普通高等教育“十二五”规划教材
高等院校计算机系列教材

# 软件工程

李浪　朱雅莉　熊江 ◉ 主编

華中科技大學出版社
http://www.hustp.com

普通高等教育“十二五”规划教材
高等院校计算机系列教材

# 软 件 工 程


**主　编**　李　浪　朱雅莉　熊　江
**副主编**　赵辉煌　易小波　屈喜龙
赵　磊　李　翔　陈紫薇
刘福明　邹　祎　张　佳
张铁楠



华中科技大学出版社
中国·武汉

## 内 容 提 要

本书是结合多年教学和实践经验、参考国内外有关著作(文献)而编写的一本软件工程实用教程。全书针对初学者的特点,由浅入深、系统地讲述了软件工程的基本概念、原理、方法、过程和工具,包括软件生存周期、软件分析、软件设计、软件实现与维护、软件管理等。其目的是使学习者学习本书后,能够掌握软件工程的基本原理和过程,应用 UML 建模,熟悉面向对象方法和结构化分析与设计方法。每个章节均配有习题,书后附有习题参考答案。

本书内容详实、重点难点突出,所选案例具有较强的代表性,有助于读者举一反三。本书注重理论性和实用性的结合,收集的例题与习题大多是计算机技术与软件专业技术资格(水平)考试或研究生入学考试的相关内容,特别适合作为大中专院校、各类职业院校及计算机培训学校相关专业课程的教材,也可作为计算机技术与软件专业技术资格(水平)考试的参考用书。

**图书在版编目(CIP)数据**

软件工程/李　浪　朱雅莉　熊　江　主编. —武汉:华中科技大学出版社,2013.9
ISBN 978-7-5609-9157-3
Ⅰ.软…　Ⅱ.①李…　②朱…　③熊…　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(2013)第 132346 号

**软件工程**　　李　浪　朱雅莉　熊　江　主编

策划编辑:朱建丽
责任编辑:王汉江
责任校对:周　娟
责任监印:周治超
出版发行:华中科技大学出版社(中国·武汉)
　　武昌喻家山　　邮编:430074　　电话:(027)81321915
录　　排:武汉金睿泰广告有限公司
印　　刷:华中理工大学印刷厂
开　　本:787mm×1092mm　1/16
印　　张:18
字　　数:458 千字
版　　次:2013 年 9 月第 1 版第 1 次印刷
定　　价:38.00 元

# 高等院校计算机系列教材

# 编　委　会

# 前　言

软件作为信息产业的核心产业之一，深受IT行业的高度重视。近年来，我国软件产业进入快速发展期，随着软件产业规模的日益扩大，人们不得不采用工程化的方法来开发软件，以求经济有效地解决复杂问题。

本书是软件工程学习的基础教程，主要面向初学者，重点讲述了目前软件工程采用的、比较成熟的过程、方法和工具，突出基本原理和技术。在章节内容的编排上结合了作者多年实践教学的经验，力求做到深入浅出、通俗易懂。同时，依据现有考试大纲，涵盖了计算机技术与软件专业技术资格（水平）考试和研究生入学考试中软件工程的知识点，对重点、难点知识进行举例说明，章节后有相应的习题，并在书后附有参考答案。此外，本书为了加深读者对软件工程理论的深入理解，培养读者的实际应用能力，还结合案例讲述了面向对象的开发过程。

全书共分9章，第1章介绍了软件、软件危机、软件工程的概念；第2章对软件生存周期和常用的软件过程模型进行了剖析；第3章介绍了可行性研究、需求分析、结构化分析方法及Visio工具；第4章介绍了结构化设计方法及其常用的详细设计工具；第5章介绍了软件编码和软件测试知识，包括程序复杂性度量、白盒测试、黑盒测试等重要方法的讲解与举例说明；第6章介绍了面向对象方法的基本概念和特点，并重点讲述了统一建模语言；第7章以一个小型的教务信息管理系统为案例介绍了面向对象的开发过程，并介绍了面向对象分析、面向对象设计、面向对象实现三阶段；第8章介绍了软件项目管理的主要知识；第9章介绍了软件工程标准化和新趋势。

