现的错误，实现软件功能的扩充、性能的改善，使软件适应新的工作环境等。

2. 瀑布模型　瀑布模型是传统的软件工程过程模型。根据前面对软件生存周期各个阶段的划分形成的瀑布模型，如图 1-2 所示。瀑布模型将软件生存周期中的各个软件工程活动自上而下进行排列，各个活动之间相互衔接，次序固定，构成如同瀑布流水、逐级下落的结构框架，每个阶段的任务完成后都要提交相应的文档资料，进行评审和复审，审查通过后再进入到下一个阶段的工作，逐步完成各阶段的开发任务。其中前两个阶段可称为软件的定义时期，中间的几个阶段称为软件的开发时期，最后的一个阶段称为维护时期。

瀑布模型中各个阶段要提交的文档见表 1-1。除此表中的文档外，在开发过程中还要填写进度月报，开发完成后还要撰写开发总结报告等文档，有关文档的规范格式及要求详见附录 A。

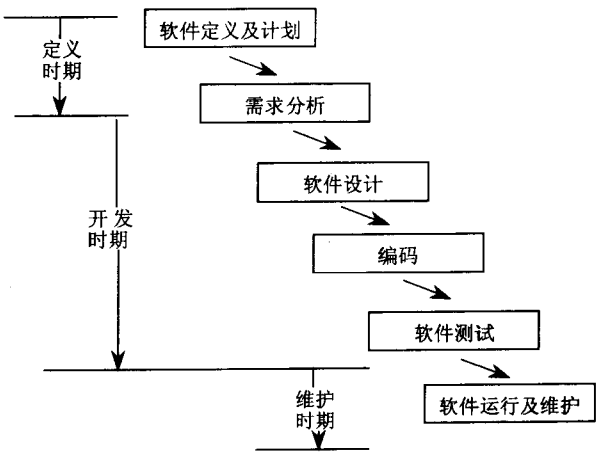

图 1-2　瀑布模型

表 1-1　瀑布模型各阶段要提交的文档

阶　段	提交的主要文档
软件定义及计划	可行性研究报告、软件开发计划
需求分析	软件需求规格说明书
软件设计	软件总体设计和详细设计说明书
软件编码	源程序清单和用户使用手册
软件测试	测试计划、测试报告
软件维护	软件维护问题报告、修改报告、维护手册

瀑布模型的优点主要是简单、便于分工协作、降低开发难度、开发成功率高、保证质量等，但它的缺点也比较明显，主要包括：

(1) 用户往往在开始阶段难以清楚地给出所有的需求，开发后期要改正早期存在的问题需要付出很高的代价。

(2) 用户需要等待较长时间，因为程序运行版本要在开发晚期才能得到。

(3) 开发过程存在着阻塞问题，影响开发的效率。因为有些人要等待其他人任务完成后才能开始工作，且有时时间会比较长。

2. 原型模型　原型模型是针对瀑布模型的的缺点提出的，如图 1-3 所示(为简单起见，只绘出其中主要的阶段)。由于在软件计划时期定义的用户需求往往是不完全和不准确的，因此原型模型不要求一开始就有一个完整的软件需求定义，而只要有一个总体的、轮廓并不分明的要求，适用于用户不能标识出详细的输入、输出及处理，开发者也不能确定算法的有效性、系统的适应性及人机交互的形式的情况。

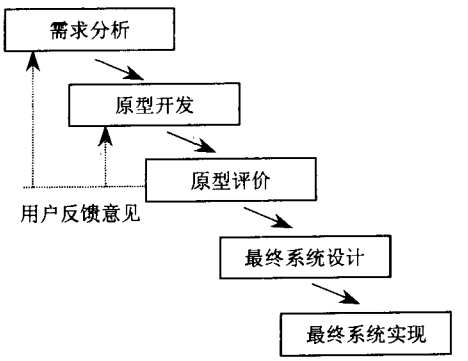

图 1-3　原型模型

原型模型的主要思想是：先建立一个能够反映用户需求的原型系统，使得用户和开发者可以对目标系统的概貌进行评价和判断，然后对原型进行反复的扩充、改进和求精，最终建立完全符合用户需求的目标系统。

根据最初原型转化为目标系统的途径不同，原型模型又可分为抛弃原型模型和演化原型模型两种。抛弃原型模型一开始的原型使用完毕后抛弃，重新建立目标系统，相当于生命周期法中的需求分析与设计的过程。演化原型模型的目标系统是对初始模型不断扩充、完善、迭代的结果，每次迭代都要求再设计、再分析、再实现以及再测试评价，直到认为得到问题的基本满意的解法时就可以交付系统的初始版本了。许多面向市场的软件产品采用 β 版发行，就是演化原型模型开发过程的一种具体实施策略。

