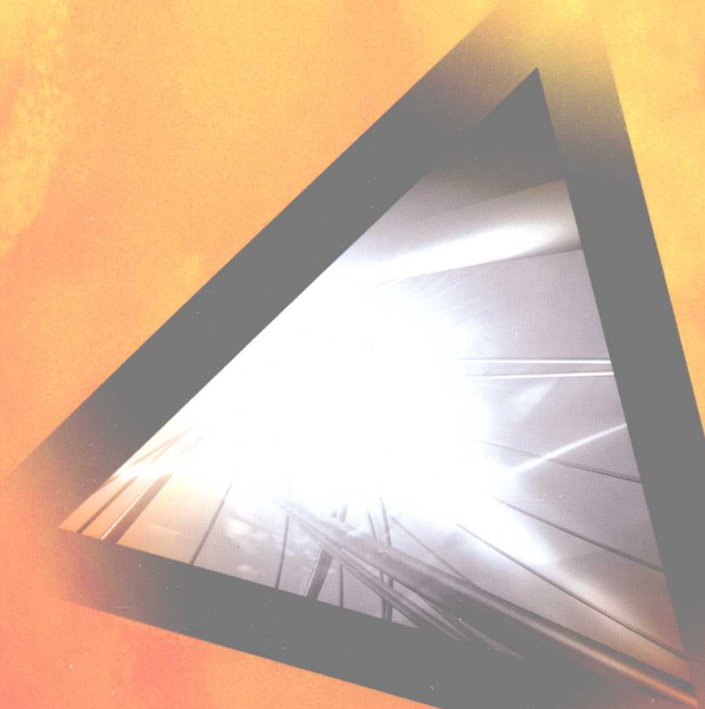
赠电子课件

职业教育“十一五”规划教材

计算机类专业

软件工程基础

李国彬　主编



实例引导

机械工业出版社
CHINA MACHINE PRESS

职业教育“十一五”规划教材
——计算机类专业

软件工程基础

主　编　李国彬
参　编　王洪香　张　莹
　　包　剑　郑艳杰
主　审　孙心义

机械工业出版社

软件工程已成为计算机科学领域中的一个重要学科。本书着重从实用角度讲述软件工程的基本概念、原理、方法和工具，系统地介绍目前较成熟的、广泛使用的软件工程技术。

本书内容包括：软件工程概述，可行性研究，需求分析，系统设计（概要设计、详细设计），面向对象设计方法和 UML 的使用，编码，质量保证与软件测试、软件维护以及软件工程管理技术等。每章都有小结，并配有习题供读者练习、提高。

本书可作为职业院校计算机专业及其相关专业的教材，也可供从事计算机软件开发及应用的广大科技人员参考。

本书赠送电子课件，方便教学，需要者请登录www. cmpedu. com 免费注册后下载，或联系责任编辑（010-88379194）索取。

图书在版编目(CIP)数据

软件工程基础/李国彬主编 .—北京：机械工业出版社，2008.6
职业教育“十一五”规划教材．计算机类专业
ISBN 978-7-111-24488-2

Ⅰ．软… Ⅱ．李… Ⅲ．软件工程—专业学校—教材 Ⅳ．TP311.5

中国版本图书馆 CIP 数据核字(2008)第095473号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：王玉鑫 孔熹峻 责任编辑：孔熹峻 版式设计：霍永明
责任校对：李汝庚 封面设计：马精明 责任印制：洪汉军
北京振兴源印务有限公司印刷厂印刷
2008 年 8 月第 1 版第 1 次印刷
184mm × 260mm · 11.5 印张 · 284 千字
0001 – 4000 册
标准书号：ISBN 978-7-111-24488-2
定价：19.00 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) 88379194
封面无防伪标均为盗版

前　言

随着计算机科学技术的快速发展和计算机应用领域的不断扩大，软件的复杂程度不断增加，对软件的规范化、可维护性和可重用性的要求也越来越高，因而，软件工程技术变得日益重要。

借鉴职业教育教学经验，为适应当前教学的需要，按照职业院校的教学要求和教材特点，编写了这本教材。本书编写时除保证应用性强外，还特别注意保持语言精炼、易于理解、可读性好的特点。为方便读者掌握重点和巩固学习内容，每章后都有小结和习题，供读者复习选用。

