面向21世纪

高职高专系列教材

软件工程

◎王宜贵　主编

◎眭碧霞　审

机械工业出版社
China Machine Press

面向21世纪高职高专系列教材

软 件 工 程

王宜贵 主编
眭碧霞 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、软件开发方法及开发工具。第1章介绍软件危机、软件生命周期、软件开发模型和方法；第2～6章及第8章介绍软件生命周期各个阶段的任务和方法，以结构化方法为主；第7章介绍面向对象的软件工程；第9章介绍软件复用技术；第10、11、12章分别介绍软件管理、软件质量保证、软件工具和软件开发环境；第13章是一个软件开发实例。

本书适用于高职高专技术学院计算机专业学生。

图书在版编目（CIP）数据

软件工程/王宜贵主编. —北京：机械工业出版社，2002.9
面向21世纪高职高专系列教材
ISBN 7-111-10789-6

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（2002）第061205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：胡毓坚
责任编辑：田　梅
责任印制：付方敏

北京市密云县印刷厂印刷·新华书店北京发行所发行

2002年9月第1版·第1次印刷
1000mm×1400mm　B5·7.25印张·329千字
0001—5000册
定价：18.00元

面向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

积极发展高职高专教育，完善职业教育体系，是我国职业教育改革和发展的一项重要任务。为了深化职业教育的改革，推进高职高专教育的发展，培养21世纪与我国现代化建设要求相适应的，并在生产、管理、服务第一线从事技术应用、经营管理、高新技术设备运作的高级职业技术应用型人才，尽快组织一批适应高职高专教学特色的教材，已成为各高职高专院校的迫切要求。为此，机械工业出版社与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联合组织了全国40多所院校的骨干教师，共同研究开发了一批计算机专业、电子技术专业和机电专业的高职高专系列教材。

各编委会确立了“根据高职高专学生的培养目标，强化实践能力和创新意识的培养，反映现代职业教育思想、教育方法和教育手段，造就技术实用型人才为立足点”的编写原则。力求使教材体现“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。

本套系列教材是由高职高专计算机专业、电子技术专业、机电专业教材编委会分别会同各院校第一线专业教师针对高职高专计算机、电子技术和机电各专业的教学现状和教材存在的问题开展研讨，尤其针对目前高职高专教学改革的新情况，分别拟定各专业的课程设置计划和教材选题计划。在教材的编制中，将教学改革力度比较大、内容新颖、有创新精神、比较适合教学、需要修编的教材以及院校急需、适合社会经济发展的新选题优先列入选题规划。在广泛征集意见及充分讨论的基础上，由各编委会确定每个选题的编写大纲和编审人员，实行主编负责制，编委会通过责任编委和主审对教材进行质量监控。

担任本套教材编写的老师们都是来自各高职高专院校教育第一线的教师，他们以高度的责任感和使命感，经过近一年的努力，终于将本套教材呈现在广大读者面前。由于高职高专教育还处于起步阶段，加上我们的水平和经验有限，在教材的选题和编审中可能出现这样那样的问题，希望使用这套教材的教师和学生提出宝贵的意见和建议，以利我们今后不断改进，为我国的高职高专教育事业的繁荣而共同努力。

高职高专系列教材编委会

机 械 工 业 出 版 社

前　言

高职高专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。教材内容强调针对性，理论上强调必需、够用，技术方法上强调实用。编写本教材的指导思想就是基于这些要求。

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、软件开发方法和开发工具，包括软件工程中软件定义、设计、测试、维护、面向对象技术、软件复用技术、软件管理和质量保证、软件开发工具和环境等。

本书的编写过程中，在注意系统性完整性的同时，注重针对高职高专教材内容的实用性，符合高职高专技术院校学生的需要。第 1 章介绍软件危机、软件生命周期、软件开发模型和方法；第 2～6 章和第 8 章介绍软件生命周期各个阶段的任务和方法，以结构化方法为主；第 7 章介绍面向对象的软件工程；第 9 章介绍软件复用技术；第 10、11、12 章分别介绍软件管理、软件质量保证、软件工具和软件开发环境；第 13 章是一个软件开发实例。

本教材责任编委由赵佩华老师担任，王宜贵老师主编、眭碧霞老师审稿。第 1、2、3、4、9、12、13 章由王宜贵编写，第 5、6、7、8 章由李晓方编写，第 10、11 章由张静妙编写。