第1章由李浪编写，第2、5章由李翔编写，第3、4章由赵辉煌编写，第6章由朱雅莉编写，第7章由赵磊、张佳编写，第8章由易小波编写，第9章由陈紫薇编写。参加审校与部分章节编写的有熊江、屈喜龙、刘福明、邹祎、张铁楠等。本书的作者都是从事多年计算机软件教学和科研的大学教师，在编写的过程中，参考了国内外大量文献资料，结合了多年教学科研经验和成果。尽管我们再三校对，书中可能还存在错误和不足，恳请专家和广大读者指正和谅解。

本书不仅可以作为大中专院校、各类职业院校及计算机培训学校相关专业课程的教材，还可作为计算机技术与软件专业技术资格（水平）考试的参考用书。同时，本书已开发好相应的PPT教学课件，有需要的老师可以在华中科技大学出版社的网站上下载，也可发邮件向我们索取，我们的交流联系方式：zhu-mary@163.com；lilang911@126.com。

作　者

2013年5月

# 目　　录

# 第1章　概　　论

软件在当今的信息社会中占有重要的地位，软件产业是信息社会的支柱产业之一，随着软件应用日益广泛，软件规模日益扩大，人们不得不采用工程的方法来开发、使用和维护软件，以求经济有效地解决软件问题。如今，借助计算机科学与技术、数学、管理科学与工程等学科，软件工程已经从最初计算机科学下的一个学科方向发展成一个以计算为基础的新兴交叉学科。

软件工程作为一门独立的指导计算机软件系统开发和维护的工程学科，其发展已逾50年。20世纪60年代，高级语言的流行，使得计算机的运用范围得到了较大扩展，对软件系统的需求急剧上升，从而产生了“软件危机”。20世纪60年代末，如何克服“软件危机”，为软件开发提供高质、高效的技术支持，受到人们的高度关注。多年来，软件工程的研究和实践取得了长足的发展，虽然距离彻底解决“软件危机”尚有较大的差距，但对软件开发的工程化及软件产业的发展起到了积极的推动作用，提供了良好的技术支持。

## 1.1　软件

### 1.1.1　软件的定义和特点

**1. 软件的定义**

软件是计算机系统中与硬件相互依存的另一部分，是程序、数据及其相关文档的完整集合。一种公认的传统的软件定义为

$$\text{软件} = \text{程序} + \text{数据} + \text{文档}$$

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；数据是使程序能够正确地处理信息的数据结构；文档是与程序开发、维护和使用有关的图文资料。

为了更全面、正确地理解计算机和软件，必须了解软件的特点。软件是整个计算机系统中的一个逻辑部件，而硬件是一个物理部件。因此，软件具有与硬件完全不同的特点。

**2. 软件的特点**

1）形态特性

软件是一种逻辑实体，不具有具体的物理实体形态特性。软件具有抽象性，可以存储在介质中，但是无法看到软件本身的形态，必须经过观察、分析、思考和判断去了解它的功能、性能及其他的特性。

