

北京市教育委员会共建项目专项资助
全国普通高等院校计算机专业精品规划教材

软件工程学 教程

陈明 编著



 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

北京市教育委员会共建项目专项资助
全国普通高等院校计算机专业精品规划教材

软件工程学教程

陈 明 编著

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书是软件工程方面的教材，主要内容包括软件工程概述、软件项目分析、软件需求工程、软件项目概要设计、软件项目详细设计、面向对象的分析与设计、用户界面的设计、软件项目编码、软件质量与质量保证、软件项目测试、软件项目交付与维护、软件项目管理、软件建模及 UML 建模语言等。

本书的特点是内容系统、语言精练、概念准确。本书可作为高等学校软件工程课程的教材，也可作为从事软件工程人员的参考书。

版权专有 侵权必究

图书在版编目 (CIP) 数据

软件工程学教程 / 陈明编著. —北京: 北京理工大学出版社, 2013.8
ISBN 978-7-5640-7926-0

I. ①软… II. ①陈… III. ①软件工程—高等学校—教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2013) 第 159492 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

82562903 (教材售后服务热线)

68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京高岭印刷有限公司

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 23.5

字 数 / 577 千字

版 次 / 2013 年 8 月第 1 版 2013 年 8 月第 1 次印刷

定 价 / 42.00 元

加工编辑/ 陈莉华

责任编辑/ 陈莉华

责任校对/ 周瑞红

责任印制/ 王美丽

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换

前 言

计算机软件是逻辑产品，而不是物理产品。软件与硬件具有完全不同的特征。计算机软件现已成为一种新的驱动力，是进行决策的引擎，是现代工程研究和解决问题的基础。软件已成为独立的一级学科，在各领域中发挥越来越重要的作用。

软件危机的出现导引了软件工程学。软件工程历经近 50 多年的飞速发展，现已成为一门重要的独立学科和专业，具有较完整的学科体系。软件工程的目的是以解决软件生产的质量与效率为宗旨，研究科学的工程方法以及与此相应的方便的软件工具系统，用来指导和帮助软件的开发与研究，在软件的开发与研究中起到重要的技术保障与促进作用。

软件工程是研究开发大型软件系统的学科，软件工程不仅覆盖了构建软件系统的相关技术问题，还包括指导开发团队、安排进度及预算等管理问题。软件工程不仅包括编写程序代码所涉及的技术，还包括所有对软件开发能够造成影响的问题。不存在任何单一开发技术或管理技术能够解决软件工程所面临的所有问题。因而，软件工程是包括一系列概念、理论、模式、语言、方法及工具的综合性学科。软件工程技术可分为产品实现技术及开发管理技术，产品实现技术主要涉及软件系统开发相关问题，为实现软件产品提供支持；开发管理技术通常不针对某个软件开发项目，而是为管理和改进软件，组织所有的业务活动提供技术支持。

软件开发工具是支持软件生存期中某一阶段的任务实现而使用的计算机程序。软件开发环境是一组相关的软件工具的集合，将它们集成在一起支持某种软件开发方法或某种软件开发模型。软件开发工具与环境是软件工程的重要组成部分，对于提高软件生产率，改进软件质量有越来越大的作用。

软件工程虽然彰显工程特色，但不是刻板的学科，需要丰富的想象力。软件工程是一门科学，它是有规律和步骤可循的。软件工程又是一个实践性极强的实用学科，在学习中，不仅要掌握其理论原则与方法，更重要的是能学会熟练地应用。计算机科学与技术专业和相近专业的毕业生，有相当部分的同学要从事计算机软件开发和应用工作，通过软件工程的理论学习与实践，可以培养学生以软件工程的方法开发软件的习惯和素质，并在软件开发的工作中得以贯彻，展现了学科的力量。

学习软件工程应注重计算思维能力、算法设计与分析能力、程序设计和实现能力、系统能力（系统的认知、设计、开发、应用）的培养，尤其是系统能力。系统能力包含两个层面的含义：一是对一定规模的系统的全局掌控能力，二是能够在构建系统时，系统地考虑问题的求解，充分体现了构造性思维方法。