限于我们的水平，书中缺点和错误难免，恳请读者批评指正。

编　者

目　录

第1章　软件工程概述

1.1　软件的概念

1. 软件的含义

随着计算机技术的发展，对软件在不同阶段有不同的认识。计算机发展的初期，硬件的设计和生产是主要问题，那时的所谓软件就是程序，甚至是机器指令程序，它们处于从属的地位。软件的生产方式是个体手工方式，设计过程是在一个人的头脑中完成的，程序的质量完全取决于个人的编程技巧。其后，人们认识到在计算机上增加软件的功能会使计算机系统的功能大大提高，因此在研制计算机体系时既要考虑其硬件又考虑与之配套的软件，方可开始编制一些大型程序系统。这时的生产方式为互助合作的手工方式，认为软件就是程序加说明书。

随后社会对计算机提出了更高的要求，有的大型系统的设计和生产的工作量高达几千人/年(一个人工作一年其工作量为一人年)，指令数百万条。现在，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，研制软件设计和生产的方法及工具，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。

现在对软件的正确理解应该是，软件是计算机系统中与硬件相互依存的部分，它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。

2. 软件的特点

和硬件相比，软件主要有以下特点：

(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，一直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，对软件的质量控制必须立足于软件开发过程。

(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，但却存在需要更新的问题。因为在软件的生存期中，它一直处于改变(维护)状态，随着针对某些缺陷的维护，可能带来一些新的缺陷，使软件的故障率增加。而软件不像硬件一样有备件，它的维护要比硬件复杂的多。

3. 软件的分类

(1) 按软件的功能

- 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。
- 支撑软件。是协助用户开发应用软件的工具性软件，包括帮助程序人员开发软件产品的工具，也包括帮助管理人员控制开发进程的工具。
- 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

按开发软件所需人力、时间以及完成的源程序的行数，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

- 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。
- 分时软件。允许多个联机用户同时使用计算机的软件。
- 交互式软件。实现人—机通信的软件。
- 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

- 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、卫星控制系统等。
- 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，软件开发的生产率相对迟缓。

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，且经费上大大超出了预算；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系统时这些矛盾最为突出。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”，其表现为：

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可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”需作好以下几个方面工作：

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。

1.3 软件工程的目标和原则

1. 软件工程的概念

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、可靠性好并能高效运转的软件，从而解决或缓解软件危机。

软件工程学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。

软件工程包括三个要素，即方法、工具、过程。软件工程方法是指导研制软件的某种标准规范；软件工具是指软件开发、维护和分析中使用的程序系统，为软件工程方法提供自动的或半自动的软件支撑环境；软件工程过程指将软件工程的方法和工具结合起来，以达到合理、及时地进行计算机软件开发的目的。过程定义了方法使用的顺序、要求交付的文档资料、保证质量和协调变化所需的管理。

2. 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，即达到以下几个主要的目标：

- 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。

在项目实际开发中，使以上几个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1-1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。有些目标之间是互斥的，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以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。

3. 软件工程的原则

(1) 选取适宜的开发模型

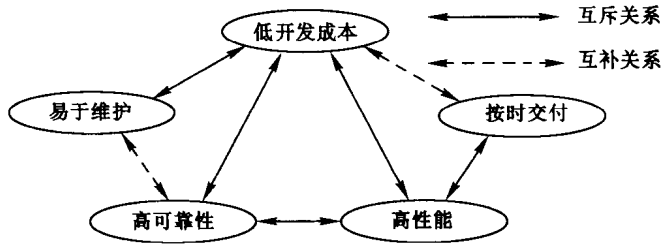

图 1-1　软件工程目标之间的关系

该原则与系统设计有关。在系统设计中，软件需求、硬件需求以及其他因素之间是相互制约、相互影响的，经常需要权衡。因此，必须认识需求定义的易变性，采取适宜的开发模型予以控制，以保证软件产品满足用户的需求。

(2) 采取合适的设计方法

在软件设计中，通常要考虑软件的模块化、抽象与信息隐蔽、局部化、一致性以及适应性等。

(3) 提供高质量的工程支持

在软件工程中，软件工具与环境对软件过程的支持颇为重要。软件工程项目的质量与开销直接取决于对软件工程所提供的支撑质量和效用。

(4) 重视开发过程的管理

软件工程的管理直接影响可用资源的有效利用，有了有效的管理才能生产满足目标的软件产品，提高软件组织的生产能力。

1.4　软件生命周期和开发模型

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，直到停止该软件的使用的全过程。包括制定计划、需求分析、软件设计、程序编码、软件测试及运行维护 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要求；研制完成该项软件任务的可行性，探讨解决问题的可能方案；指定完成开发任务的实施计划，连同可行性研究报告，提交管理部门审查。