本书从实用角度介绍了软件工程的基础知识和软件工程技术方法。全书共分10章。第1章软件工程概述，主要介绍了软件工程基本概念以及软件开发模型；第2章软件的可行性研究，主要介绍了可行性研究的任务、系统流程图、软件的成本-效益分析；第3章软件需求分析，主要介绍了软件需求分析的步骤、原则和方法，重点介绍了面向数据流的需求分析方法；第4章软件的概要设计，主要介绍了软件概要设计任务、设计过程、设计原则，重点介绍了面向数据结构的分析设计方法等；第5章软件的详细设计，主要讲述了软件详细设计的任务和方法，重点介绍了详细设计工具；第6章面向对象的分析和设计方法，主要讲述了面向对象方法学、面向对象的基本特征、统一建模语言UML等，重点介绍了面向对象的分析和设计；第7章编码，主要讲述了程序设计语言的分类、特点、选择、编程风格、程序设计效率和编码安全等；第8章软件质量与软件测试，主要讲述了软件质量、软件质量保证策略、软件测试方法和步骤、软件测试中的可靠性分析，重点介绍了软件测试用例技术；第9章软件维护，主要讲述了软件维护的基本概念，软件维护的过程，软件维护的副作用，软件的可维护性等；第10章项目计划与管理，主要讲述了软件项目特点、软件管理的功能、软件配置管理的过程、开发进度的安排、软件项目的组织等。

本书主要作为职业院校计算机专业及其相关专业的教材，同时也可供从事计算机软件开发及应用的广大科技人员作为工具参考书。

本书由李国彬主编，孙心义担任主审，王洪香、张莹、包剑、郑艳杰参与了编写。其中第1、2、9章由李国彬编写，第7章由王洪香编写，第5、6章由张莹编写，第8、10章由包剑编写，第3、4章由郑艳杰编写。在本书的编写过程中，辽宁工程技术大学职业技术学院计算机系的领导给予了大力的支持，黄金波、沈淑清、冀明等老师为本书提出了许多宝贵意见，阜新市五星电子商务中心为本书的编写提供了方便，在此一并致以谢意！

由于作者水平有限，加之编写时间仓促，书中疏漏和欠妥之处在所难免，欢迎广大师生和读者提出批评指正。

编　者

目　　录

第 1 章　软件工程概述

软件工程学是引导计算机软件产业发展的一个标志性学科。随着微电子技术的飞速发展，计算机硬件设备的价格急剧下降，功能不断提高，但软件生产的成本却居高不下，导致软件系统的开发成本逐年上升，产品质量难以控制，成品不便于维护，生产率也远远跟不上实际需求。因此，软件的生产与维护，成为了限制计算机应用系统继续快速发展的关键因素。西方计算机科学家把在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”，并从 20 世纪 60 年代末开始认真研究解决软件危机的方法，从而逐步形成了一门新兴的学科——计算机软件工程学。

1.1　软件

1946 年，世界上诞生了第一台电子计算机，当时并没有软件的概念，更没有软件工程的概念，程序的编写多是由经过训练的数学家和电子工程师来完成。进入 20 世纪 60 年代，美国大学里开始出现计算机专业，软件产业从此开始起步。经过多年的发展，软件产业已经成为推动人类社会发展的龙头产业，并造就了一批百万、亿万富翁。随着信息产业的不断发展，软件产业对人类社会产生的影响将越来越重要。

1.1.1　软件的发展

随着计算机硬件性能的极大提高和计算机体系结构的不断变化，计算机软件系统更加成熟且更为复杂，从而促使计算机软件的角色发生了巨大的变化，人们对软件的认识也因此经历了一个由浅到深的过程，其发展过程大致可以分为如图 1-1 所示的 4 个阶段。

1. 程序设计阶段

20 世纪 50 年代初期至 60 年代初期的十余年，是计算机系统开发的初期阶段，称为程序设计阶段。这个阶段硬件已经通用化，而软件几乎都是为每个具体应用专门编写的，编写者和使用者往往是同一个或同一组人。在这些个体化的程序设计环境中，人们的头脑中根本就没有软件设计这个概念，除了最后的程序清单外，没有其他任何文档资料保存下来。

在程序设计阶段，人们认为计算机的主要用途就是快速运算，程序的编写非常简单，不存在什么系统化的方法，对开发没有任何管理，程序的质量完全依赖于程序员个人的技巧。而且，为了提高计算机的使用效率，初期开发的计算机系统采用了批处理技术，但事与愿违的是，这种做法使得程序的设计、调试和修改都变得非常麻烦。

2. 程序系统阶段

从 20 世纪 60 年代中期到 70 年代末期的十多年间，多道程序系统、多用户系统引入了人机交互的新概念，交互技术打开了计算机应用和硬件与软件配合的新局面，出现了实时系统和第一代数据库管理系统。另一方面，这个阶段出现了软件产品和“软件作坊”的概念，设计人员开发软件不再像早期那样只为自己的研究工作需要，而是为了用户更好地使用计算