本书是软件工程教程，共分为 13 章，包括软件工程概述、软件项目分析、软件需求工程、软件项目概要设计、软件项目详细设计、面向对象的分析与设计、用户界面的设计、软



件项目编码、软件质量与质量保证、软件项目测试、软件项目交付与维护、软件项目管理、软件建模及 UML 建模语言等。

在内容选择上，注重先进和系统，各章呈模块化，有助于快速掌握软件工程的原则和方法。

由于作者水平有限，书中不足之处在所难免，敬请读者批评指正。

陈明

第 1 章 软件工程概述	(1)
1.1 软件	(1)
1.1.1 软件的发展	(1)
1.1.2 软件的定义	(3)
1.1.3 软件的特点	(3)
1.1.4 软件的分类	(5)
1.2 软件工程的内容与方法	(7)
1.2.1 软件危机与软件工程的定义	(7)
1.2.2 软件工程的基本内容与目标	(10)
1.2.3 软件的基本开发方法	(10)
1.2.4 软件工程的基本原则	(11)
1.3 软件生存周期与软件开发模型	(11)
1.3.1 软件生存周期	(12)
1.3.2 软件开发模型	(13)
1.4 软件工具与环境	(21)
1.4.1 软件工具概述	(21)
1.4.2 软件开发工具的功能	(22)
1.4.3 软件开发工具的特性	(23)
1.4.4 软件开发工具的分类	(24)
1.4.5 软件开发环境	(25)
1.5 软件工程学的知识体系	(26)
小结	(27)
习题 1	(27)
第 2 章 软件项目可行性分析	(28)
2.1 软件项目可行性分析的任务	(28)
2.2 软件项目可行性分析的步骤	(29)
2.3 软件项目可行性分析的要素	(30)
2.3.1 经济可行性	(31)
2.3.2 技术可行性	(31)
2.3.3 社会环境可行性	(31)



2.4	系统流程图	(32)
2.5	成本-效益分析	(33)
2.5.1	成本估计	(33)
2.5.2	费用估计	(35)
2.5.3	度量效益的方法	(36)
	小结	(37)
	习题 2	(37)
第 3 章	软件需求工程	(38)
3.1	概述	(38)
3.1.1	软件需求分类	(39)
3.1.2	需求规格说明	(40)
3.1.3	需求工程概念	(41)
3.1.4	需求工程过程	(41)
3.2	需求获取方法	(42)
3.3	需求分析的任务与原则	(44)
3.3.1	需求分析的任务	(44)
3.3.2	需求分析的原则	(45)
3.4	需求建模方法	(45)
3.4.1	需求分析模型	(46)
3.4.2	结构化需求建模方法	(46)
3.4.3	数据流图	(49)
3.4.4	数据字典	(55)
3.5	需求图形工具	(59)
3.5.1	层次框图	(59)
3.5.2	Warnier 图	(60)
3.5.3	IPO 图	(60)
3.6	需求描述	(61)
3.6.1	需求规格说明文档的作用	(61)
3.6.2	需求规格说明文档的类型	(62)
3.7	需求验证	(62)
3.7.1	目的与任务	(63)
3.7.2	内容与方法	(63)
3.7.3	需求评审	(64)
3.8	需求管理	(65)
3.8.1	需求管理的目标	(65)
3.8.2	需求管理的原则	(65)
3.8.3	需求开发的管理	(66)
3.8.4	需求管理活动	(67)
	小结	(68)



习题 3	(69)
第 4 章 软件项目概要设计	(70)
4.1 软件体系结构	(70)
4.1.1 概述	(70)
4.1.2 系统构成	(72)
4.1.3 控制模型	(77)
4.1.4 模块化分解	(80)
4.1.5 领域相关的体系结构	(82)
4.2 概要设计任务与过程	(84)
4.2.1 概要设计任务	(84)
4.2.2 概要设计过程	(84)
4.3 软件设计的基本概念	(85)
4.3.1 模块化与模块独立性	(86)
4.3.2 抽象	(91)
4.3.3 结构设计原则	(93)
4.3.4 软件复用	(96)
4.3.5 设计模式	(98)
4.4 面向数据流的设计方法	(100)
4.4.1 基本概念	(100)
4.4.2 系统结构图的组成	(101)
4.4.3 变换分析	(103)
4.4.4 事务分析	(106)
4.4.5 设计优化	(108)
4.5 面向数据结构的设计方法	(108)
4.5.1 Jackson 系统开发方法	(108)
4.5.2 Warnier 方法	(117)
4.6 概要设计文档评审	(118)
4.7 设计特征与设计原则	(119)
小结	(119)
习题 4	(119)
第 5 章 软件项目详细设计	(121)
5.1 详细设计的任务与原则	(121)
5.1.1 详细设计的任务	(121)
5.1.2 详细设计的原则	(122)
5.2 详细设计工具	(122)
5.2.1 程序流程图	(123)
5.2.2 N-S 图	(124)
5.2.3 PAD 图	(125)
5.2.4 PDL	(127)