(2) 需求分析

对待开发软件提出的需求进行分析并给出详细的定义，编写软件需求说明书及初步的用户手册，提交管理机构评审。

(3) 软件设计

把已确定的各项需求转换成相应的体系结构，进而对每个模块需完成的工作进行具体描述，编写设计说明书，提交有关部门评审。

(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。

(5) 软件测试

在设计测试用例的基础上,检查软件的各个组成部分。

(6) 运行和维护

已交付的软件正式运行,并在使用过程中进行适当维护。

软件开发模型是从软件项目需求定义直至软件使用后废弃为止,针对系统开发、运作和维护所实施的全过程、活动和任务的结构框架。

最早的软件开发模型是 1970 年 W. Royce 提出的瀑布模型,而后随着软件工程学科的发展和软件开发的实践,相继提出了演化模型、螺旋模型、增量模型、喷泉模型等。

1. 瀑布模型

瀑布模型规定了各项软件工程活动,包括开发计划制定、需求分析和说明、软件设计、程序编码、测试及运行维护,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 1-2 所示。

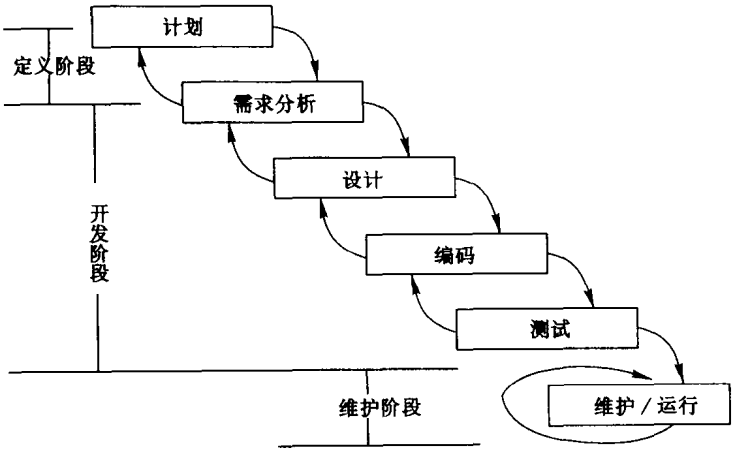

图 1-2 瀑布模型

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。

瀑布模型 20 多年来之所以广泛流行,是因为它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，为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。

2. 演化模型

由于在项目的初始阶段对软件的需求认识往往不够清晰，使得开发项目难以做到一次开发成功，出现返工再开发的情况在所难免。因此，可以先做试验开发，其目标只是探索可行性，弄清软件需求。用户试用，并提出精化系统、增强系统能力的需求。然后在此基础上获得较为满意的软件产品。通常把第一次得到的试验性产品成为“原型”。

3. 螺旋模型

螺旋模型是将瀑布模型与演化模型结合起来，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。螺旋模型的结构如图 1-3 所示，它由四部分组成：制定计划、风险分析、工程实现、用户评价。

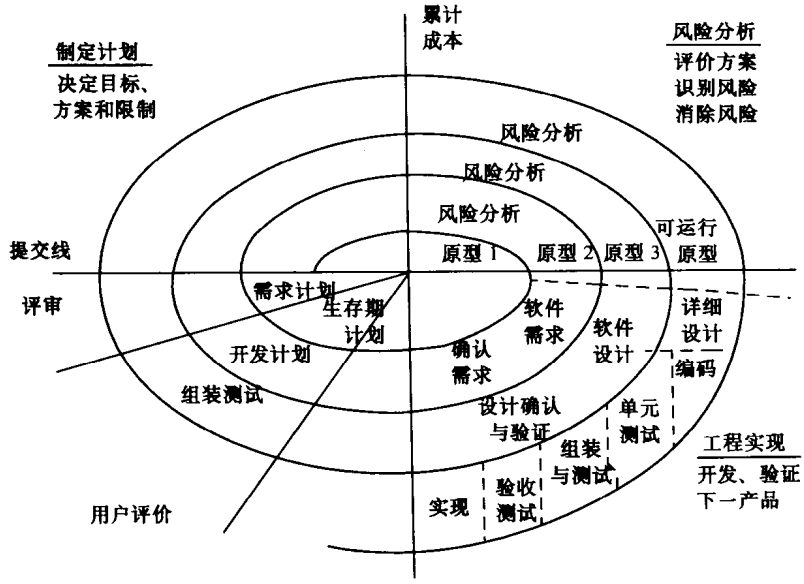

图 1-3　螺旋模型

- 制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件。
- 风险分析：分析所选方案，考虑如何识别和消除风险。
- 工程实现：实施软件开发。
- 用户评价：评价开发工作，提出修改建议。

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。

螺旋模型适合于大型软件的开发，它吸收了软件工程“演化”的概念，使得开发人员和用户对每个演化曾出现的风险有所了解，继而做出应有的反映。但要求许多用户接受和相信演化方法并不容易。这个模型的使用需要具有相当丰富的风险评估经验和专门知