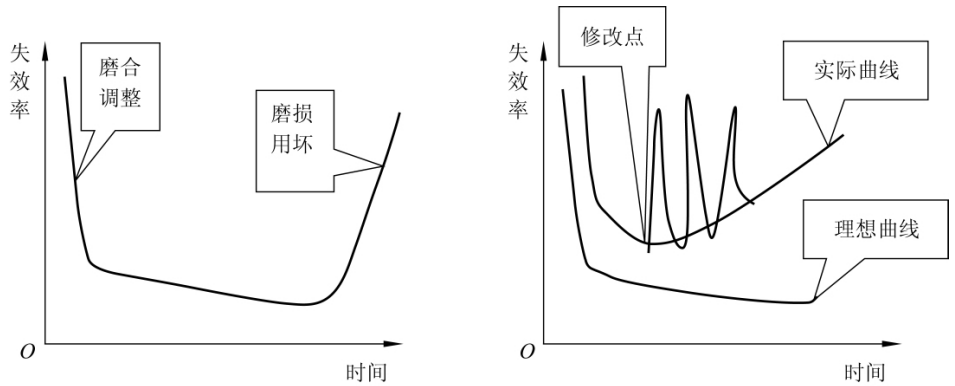
2）生产特性

软件与硬件的生产方式不同。与硬件或传统的制造产品的生产不同，软件一旦设计开发出来，如果需要提供给多个用户，它的复制十分简单，其成本也极为有限，正因为如此，软件产品的生产成本主要是设计开发的成本，同时也不能采用管理制造业生产的办法来解决

软件开发的管理问题。

3)维护特性

软件与硬件的维护不同。硬件是有耗损的,其产生的磨损和老化会导致故障率增加甚至使得硬件损坏,其失效率曲线大致如图 1-1(a)所示。软件不存在磨损和老化的问题,但是却存在退化的问题。在软件的生存期中,为了使它能够克服以前没有发现故障、适应软件和软件环境的变化及用户新的要求,必须要对其进行多次修改,而每次的修改都有可能引入新的错误,导致软件失效率升高,从而使软件退化,其失效率曲线大致如图 1-1(b)所示。




(a) 硬件失效率曲线　　(b) 软件失效率曲线

**图 1-1　失效率曲线**

4)复杂特性

软件的复杂性一方面来自它所反映的实际问题复杂性,另一方面也来自程序结构的复杂性。软件技术的发展明显落后于复杂的软件需求,这个差距日益加大。软件复杂性与时间曲线如图 1-2 所示。

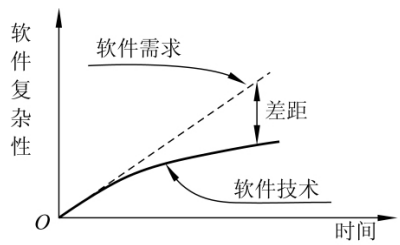



**图 1-2　软件技术的发展落后于需求**

5)智能特性

软件是复杂的智力产品,它的开发凝聚了人们大量的脑力劳动,它本身也体现了知识、实践经验和人类的智慧,具有一定的智能。它可以帮助我们解决复杂的计算、分析、判断和决策问题。

6)质量特性

软件产品的质量控制存在着一些实际困难,难以克服,表现在以下方面。

(1)软件产品的需求在软件开发之初常常是不确切的,也不容易确切地给出,并且需求

还会在开发过程中变更，这就使软件质量控制失去了重要的可参照物。

(2)软件测试技术存在不可克服的局限性。任何测试都只能在极大数量的应用实例数据中选取极为有限的数据，致使我们无法检验大多数实例，也使我们无法得到完全没有缺陷的软件产品。没有在已经长期使用或反复使用的软件中发现问题，并不意味着今后的使用也不会出现问题。

这一特性提醒我们：一定要警惕软件的质量风险，特别是在某些重要的应用场合需要提前准备好应对策略。

7)环境特性

软件的开发和运行都离不开相关的计算机系统环境，包括支持它的开发和运行的相关硬件和软件。软件对计算机系统的环境有着不可摆脱的依赖性。

8)软件的管理特性

上述的几个特点，使得软件的开发管理显得更为重要，也更为独特。这种管理可归结为对大规模知识型工作者的智力劳动管理，其中包括必要的培训、指导、激励、制度化规程的推行、过程的量化分析与监督，以及沟通、协调，甚至是软件文化的建立和实施。

9)软件的废弃特性

等到软件的运行环境变化过大，或是用户提出了更大、更多的需求变更，再对软件实施适应性维护已经不划算，说明该软件已走到它的生存期终点而将废弃(或称为退役)，此时用户应考虑采用新的软件代替。因此，与硬件不同，软件并不是由于“用坏”而被废弃的。