5.2.5	HIPO 图	(128)
5.2.6	详细设计工具的属性	(129)
5.3	详细设计规格说明与复审	(130)
5.3.1	详细设计说明书	(130)
5.3.2	设计复审	(131)
5.4	模块文件夹	(131)
	小结	(132)
	习题 5	(132)
第 6 章	面向对象的分析与设计	(133)
6.1	面向对象方法	(133)
6.1.1	面向对象方法概述	(134)
6.1.2	面向对象软件工程 (OOSE)	(135)
6.1.3	面向对象的基本概念和特征	(138)
6.2	面向对象的分析	(145)
6.2.1	面向对象分析过程与原则	(145)
6.2.2	确定对象与类	(149)
6.2.3	确定属性	(152)
6.2.4	定义服务	(152)
6.2.5	对象间通信	(156)
6.3	面向对象设计	(162)
6.3.1	面向对象设计的概念	(162)
6.3.2	面向对象设计的方法	(163)
	小结	(165)
	习题 6	(166)
第 7 章	用户界面的设计	(167)
7.1	用户界面的主要特征	(167)
7.1.1	用户界面的质量特征	(167)
7.1.2	用户界面的图形特征	(168)
7.1.3	用户界面的优点	(168)
7.2	设计原则	(168)
7.3	用户交互	(170)
7.3.1	交互方式	(170)
7.3.2	交互方式的比较	(171)
7.4	信息表示	(172)
7.4.1	用户界面的信息表示方法	(172)
7.4.2	用户界面信息表示的一些问题	(172)
7.5	帮助系统	(174)
7.6	界面设计	(175)
7.6.1	数据输入界面设计	(176)



7.6.2	数据显示界面设计	(176)
7.6.3	控制界面设计	(177)
7.7	界面设计评价	(177)
7.7.1	界面设计评价指标	(178)
7.7.2	界面评价方法	(178)
	小结	(179)
	习题 7	(179)
第 8 章	软件项目编码	(180)
8.1	程序设计语言	(180)
8.1.1	程序设计语言的分类	(180)
8.1.2	程序设计语言的特点	(182)
8.1.3	程序设计语言的选择	(184)
8.2	编码风格	(186)
8.2.1	源程序文档化	(186)
8.2.2	数据说明	(188)
8.2.3	语句结构	(188)
8.2.4	输入/输出	(191)
8.3	程序效率	(191)
8.3.1	程序效率准则	(191)
8.3.2	算法对效率的影响	(192)
8.3.3	影响存储器效率的因素	(192)
8.3.4	影响输入/输出的因素	(192)
8.4	编程安全	(193)
8.4.1	冗余程序设计	(193)
8.4.2	防错程序设计	(193)
8.5	面向对象程序设计步骤	(194)
8.6	编码优化	(195)
8.6.1	优化的内容与原则	(195)
8.6.2	程序结构优化	(196)
8.6.3	程序代码优化	(196)
	小结	(196)
	习题 8	(197)
第 9 章	软件质量与质量保证	(198)
9.1	软件质量的定义	(198)
9.2	影响软件质量的因素	(199)
9.3	软件质量保证	(200)
9.3.1	软件质量保证概念	(200)
9.3.2	软件质量保证策略	(200)
9.3.3	SQA 小组的任务	(201)