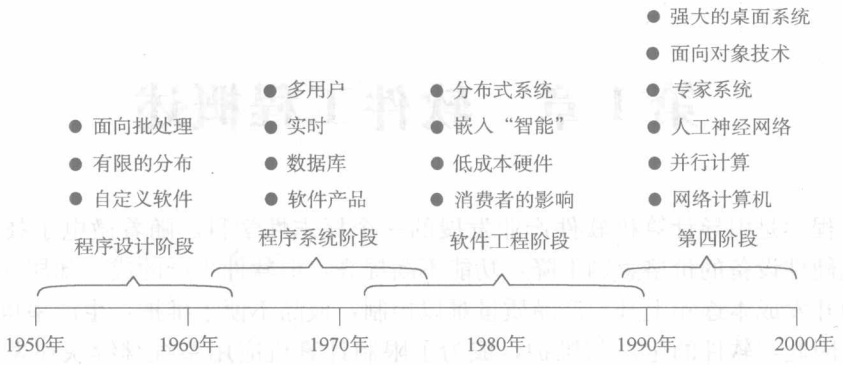


图 1-1　计算机软件发展的 4 个阶段

机，但这种“软件作坊”仍然沿用早期形成的个体式的软件开发方法，这个阶段被称为程序系统阶段。

随着计算机应用的日益普及，软件需求量急剧膨胀，程序在运行时发现的错误必须设法得以纠正；用户有了新需求时，程序必须进行相应的修改；硬件或操作系统更新时，程序可能又要进行修改以适应新的环境。因此，软件的维护工作以惊人的比例耗费资源。更严重的是，程序设计的个体化和作坊式特性使得软件无人维护而最终成为不可维护，从而出现了早期的软件危机，于是人们开始寻求采用软件工程的方法来解决软件危机问题。

3. 软件工程阶段

从 20 世纪 70 年代中期至 80 年代末期属于软件工程阶段。在这个阶段，计算机系统的复杂性极大提高，网络的迅速发展对软件开发提出了更高的要求，硬件的发展速度已经超出了人们的想象，特别是微处理器的出现和广泛应用，导致了一系列的智能产品的问世。由于硬件价格不断下降，使人们对软件的需求速度迅速上升，软件的价格随之急剧攀升，导致了软件危机的加剧，致使更多的科学家不得不着手研究软件工程学的理论、方法和时限等一系列问题，软件开发技术的标准和度量问题也必须一并着手解决。

4. 第四阶段

从 20 世纪 80 年代末期开始至今为第四阶段，目前还没有很好的名字来定义这个阶段。在这个阶段里，计算机系统发展已经不再着重于单台计算机系统和程序，而是面向计算机和软件的综合。强大的桌面系统和计算机网络迅速发展，计算机体系结构由中央主机控制方式变为客户机/服务器方式，专家系统和人工智能软件终于走出实验室进入了实际应用，虚拟现实和多媒体系统改变了与最终用户的通信方式，出现了并行计算和网络计算的研究，面向对象技术在许多领域迅速取代了传统软件开发方法。

1.1.2　软件的定义及其特点

1. 软件的定义

软件是一种产品，同时又是开发和运行产品的载体。作为一种产品，它表达了由计算机硬件体现的计算潜能。作为开发运行产品的载体，软件是计算机工作和信息通信的基础，也是创建和控制其他程序的基础。软件从个性化的程序变为工程化的产品，人们对软件的看法发生了根本性的变化。软件（software）是计算机系统中与硬件（hardware）相互依存的另

一部分，是包括程序（program）、相关数据（data）及其说明文档（document）的一个完整集合。其中，程序是按照事先设计的功能和性能要求执行的指令序列；数据是程序能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各种图文资料。可以看出，软件定义由以下三部分组成。

1）在运行中能提供所希望的功能和性能的指令集（即程序）。

2）使程序能够正确运行的数据结构。

3）描述程序研制过程、方法所用的文档。

在软件的发展过程中，软件的需求成为软件发展的动力，软件的开发从自给自足模式发展为在市场中流通以满足广大用户的需要。软件工作的考虑范围也发生了很大变化，人们不再只顾及程序的编写，而是涉及软件的整个生存周期。

2. 软件的特点

计算机的硬件是一个物理部件，而计算机的软件却是一个逻辑部件。因此，同硬件相比，软件具有以下独特的特点。

（1）软件是一种逻辑产品　与物质产品不同，软件产品是看不见摸不着的，因而具有无形性，是脑力劳动的结晶，是以程序和文档的形式出现的，保存在计算机存储器和光盘介质中，通过计算机的执行才能体现其功能和作用。

（2）软件产品没有生产过程　软件产品的生产主要是研制，软件产品的成本主要体现在软件的开发和研制上。软件一旦研制开发成功后，通过复制就产生了大量软件产品。

（3）软件在使用过程中没有磨损、老化的问题　软件在使用过程中不会因为磨损而老化，但会为了适应硬件、软件环境以及需求的变化而进行修改，而这些修改又不可避免地引入错误，导致软件失效率升高，从而使得软件退化，如图 1-2 所示。当修改的成本变得难以接受时，软件就被抛弃。