10)应用特性

软件的应用极为广泛，如今它已渗透到国民经济和国防的各个领域，现已成为信息产业、先进制造业和现代服务业的核心，占据了无可取代的地位。

11)软件成本比较高

软件的研制工作需要投入大量、复杂、高强度的脑力劳动，研制成本比较高。在 20 世纪 50 年代末，软件的开销大约占总开销的百分之十几，大部分成本花在硬件上。但今天，这个比例完全颠倒过来，软件的开销远远超过硬件的开销。软、硬件成本随时间变化而变化的比例如图 1-3 所示。

## 1.1.2　软件的发展

软件工程是在克服 20 世纪 60 年代出现的“软件危机”过程中逐渐形成和发展的。在过去的 50 年时间里，软件工程在理论和实践方面都取得了长足的进步。它的发展已经经历了四个重要阶段。

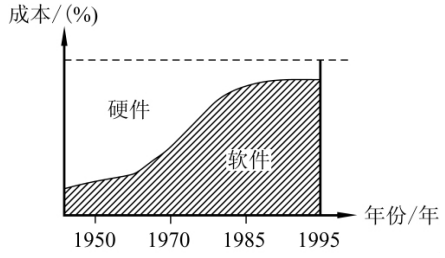



**图 1-3　软、硬件成本比例的变化**

**1.第一代软件技术**

20 世纪 60 年代末，软件生产主要采用“生产作坊”方式。随着软件需求量及规模的迅速扩大，生产作坊方式已不能适应软件生产的需要，出现了所谓的“软件危机”，其主要表现为软件生产效率低下、软件产品质量低劣，在大量劣质的软件涌入市场后不久就在开发过程中夭折。由于“软件危机”的不断扩大，国际

软件界面临着巨大的灾难，软件产业濒临崩溃。

为了克服“软件危机”，1968 年在北大西洋公约组织（NATO）举行的软件可靠性学术会议上第一次提出了“软件工程”的概念，其核心是将软件开发纳入工程化的轨道，以保证软件开发的效率和质量。

此后，逐渐形成了软件工程的基本概念、框架、技术和方法，以结构化开发方法、Jackson 方法等为代表的软件开发方法成为这一阶段的主要开发方法。这一阶段又称为传统的软件工程阶段。

**2. 第二代软件技术**

从 20 世纪 80 年代中期开始，相继推出以 Smalltalk 为代表的面向对象的程序设计语言，面向对象的方法与技术得到迅速发展；从 20 世纪 90 年代起，软件工程研究的重点从程序设计语言逐渐转移到面向对象的分析与设计，演化为一种完整的软件开发方法和系统的技术体系。

20 世纪 90 年代以来，形成了以 Booch 方法、OOSE、OMT 等许多面向对象开发方法的流派，面向对象的方法逐渐成为软件开发的主流。尤其是 1997 年 1 月，综合了各种面向对象方法优点的统一建模语言 UML1.0 的正式推出，使面向对象的方法得到了进一步发展，所以这一阶段又称为对象工程。

**3. 第三代软件技术**

随着软件工程规模和复杂度的不断增大，开发人员也随之增多，开发周期相应延长，加之软件是知识密集型的逻辑思维产品，这些都增加了软件工程管理的难度。人们在软件开发的实践过程中逐渐认识到：提高软件生产效率、保证软件质量的关键是对“软件过程”的控制和管理，是软件开发和维护的管理和支持能力。因此提出了对软件项目管理的计划、组织、成本估算、质量保证、软件配置管理等技术与策略，逐步形成了软件过程工程。

**4. 第四代软件技术**

20 世纪 90 年代起，软件复用和基于构件（Component）的开发方法取得重要进展，软件系统的开发可通过使用现成的可复用软件组装完成，而无须从头开始构造，以此达到提高效率和质量、降低成本的目的，软件复用技术及构件技术的发展，为克服软件危机提供了一条有效途径，这已成为当前软件工程的重要研究方向，这一阶段就称为构件阶段。