9.4	软件质量保证活动	(202)
9.5	软件评审	(202)
9.5.1	设计质量的评审内容	(203)
9.5.2	程序质量的评审内容	(207)
9.6	软件质量保证的标准	(210)
9.7	软件质量评价	(211)
9.7.1	软件质量评价体系	(211)
9.7.2	软件质量评价标准	(213)
9.8	软件质量框架	(214)
9.8.1	高质量软件的特性	(214)
9.8.2	软件质量框架的组成	(214)
9.9	软件开发质量的定量描述	(216)
9.9.1	基本的定量估算	(216)
9.9.2	软件需求的估算	(217)
9.9.3	估算验收测试阶段预期发现的缺陷数	(217)
9.9.4	维护活动设计的度量	(218)
9.9.5	软件可用性的计算	(218)
9.9.6	基于植入故障方法的原有故障总数 E_N 估算	(218)
	小结	(219)
	习题 9	(219)
第 10 章	软件项目测试	(220)
10.1	概述	(220)
10.1.1	软件测试技术的发展	(220)
10.1.2	软件错误与缺陷	(221)
10.1.3	软件测试的定义	(222)
10.1.4	软件测试的对象	(222)
10.1.5	软件测试的目的	(222)
10.1.6	软件测试的原则	(223)
10.1.7	软件测试的复杂性	(223)
10.1.8	软件开发各阶段的测试	(224)
10.2	软件测试方法	(224)
10.2.1	静态分析	(224)
10.2.2	动态测试	(226)
10.2.3	人工测试与机器测试	(227)
10.2.4	黑盒测试	(228)
10.2.5	白盒测试	(237)
10.2.6	白盒测试和黑盒测试的比较	(243)
10.3	单元测试与集成测试	(244)
10.3.1	单元测试	(245)



10.3.2	集成测试	(248)
10.3.3	确认测试	(253)
10.3.4	系统测试	(256)
10.3.5	终止测试	(260)
10.4	面向对象测试	(261)
10.4.1	面向对象测试基础	(261)
10.4.2	面向对象测试模型	(262)
10.4.3	类测试	(267)
10.4.4	面向对象的集成测试	(271)
10.4.5	面向对象的系统测试	(273)
10.4.6	面向对象测试与传统测试的比较	(274)
10.5	测试的设计与实现	(274)
10.5.1	测试计划	(275)
10.5.2	测试设计	(281)
10.5.3	测试执行	(287)
10.5.4	测试总结	(289)
	小结	(290)
	习题 10	(290)
第 11 章	软件项目交付与维护	(293)
11.1	软件发布与部署	(293)
11.1.1	软件产品发布	(293)
11.1.2	软件产品实施	(294)
11.2	软件演化的特征	(295)
11.2.1	软件演化的动态特征	(295)
11.2.2	软件体系结构的进化	(296)
11.3	软件维护	(296)
11.3.1	软件维护分类与特点	(296)
11.3.2	软件维护步骤	(298)
11.3.3	软件的可维护性	(301)
11.3.4	软件维护的副作用	(302)
11.4	逆向工程和再生工程	(303)
	小结	(304)
	习题 11	(304)
第 12 章	软件项目管理	(305)
12.1	软件项目特点及软件管理功能	(305)
12.1.1	软件项目的特点	(305)
12.1.2	软件管理的功能	(306)
12.1.3	软件项目的工作范围	(306)
12.2	软件与硬件资源	(307)



12.2.1	人力资源	(307)
12.2.2	硬件	(307)
12.2.3	软件	(307)
12.3	人员的计划和组织	(308)
12.4	成本估计及控制	(309)
12.4.1	软件开发成本估算方法	(309)
12.4.2	专家估算法	(310)
12.4.3	成本估算模型	(310)
12.5	进度计划	(313)
12.5.1	软件工作的特殊性	(313)
12.5.2	各阶段工作量的分配	(313)
12.5.3	制定开发进度	(314)
12.6	软件配置管理	(314)
12.6.1	基线	(314)
12.6.2	软件配置项	(315)
12.6.3	软件配置管理过程	(317)
12.7	软件管理方案	(319)
12.8	能力成熟度模型	(320)
12.8.1	软件过程评估的必要性	(320)
12.8.2	CMM 的主要用途	(322)
12.8.3	CMM 的体系结构	(322)
12.8.4	CMM 的等级	(324)
12.8.5	CMM 的内部结构	(327)
12.8.6	CMM 的应用	(330)
	小结	(331)
	习题 12	(331)
第 13 章	软件建模及 UML 建模语言	(332)
13.1	模型概述	(332)
13.1.1	模型的定义	(332)
13.1.2	模型的分类	(332)
13.2	常用的软件建模模型	(333)
13.2.1	业务模型	(333)
13.2.2	功能模型	(333)
13.2.3	数据模型	(333)
13.3	需求建模	(334)
13.3.1	关联模型	(334)
13.3.2	行为模型	(334)
13.3.3	面向对象模型	(334)
13.3.4	需求数据模型	(334)