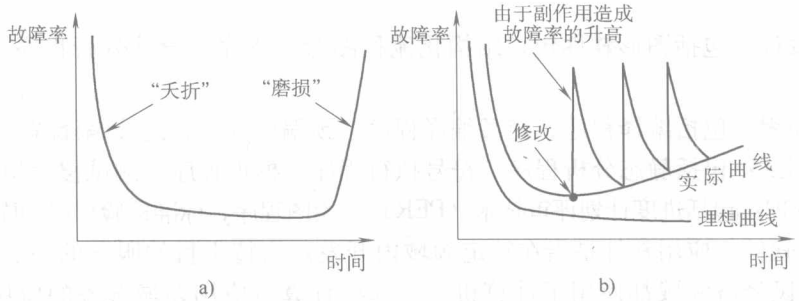


图 1-2　故障率曲线

a）硬件的故障曲线　b）软件的理想故障曲线和实际故障曲线

（4）软件的运行依赖于硬件和环境　软件对硬件和环境有着不同程度的依赖性，这直接导致了软件移植的问题。

（5）软件的开发是一种智力活动　软件的开发至今尚未完全摆脱手工作坊式的脑力劳动的开发方式，因此生产效率不高，且大部分产品是“定做”的。

（6）软件是复杂度最高的工业产品　软件是人类有史以来生产的复杂度最高的工业产品，而且以后会更加复杂。软件开发常常会涉及其他众多领域的专门知识，才能使得软件产

品最终适合人类社会的各行各业，因此对软件工程师提出了很高的要求。

（7）软件的成本相当昂贵　软件开发需要投入大量、高强度的脑力劳动，成本非常高，风险也大。现在，软件的开销已大大超过了硬件的开销。

（8）软件工作牵涉很多社会因素　许多软件的开发和运行涉及机构、体制和管理方式等问题，还会涉及人们的观念和心理等因素。这些人的因素，常常成为软件开发的困难所在，直接影响到项目的成败。

1.1.3　软件的分类

在学习和工作中，我们经常接触到各种各样的软件。随着软件复杂性的增加，给出一个科学、统一而且严格的软件分类标准既困难，也不现实，但根据不同类型的工程对象采用不同的开发和维护方法对软件的类型进行必要的划分，是很有必要的。因此，从不同的角度对计算机软件进行适当的分类还是很有价值的。

1. 基于软件功能的划分

按照软件的功能，可将软件划分为如下几种类型。

（1）系统软件　系统软件是与计算机硬件紧密配合，以使计算机各个部件与相关软件及数据协调、高效工作的软件。例如，操作系统、数据库管理系统等。系统软件在工作时频繁地与硬件交互，以便为用户服务，共享系统资源，在这中间伴随着复杂的进程管理和数据结构的处理。系统软件是计算机系统必不可少的重要组成部分。

（2）支撑软件　支撑软件是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和帮助管理人员控制开发进程的工具，又可分为以下几类。

① 一般类型：包括文本编辑程序、文件格式化程序、程序库系统等。

② 支持需求分析：包括问题描述语言、问题描述分析器、关系数据库系统、一致性检验程序等。

③ 支持设计：包括图形软件包、结构化流程图绘图程序、设计分析程序、程序结构图编辑程序等。

④ 支持编程：包括编译程序、交叉编译程序、预编译程序、连接编译程序等。

⑤ 支持测试：包括静态分析程序、符号执行程序、模拟程序、测试覆盖检验程序等。

⑥ 支持管理：包括进度计划评审技术（PERT）、绘图程序、标准检验程序和库管理程序等。

（3）应用软件　应用软件是指在特定领域内开发，为特定目的服务的一类软件。现在，几乎所有的国民经济领域都使用了计算机，为这些计算机应用领域服务的应用软件种类繁多。其中商业数据处理软件是占比例最大的一类，工程与科学计算软件大多属于数值计算问题。应用软件还包括计算机辅助设计/制造（CAD/CAM）、系统仿真、智能产品嵌入软件（如汽车油耗控制、仪表盘数字显示、刹车系统），以及人工智能软件（如专家系统、模式识别）等。此外，在事务管理、办公自动化、中文信息处理、计算机辅助教学（CAI）等方面的软件也得到了迅速发展，产生了惊人的生产效率和巨大的经济效益。