## 1.2 软件危机

### 1.2.1 软件危机的主要特征

软件危机的主要特征体现在以下几个方面。

1）软件开发进度难以预测

软件开发过程中的拖延工期现象并不罕见，这种现象降低了软件开发组织的信誉。

2）软件开发成本难以控制

软件开发中投资一再追加，往往实际成本要比预算成本高出一个数量级。而为了赶进

度和节约成本所采取的一些权宜之计往往又损害了软件产品的质量，从而不可避免地引起用户的不满。

3)产品功能难以满足用户需求

开发人员和用户之间很难沟通，矛盾很难统一，往往软件开发人员不能真正了解用户的需求，而用户又不了解计算机求解问题的模式和能力，双方无法用共同熟悉的语言进行交流和描述。在双方不充分了解的情况下，就仓促上阵设计系统，匆忙着手编写程序，这种“闭门造车”的开发方式必然导致最终的产品不符合用户的实际需要。

4)软件产品质量无法保证

系统中的错误很难消除。软件是逻辑产品，质量问题很难以统一的标准来度量，因而造成质量控制困难。软件产品并不是没有错误，而是盲目检测很难发现错误，而隐藏下来的错误往往是造成重大事故的隐患。

5)软件产品难以维护

软件产品本质上是开发人员的代码化的逻辑思维活动的产物，他人难以替代，除非是开发者本人，否则很难及时检测、排除系统故障。为使系统适应新的硬件环境，或根据用户的需要，要在原系统中增加一些新的功能，这又有可能增加系统中的错误。

6)软件缺少适当的文档资料

文档资料是软件必不可少的组成部分。缺乏必要的文档资料或文档资料不合格，将给软件开发和维护带来许多严重的困难和问题。

### 1.2.2 软件危机的具体体现

20世纪60年代末期所发生的软件危机，反映在软件可靠性没有保障、软件维护工作量大、费用不断上升、进度无法预测、成本增长无法控制、程序设计人员无限度增加等各个方面，以致形成人们难以控制软件开发的局面。

软件危机主要表现在如下两个方面：

(1)软件产品质量低劣，甚至在开发过程中就夭折；

(2)软件生产率低，不能满足要求。

软件危机所造成的严重后果已使世界各国的软件产业危机四伏，面临崩溃，因此克服软件危机刻不容缓。自从NATO会议以来，世界各国的软件工作者为克服软件危机进行了许多开创性的工作，在软件工程的理论研究和工程实践两个方面都取得了长足的进步，缓解了软件危机。但距离彻底地克服软件危机这个软件工程的最终目标还任重道远，需要软件工作者付出长期艰苦的努力。

### 1.2.3 软件危机产生的原因

20世纪60年代，计算机已经应用在很多行业，解决问题的规模及难度逐渐增加，由于软件本身的特点及软件开发方法等多方面问题，软件的发展速度远远滞后于硬件的发展速度，不能满足社会日益增长的软件需求。软件开发周期长、成本高、质量差、维护困难，导致20世纪60年代末软件危机的爆发。导致软件危机爆发的原因主要可以概括为以下几点。

1)用户需求不明确

在软件被开发出来之前,用户自己也不清楚软件开发的具体需求;用户对软件开发需求的描述不精确,可能有遗漏、有二义性甚至有错误;在软件开发过程中,用户还会提出修改软件开发功能、界面、支撑环境等方面的要求;软件开发人员对用户需求的理解与用户的本来愿望有差异。

2)缺乏正确的理论指导

由于软件开发不同于大多数其他工业产品,其开发过程是复杂的逻辑思维过程,其产品很大程度上依赖于开发人员高度的智力投入。过分地依靠程序设计人员在软件开发过程中的技巧和创造性,加剧了软件开发产品的个性化,这也是发生软件危机的一个重要原因。

3)软件开发规模越来越大

随着软件开发应用范围的扩大,软件开发规模越来越大。大型软件开发项目需要组织一定的人力共同完成,然而多数管理人员缺乏开发大型软件系统的经验,多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确,有时还会产生误解。软件开发人员不能有效地、独立自主地处理大型软件开发的全部关系和各个分支,因此容易产生疏漏和错误。