13.4 统一建模语言	(334)
13.4.1 UML 的概述	(335)
13.4.2 UML 的表示法	(340)
13.4.3 UML 软件开发过程概述	(351)
13.4.4 Rational Rose 概述	(353)
13.5 形式化方法	(355)
小结	(356)
习题 13	(356)
参考文献	(357)

第1章

软件工程概述

学习目标

- 了解软件的定义、软件工程的定义和软件工程学的知识体系。
- 理解软件的特点、软件的分类、软件工程的基本开发方法等。
- 掌握软件工程的基本内容与目标、软件工程的基本原则、软件生存周期和软件开发模型等。
- 了解软件开发工具的功能、特性和分类。

1.1 软件

软件是一种产品，也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。不管它是驻留在设备中还是在主机中，软件是一个信息转换器，能够产生、管理、获取、修改、显示或转换信息。这些信息可以很简单，也可以很复杂，如多媒体信息。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。

信息是 21 世纪最重要的产品，软件充分地体现了这一点。通过软件处理数据，凸显了数据的重要性；软件管理商业信息，更增强了商业竞争力。软件不仅提供了通往全球信息网络的途径，而且也提供了获取信息的多种手段。

1.1.1 软件的发展

计算机硬件的发展规律基本遵循了摩尔定律，即每 18 个月芯片的性能提高一倍。但软件的发展也非常迅速，在软件体系结构上，经历了从主机结构到文件服务器结构、从客户/服务器到基于 Internet 的浏览器/服务器结构的变化；在程序语言上，经历了从机器语言到汇编语言、从高级程序语言到第四代语言的变化；在开发工具上，从分离开发工具到集成的可视化开发工具、从简单的命令行调试器到多功能调试器等变化。



软件早期依附于硬件，现在已经成为单独产品，形成了专门的软件工程学科。在 40 多年的发展过程中，软件工程发生了从面向过程、面向对象、面向构件到面向领域等变化，积累了工程化方法和大量的可重用性资源。在 21 世纪，软件生产更多地要满足网络环境下大众用户的多元化和个性化需求，面向服务成为软件新的范型。随着 Web 服务和语义 Web 等新理念的蓬勃发展，软件将以更灵活的、开放的服务方式通过网络来组织并提供给用户。软件的发展经历了下述过程。

1. 第一阶段（程序设计阶段）

计算机发展的早期阶段（20 世纪 50 年代初期至 20 世纪 60 年代中期）为程序设计阶段。在这个阶段硬件已经通用化，而软件的生产却是个体化的。这时，由于程序规模小，几乎没有什么系统化的方法可遵循。对软件的开发没有任何管理方法，一旦计划推迟了或者成本提高了，程序员才开始弥补。在通用的硬件已经非常普遍的时候，软件产品还处在初级阶段，对每一类应用均需自行再设计，应用范围很有限。设计往往仅是人们头脑中的一种模糊想法，而文档根本不存在。

2. 第二阶段（程序系统阶段）

计算机系统发展的第二阶段（20 世纪 60 年代中期至 20 世纪 70 年代末期）为程序系统阶段。多道程序设计、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的新世界，硬件和软件配合达到了一个新层次，出现了实时系统和第一代数据库管理系统。这个阶段的另一个特点就是软件产品的使用和“软件作坊”的出现。被开发的软件可以在较广泛的范围内应用。主机和微机上的程序能够有数百甚至上千的用户。

在软件的使用过程中，若发现错误时，当用户需要的硬件环境发生变化时都要修改软件，这些活动统称为软件维护。在软件维护上的花费以惊人的速度增长，更为严重的是，许多程序的个体化特性使得它们根本不能维护。这样，“软件危机”就出现了。