2. 基于软件用途的划分

按照软件的用途，软件可以大致划分成如下几种类型。

（1）系统软件　就一般情况来说，系统软件是为其他软件服务的软件。系统软件与计算机硬件交互频繁，处理大量的确定或不确定的复杂数据，往往需要具有多用户支持、资源精

细调度、并发操作管理、多种外部设备接口支持等功能。

（2）实时软件　管理、分析、控制现实世界中所发生的事件的软件称为实时软件。它一般有数据采集、数据分析、输出控制等三方面的功能。实时软件需要保持一个现实任务可以接受的响应时间，即必须保证能够在严格限定的时间范围内对输入做出响应。

（3）商业管理软件　商业信息处理是最大的软件应用领域，包括常规的数据处理软件和一些交互式的计算处理软件（如 POS 软件）。它的基本功能是将已有的数据重新构造，变换成一种可以辅助商业操作和管理决策的形式。在这个过程中，几乎都要涉及对于大型数据库的访问。各类管理信息系统（MIS）、企业资源计划（ERP）、客户关系管理（CRM）等都是典型的商业管理软件。

（4）工程与科学计算软件　此类软件的特征是要实现特定的“数值分析”算法。例如，离散傅里叶变换、有限元分析、演化计算等。CAD/CAM 软件一般也可以归属到这一类型中来。

（5）嵌入式软件　驻留在专用智能产品的内存中，用于控制这些产品进行正常工作，完成很有限、很专业的功能的软件。例如，各类智能检测仪表、数码相机、移动电话、微波炉等智能产品都必须在嵌入式软件的支持下才能正常工作。

（6）人工智能软件　利用非数值算法去解决复杂问题的软件。各类专家系统、模式识别软件、人工神经网络软件都属于人工智能软件。

（7）个人计算机软件　文字处理系统、电子表格、游戏娱乐软件等。

3. 基于软件工作方式的划分

按照软件工作方式，可将软件划分为以下四种类型。

（1）实时处理软件　实时处理软件是指在事件或数据产生时，立即处理，并及时反馈信号，控制需要监测和控制过程的软件。主要包括数据采集、分析、输出三部分。其处理时间是应严格限定的，如果在任何时间超出了这一限制，都将造成事故。

（2）分时软件　分时软件是指允许多个联机用户同时使用计算机的软件。系统把处理机时间轮流分配给各联机用户，使各用户都感到只有自己在使用计算机。

（3）交互式软件　交互式软件是指能实现人机通信的软件。这类软件接收用户给出的信息，但在时间上没有严格的限定，这种工作方式给予用户很大的灵活性。

（4）批处理软件　批处理软件是指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. 基于软件规模的划分

按照软件规模，主要是从软件项目的规模大小、时间长短和参加人数多少几个方面来划分，可将软件分成如下几种类型。

（1）微型软件　微型软件是指一个人在几天之内完成的、程序不超过 500 行语句且仅供个人专用的软件。通常这类软件没有必要作严格的分析，也不必有完整的设计、测试资料。

（2）小型软件　小型软件是指一个人在半年之内可以完成的 2000 行以内的程序。这种程序通常没有与其他程序的接口，但需要按一定的标准化技术、正规的资料书写以及定期的系统审查，只是没有大型软件那样严格。

（3）中型软件　中型软件是指 5 个人以内在一年多时间里完成的 5000 到 50000 行的程序。中型软件开始出现了软件人员之间、软件人员与用户之间的联系和协调的配合关系问

题，因而计划、资料书写以及技术审查比较严格。在开发中使用软件工程方法是完全必要的，这对提高软件产品质量和程序开发人员的工作效率起着重要的作用。

（4）大型软件　大型软件是指5至10个人在两年多的时间里完成的5万到10万行的程序，对参加工作的软件人员需要进行二级管理。对于这样规模的软件，采用统一的标准、实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性，往往在开发过程中会出现一些事先难以估计的不测事件。

（5）巨大型软件　巨大型软件是指100至1000人在四五年时间里可以完成的具有100万行的程序。这种大型项目可能会划分成若干个子项目，每一个子项目都是一个大型软件。子项目之间具有复杂的接口。例如，实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统。很显然，如果这类问题没有软件工程方法的支持，它的开发工作是不可想象的。

（6）超大型软件　超大型软件是指2000至5000人在10年内可以完成的具有1000万行以内的程序。这类软件很少见，往往用于军事指挥、弹道导弹防御系统等。

可以看出，规模大、时间长、多人参加的软件项目，其开发工作必须有软件工程的知识做指导；而规模小、时间短、参加人员少的软件项目的开发也得从软件工程的概念出发，遵循软件工程开发规范，其基本原则是一样的，见表1-1。

表1-1　基于软件规模的分类

分类	参加人数	开发期限	程序规模/源程序行数	特　征
微型	1	1～4周	500以下	不必有严格的设计和测试文档
小型	1～2	1～6月	1k～2k	通常没有与其他程序的接口
中型	3～5	1～2年	5k～50k	需要有严格的文档和设计规范
大型	5～20	2～3年	50k～100k	需要按照软件工程方法进行管理
巨大型	100～1000	4～5年	1M（1000k）	必须按照软件工程开发，有严格的质量管理措施
超大型	2000～5000	5～10年	1M～10M	同上