采用原型模型进行软件开发需要有快速建立原型模型的的软件工具与环境。

3. 螺旋模型　螺旋模型如图 1-4 所示。螺旋模型是瀑布模型和原型模型的融合体，并且加入了两种模型都忽略的风险分析，弥补了两者的不足，适合于复杂的大型软件的开发。因为软件项目越大，问题越复杂，各种因素的不确定性就越大，承担该项目的风险也就越大，因此，风险分析是不能忽略的。

螺旋模型沿着螺旋线自内向外旋转，每转一圈便开发出更为完善的一个新的软件版本，直到得到所期望的系统。当风险分析的结果认为风险过大时，开发有可能被终止。如果对所要开发的软件项目的需求把握较大时，无需开发原型，螺旋模型就变形为单圈螺旋线，这就是普通的瀑布模型。

当然还有其他的智能模型、喷泉模型等各种软件开发的过程模型，有兴趣的读者可参见参考资料中列出的其他参考书目。

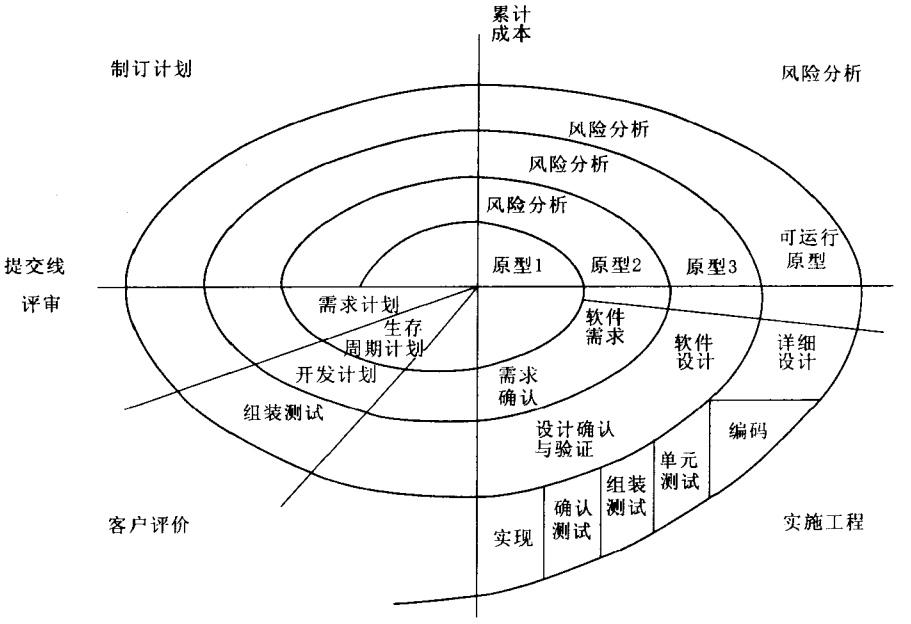

图 1-4　螺旋模型

1.2.4　软件工程的基本方法

软件工程的基本方法包括传统的结构化方法、面向对象的方法等，无论哪种方法都是以软件生存周期为基本特征的软件工程方法。

1. 结构化方法(Structure Method)　结构化方法是较传统的软件工程方法。在 20 世纪 60 年代初，就提出了用于编写程序的结构化程序设计(SP)方法，而后发展到用于设计的结构化设计(SD)方法、用于分析的结构化分析(SA)方法等。在早期软件生存周期的各个阶段中，使用的方法就是结构化方法。

结构化方法的基本思想可以概括为自顶向下、逐步求精，采用模块化技术和功能抽象将系统按功能分解为若干模块，从而就将复杂的系统分解成若干易于控制和处理的子系统，子系统又可分解为更小的子任务，最后的子任务都可以独立编写成子程序模块，模块内部由顺序、选择、循环等基本控制结构组成。这些模块功能相对独立，接口简单，使用维护非常方便。所以，结构化方法是一种非常有用的软件工程方法，是其他软件工程方法的基础。

但是，由于结构化方法将过程与数据分离为相互独立的实体，因此开发的软件可重用性较差，在开发过程中要使数据与程序始终保持相容也很困难。这些问题通过面向对象方法就能得到很好的解决。

2．面向对象方法(Object-Oriented Method)　面向对象方法是针对结构化方法的缺点，为了提高软件系统的稳定性、可修改性和可重用性而逐渐产生的。开始主要用在程序编码中，以后又逐渐出现了面向对象的分析(OOA)和设计(OOD)方法，是当前软件工程方法的主要方向，也是目前最有效、实用和流行的软件开发方法之一。