3. 第三阶段（软件工程阶段）

计算机系统发展的第三阶段始于 20 世纪 70 年代中期并跨越了近 10 年，被称为软件工程阶段。在这一阶段，以软件的产品化、系列化、工程化、标准化为特征的软件产业发展起来，打破了软件生产的个体化特征，有了可以遵循的软件工程化的设计原则、方法和标准。在分布式系统中，各台计算机同时执行某些功能，并与其他计算机通信，极大地提高了计算机系统的复杂性。广域网、局域网、高带宽数字通信以及对“即时”数据访问需求的增加都对软件开发者提出了更高的要求。

4. 第四阶段

计算机系统发展的第四阶段已经不再着重于单台计算机和计算机程序，而是针对计算机和软件的综合影响。由复杂操作系统控制的强大的桌面机、广域网络和局域网络，配以先进的软件应用已成为标准。计算机体系结构迅速地从集中的主机环境转变为分布的客户机/服务器环境。世界范围的信息网提供了一个基本结构，信息高速公路和网际空间连通已成为令人关注的热点问题。事实上，Internet 可以看作是能够被单个用户访问的软件，计算机发展正朝着社会信息化和软件产业化方向发展，从技术的软件工程阶段过渡到社会信息化的计算机系统阶段。随着第四阶段的进展，一些新技术开始涌现。面向对象技术将在许多领域中迅速取代传统软件开发方法。表 1-1 给出了计算机发展 4 个阶段的典型技术。

表 1-1 4 个阶段的典型技术

阶 段	第一阶段	第二阶段	第三阶段	第四阶段
典型技术	● 面向批处理 ● 有限的分布 ● 自定义软件	● 多用户 ● 实时 ● 数据库 ● 软件产品	● 分布式系统 ● 嵌入“智能” ● 低成本硬件 ● 消费者的影响	● 强大的桌面系统 ● 面向对象技术 ● 专家系统 ● 并行计算 ● 云计算 ● 大数据计算

1.1.2 软件的定义

计算机系统是通过运行程序来实现各种不同应用的。把各种不同功能的程序，包括用户为自己的特定目的编写的程序、检查和诊断机器系统的程序、支持用户应用程序运行的系统程序、管理和控制机器系统资源的程序等，通常称为软件。它是计算机系统中与硬件相互依存的另一部分，与硬件合为一体完成系统功能。软件定义如下：

- (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) 描述程序研制过程和方法所用的文档。

随着计算机应用的日益普及，软件变得越来越复杂，规模也越来越大，这就使得人与人、人与机器间相互沟通，保证软件开发与维护工作的顺利进行显得特别重要，因此，文档（即各种报告、说明、手册的总称）是不可缺少的。特别是在软件日益成为产品的今天，文档的作用就更加重要。

1.1.3 软件的特点

软件在整个计算机系统中是一个逻辑部件，而硬件是一个物理部件。因此，软件相对硬件而言有许多特点。为了能全面、正确地理解计算机软件及软件工程的重要性，必须了解软件的特点。软件的特点可归纳如下：

- (1) 软件是一种逻辑实体，而不是具体的物理实体，因而它具有抽象性。这个特点使它与计算机硬件或其他工程对象有着明显的差别。人们可以把它记录在介质上，但无法看到软件的形态，而必须通过测试、分析、思考、判断去了解它的功能、性能及其他特性。
- (2) 软件是通过人们的智力活动，把知识与技术转化成信息的一种产品，是在研制、开发中被创造出来的。一旦某一软件项目研制成功，以后就可以大量地复制同一内容的副本，这就是软件的生产过程。也就是说，软件的研制成本远远大于其生产成本。软件故障往往是在开发时产生而在测试时没有发现的问题，所以要保证软件的质量，必须着重于软件开发过程，加强管理。
- (3) 在软件的运行和使用期间，没有硬件那样的机械磨损、老化问题。软件维护比硬件维护要复杂得多，与硬件的维修有着本质的差别，参阅图 1-1、图 1-2 和图 1-3。图 1-1 所示为硬件的故障率随时间变化的曲线，图 1-2 所示为在理想情况下软件故障率随时间变化的曲