此外，还有基于软件失效的影响进行划分的方法、基于软件服务对象的范围进行划分的方法等，在此不再赘述。

1.2　软件工程的产生与发展

1.2.1　软件危机

1. 软件危机及其表现

20世纪60年代，很多的软件项目的开发时间大大超出了规划的时间，一些项目导致了财产的流失，甚至导致了人员伤亡。同时，软件开发人员也发现软件开发的难度越来越大。

软件的错误可能会导致巨大的财产损失。2002年12月，欧洲阿里亚娜火箭的爆炸就是一个最为惨痛的教训。而且，由于计算机软件被广泛应用于包括医院等与生命息息相关的行业，也使得软件的错误有可能会导致人员的伤亡。在工业上，某些嵌入式系统导致机器的不正常运转，从而使工作人员陷入险境。

现代计算机应用系统中，软件的地位日益重要和突出。如何满足日益增长的软件需求，如何维护应用中的大量已有软件，已经成为了计算机应用系统进一步发展的瓶颈。1968 年，北大西洋公约组织的计算机科学家们在前联邦德国召开的国际会议上讨论了软件危机问题，同时提出了“软件工程”这个名词，标志着一门新的工程学科的正式诞生。

简单地说，软件危机就是指在软件开发和软件维护过程中所存在的一系列严重问题。具体地说，软件危机具有如下一些表现。

1）软件开发没有真正的计划性，对软件开发进度和软件开发成本的估算常常很不准确，计划的制定带有很大的盲目性，因此工期超出、成本失控的现象经常困扰着软件开发者。

2）对于软件需求信息的获取常常不充分，软件产品往往不能真正地满足用户的实际需求。

3）缺乏良好的软件质量评测手段，从而导致软件产品的质量常常得不到保证。

4）对于软件的可理解性、可维护性认识不够；软件的可复用性、可维护性不如人意。有些软件因为过于“个性化”，甚至是难以理解的，更谈不上进行维护。缺乏可复用性引起的大量重复性劳动，极大地降低了软件的开发效率。

5）软件开发过程没有实现“规范化”，缺乏必要的文档资料，或者文档资料不合格、不准确，难以进行专业维护。

6）软件开发的人力成本持续上升，如美国在 1995 年的软件开发成本已经占到了计算机系统成本的 90%。

7）缺乏自动化的软件开发技术，软件开发的生产率依然低下，远远满足不了急剧增长的软件需求。

软件危机曾经是历史上的阴影，目前软件工程界也仍然在一定程度上受到它的影响。软件工程概念的提出，正是为了克服软件危机。自 1968 年以来，随着软件工程学的不断发展，软件危机得到了一定程度的遏制，但还远远没有被彻底解决。

20 世纪 90 年代中期，美国商用软件产业的情况如下：1995 年，美国公司取消了 810 亿美元的软件项目；在所考察的软件项目中，完成前就取消了其中的 31%；53%的软件项目进度拖延，通常拖延的时间超过预定工期 50%以上；只有 9%的大型软件项目能够及时交付且费用不超支（对中型和小型软件公司来说这一数据为 16%）。

从上面的统计数据不难看出，在软件开发过程中，危机的影响依然存在。再回顾一下世纪之交时，为防治软件“千年虫”问题所投入的巨大人力与物力，也反映了软件维护工作的艰巨与困难。

2. 软件危机产生的原因

软件危机的存在是不争的事实。产生软件危机的原因可以归纳为主、客观两个方面。从客观上来看，软件不同于硬件，它的生产过程和产品都具有明显的“不可视”特征，这就导致了在完成编码并且上机运行之前，对于软件开发过程的进展情况较难衡量，软件产品的质量也较难进行先期评价，因此，对于开发软件的过程进行管理和控制比较困难。在软件工程的早期，制定详细的开发计划并且进行全程跟踪调控，对于所有的阶段产品和阶段工作进展进行技术审查和管理复审，可望在一定程度上克服“开发过程不可视”造成的消极影响。

此外，软件运行过程中如果发现了错误，那么必然是在开发时期（分析、设计、编码过程）引入的在检测过程中没有能够检查出来的故障。对于此类故障的维护，通常意味着要修

改早期的分析结果、设计结果并调整编码。由于软件产品的不可视特征，维护过程不像硬件产品维护时只要简单地更换损坏部件那样容易，这在客观上造成了软件难以维护的结果。利用足够的文档资料使不可视的产品可视化，有助于提升软件产品的可理解性。通过和用户多次交流，在所要开发的软件必须“做什么”方面和用户达成一致（当然在开发过程中也允许在严格的控制下进行需求变更）。