4)软件开发复杂度越来越高

软件开发不仅仅是在规模上快速地发展扩大,而且其复杂性也急剧地增加。软件开发产品的特殊性和人类智力的局限性,导致人们无力处理“复杂问题”。所谓“复杂问题”的概念是相对的,一旦人们采用的先进组织形式、开发方法和工具提高了软件开发效率和能力,新的、更大的、更复杂的问题又会摆在人们面前。

### 1.2.4 软件危机的解决途径

1968 年,计算机科学家在德国第一次讨论软件危机问题,并正式提出“软件工程”一词,从此一门新兴的工程学科为研究和克服软件危机应运而生。作为一个新兴的工程学科,软件工程学主要研究软件产生的客观规律性,建立与系统化软件生产有关的概念、原则、方法、技术和工具,指导和支持软件系统的生产活动,以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。软件工程学从硬件工程和其他人类工程中吸收了许多成功的经验,明确提出了软件生命周期的模型,发展了许多软件开发与维护阶段适用的技术和方法,并应用于软件工程实践,取得了良好的效果。

在软件开发过程中,人们开始研制和使用软件工具,用于辅助进行软件项目管理与技术生产,人们还将软件生命周期各阶段使用的软件工具有机地集合成一个整体,形成能够连续支持软件开发与维护全过程的集成化软件支撑环境,以期从管理和技术两方面解决软件危机问题。

## 1.3 软件工程

### 1.3.1 软件工程的定义

软件工程是一门指导计算机软件开发和维护的工程学科,是一门边缘学科,涉及计算机

科学、工程科学、管理科学、数学等多学科，研究的范围广，主要研究如何应用软件开发的科学理论和工程技术来指导大型软件系统的开发。

虽然对软件工程有着众多的定义，但是其基本思想都是强调在软件开发过程中应用工程化的重要性。

例如，1983年，IEEE(电气和电子工程师协会)所下的定义是：软件工程是开发、运行、维护和修复软件的系统方法。1990年，IEEE又将定义更改为：对软件开发、运行、维护的系统化的、有规范的、可定量的方法之应用，即是对软件的工程化应用。

2004年，IEEE/ACM联合发布的CCSE2004报告强调了对软件工程的新定义，即软件工程是“以系统的、科学的、定量的途径，把工程应用于软件的开发、运行和维护；同时，开展对上述过程中各种方法和途径的研究”。这也是目前一种比较广泛认可的定义。

从软件工程的定义可见，软件工程是一门指导软件系统开发的工程学科，它以计算机理论及其他相关学科的理论为指导，采用工程化的概念、原理、技术和方法进行软件的开发和维护，把实践证明的、科学的管理措施与最先进的技术方法结合起来。软件工程研究的目标是“以较少的投资获取高质量的软件”。它包括3个要素：方法、工具和过程。

(1)软件工程方法为软件开发提供了“如何做”的技术，是指导开发软件的某种标准规范。它包括多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试及维护等。软件工程方法常采用某种特殊的语言或图形的表达方法及一套质量保证标准。

(2)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。

(3)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、为保证质量和协调变化所需的管理及软件开发各个阶段完成的“里程碑”。

有关软件工程3个要素的相关知识我们将在后续章节详细介绍。

### 1.3.2 软件工程的背景和历史

为了克服软件危机，1968年10月，NATO召开的计算机科学会议上，Fritz Bauer首次提出“软件工程”的概念，企图将工程化方法应用于软件开发上。

许多计算机和软件科学家尝试把其他工程领域中行之有效的工程学知识运用到软件开发工作中来。经过不断实践和总结，最后得出一个结论：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条主要道路。

虽然软件工程的概念提出已有40多年，但到目前为止，软件工程概念的定义并没有得到认可。在NATO会议上，Fritz Bauer对软件工程的定义是：“为了经济地获得可靠的和能在实际机器上高效运行的软件，而建立和使用的健全的工作原则。”除了这个定义，还有几种比较有代表性的定义。

B. W. Boehm给出的定义是：“运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料。”此处，“设计”一词广义上应理解为包括软件的需求分析和对软件进行修改时所进行的再设计活动。