软件被定义之后，进入开发阶段，主要对软件的体系架构、数据结构和主要算法进行设计和编码实现。对于编码结果，还要按照规范进行测试后，才能最终交付使用。如前所述，在开发阶段也可能对于此前不够准确的软件定义结果进行调整。统计数据表明，在典型的软件工程过程中，编码工作量大约只占软件开发全部工作量的 15%～20%。软件的运行与维护阶段在软件生存周期中占据的比例最大。在软件运行过程中，分析和设计阶段的一些遗留缺陷可能会逐步暴露；运行环境的演变也会对运行中的软件提出变更要求；用户新需求的提出则常常要求扩充现有软件的功能或者改进其性能，所有这些要求与问题都必须通过“软件维护”工作去解决。

在维护过程中，必须注意保持所有软件产品之间的一致性。针对不同的需求，维护工作一般可以分为纠错性维护、适应性维护、扩充性维护和预防性维护等不同类型。作为软件，应当有一个完整的配置。Boehm（美国著名的软件工程专家，加州州立大学教授）指出，“软件是程序以及开发、使用、维护程序所需要的所有文档”。所以，软件产品除包括程序之外，还应当包括完整、准确、翔实的文档资料。主要的文档应当包括“需求规格说明书”、“体系结构设计说明书”、“详细设计说明书”、“安装手册”、“操作手册”、“系统管理员手册”等。缺乏必要的配置文档，将严重影响软件的可理解性，从而给软件的维护造成严重障碍。做好包括项目策划、可行性研究、需求分析三项内容的软件定义工作，是提高软件质量、降低软件成本、保证开发进度的关键环节。

值得注意的重要问题是，在软件开发的不同阶段进行修改所付出的代价是极其不同的。在早期引入变动，涉及的面比较小，因而代价也比较低；在开发的中期，因为许多配置项（被标识的工作产品）已经完成，所以引入一个变动，就要对它所涉及的所有已经完成的配置项进行变更，不仅工作量大，而且逻辑上也更复杂，因此付出的代价剧增；如果在软件“已经完成”时再引入变更，更是要付出高得多的代价。

3. 解决软件危机的途径

可以借鉴其他工程领域的成功经验，基于软件危机产生的主、客观原因，从软件工程技术和软件工程管理两方面来采取措施，防范软件危机的发生。软件开发不是某种个体劳动的神秘技巧，而应当是一种组织良好、管理严密，分析、设计、编码、测试、维护等各类人员协同配合、共同完成的工程项目。在软件开发过程中，必须充分吸收和借鉴人类长期以来从事各种工程项目所积累的行之有效的原理、概念、技术和方法，特别要注意吸收几十年来在计算机硬件研究和开发中积累的经验、教训。

从管理层面上考虑，应当注意推广和使用在实践中总结出来的成功的技术和方法，并且探索更好的、更有效的技术和方法，注意积累软件开发过程中的经验数据财富，逐步消除在计算机系统早期发展阶段形成的一些错误概念和做法。建立适合于本组织的软件工程规范；制定软件开发中各个工作环节的流程文件、工作指南和阶段工作产品模板；实施针对软件开发全过程的计划跟踪和品质管理活动；为每一项工程开发活动配置管理库；实施严格的产品

基线管理并建立组织的软件过程数据库和软件财富库；为各类员工及时提供必要的培训等，都是加强软件开发活动管理工作的有效手段。

从技术角度考虑，应当开发和使用更好的软件开发工具，提高软件开发效率和开发工作过程的规范化程度。在计算机软件开发的各个阶段，都有大量的繁琐重复的工作要做，在适当的软件工具的辅助下，开发人员可以把这类工作做得既快又好。目前广为使用的统一建模语言（UML）、各种配置管理工具、缺陷管理工具和自动测试工具都在软件工程活动中发挥了很好的作用。计算机辅助软件工程（CASE）更是目前备受重视的旨在实现软件开发自动化的新领域。

1.2.2 软件工程

1. 软件工程的定义

1968年秋季，NATO（北约）的科技委员会召集了近50名一流的编程人员、计算机科学家和工业界巨头，讨论和制定摆脱“软件危机”的对策。在会议上第一次提出了软件工程（software engineering）这个概念。当时提出这个概念的Fritz Bauer的主要思路是想将系统工程的原理应用到软件的开发和维护中。

软件工程是一门研究如何用系统化、规范化、数量化等工程原则和方法去进行软件的开发和维护的学科。可以定义为：软件工程是一类设计软件的工程，应用计算机科学、数学、工程科学及管理科学等原理，借鉴传统工程的原则、方法创建软件以达到提高质量、降低成本的目的。其中：计算机科学、数学用于构建模型与算法；工程科学用于制定规范、设计规范、评估成本及确定权衡；管理科学用于计划、资源、质量、成本等管理。软件工程学是一门指导计算机软件开发和维护的科学。

软件工程包括两方面内容：软件开发技术和软件项目管理。其中，软件开发技术包括软件开发方法学、软件工具和软件工程环境；软件项目管理包括软件度量、项目估算、进度控制、人员组织、配置管理、项目计划等。

统计数据表明，大多数软件开发项目的失败，并不是由于软件开发技术方面的原因，而是由于不适当的管理造成的。遗憾的是，尽管人们对软件项目管理重要性的认识有所提高，但在软件管理方面的进步远比在设计方法学和实现方法学上的进步小，至今还提不出一套管理软件开发的通用指导原则。

2. 软件工程的发展

在软件的长期发展中，人们针对软件危机的表现和原因，经过不断的实践和总结，越来越清楚地认识到：按照工程化的原则和方法组织软件开发工作，是摆脱软件危机的一个主要出路。今天，尽管“软件危机”并未被彻底解决，但软件工程三十多年的发展仍可以说是硕果累累。

软件工程方法为软件开发提供了“如何作某项工作”的技术指南，它包括了多方面的任务，例如，项目策划和估算方法、软件需求分析方法、体系结构的设计方法、详细设计方法、软件测试方法等，使得整个开发过程的每一种阶段任务都能够“有章可循”。

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。目前，这样的工具已经有许多种，而且已经有人把诸多软件工程工具集成起来，使得一种工具产生的信息可以为其他工具所使用，形成了一种称之为计算机辅助软件工程（CASE）的软件开发支撑环

境。CASE 把各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来，形成了一个完整的软件工程环境。

软件工程中的“过程”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，可以将软件工程过程理解为软件工程的工艺路线。过程定义了各种方法使用的顺序、各阶段要求交付的文档资料、为保证质量和控制软件变更所需要的管理环节以及软件开发各个阶段完成的标志。

针对软件工程的基本构成，有许多计算机科学家进行了诠释，先后提出了 100 多条有关软件工程的相关原则。著名软件工程专家 B. W. Boehm 集众家所长，并总结了 TRW 公司多年开发软件的经验，在 1983 年提出了软件工程的 7 项基本原则，作为保证软件产品质量和开发效率的最小集合。

从首次提出“软件工程”的概念开始，迄今已经过了半个世纪，在此期间，计算机硬件、软件技术领域都有了长足的发展，各种新产品、新技术、新方法、新工具不断问世。伴随着计算机科学与技术的进步，软件工程作为一门新兴学科也同样有了很大的发展。从传统的软件工程到面向对象的软件工程，从一般的软件工程到净室软件工程，从软件工程到软件再工程，从人工软件工程到计算机辅助软件工程，整个软件工程学正在日趋走向成熟，并在计算机应用领域中发挥着越来越大的作用。

1.3 软件工程的研究对象和基本原理

1.3.1 软件工程的研究对象

软件工程是相当复杂的，涉及的因素很多，不同软件项目使用的开发方法和技术也是不同的，而且有些项目的开发无现成的技术，带有不同程度的试探性。一般来说，软件工程包含 4 个关键元素：方法（methodologies）、语言（languages）、工具（tools）和过程（procedures）。

软件方法提供如何构造软件的技术，包括以下内容：与项目有关的计算和各种估算，系统和软件需求分析，数据结构设计，程序体系结构，算法过程，编码，测试和维护等。软件工程的方法通常引入各种专用的图形符号，以及一套软件质量的准则。概括地说，软件工程方法规定了以下内容：明确的工作步骤与技术；具体的文档格式；明确的评价标准。

软件语言用于支持软件的分析、设计和实现。随着编译程序和软件技术的完善，传统的编程语言表述能力更强、更加灵活，而且支持过程实现更加抽象的描述。与此同时，规格说明语言和设计语言也开始有更大的可执行子集。而且，现在还发展了原型开发语言，所谓原型开发语言就是除必须具有可执行的能力外，还必须具有规格说明和设计这两种语言的能力。

软件工具是人类在开发软件的活动中智力和体力的扩展和延伸，为方法和语言提供自动或半自动化的支持。软件工具最初是零散的，后来根据不同类型软件项目的要求建立了各种软件工具箱，支持软件开发的全过程。更进一步地，人们将用于开发软件的软、硬件工具和软件工程数据库（包括分析、设计、编码和测试等重要信息的数据结构）集成在一起，建立集成化的计算机辅助软件工程（CASE，computer-aided software engineering）系统。

软件过程贯穿于软件开发的各个环节。软件过程定义了方法使用的顺序、可交付产品