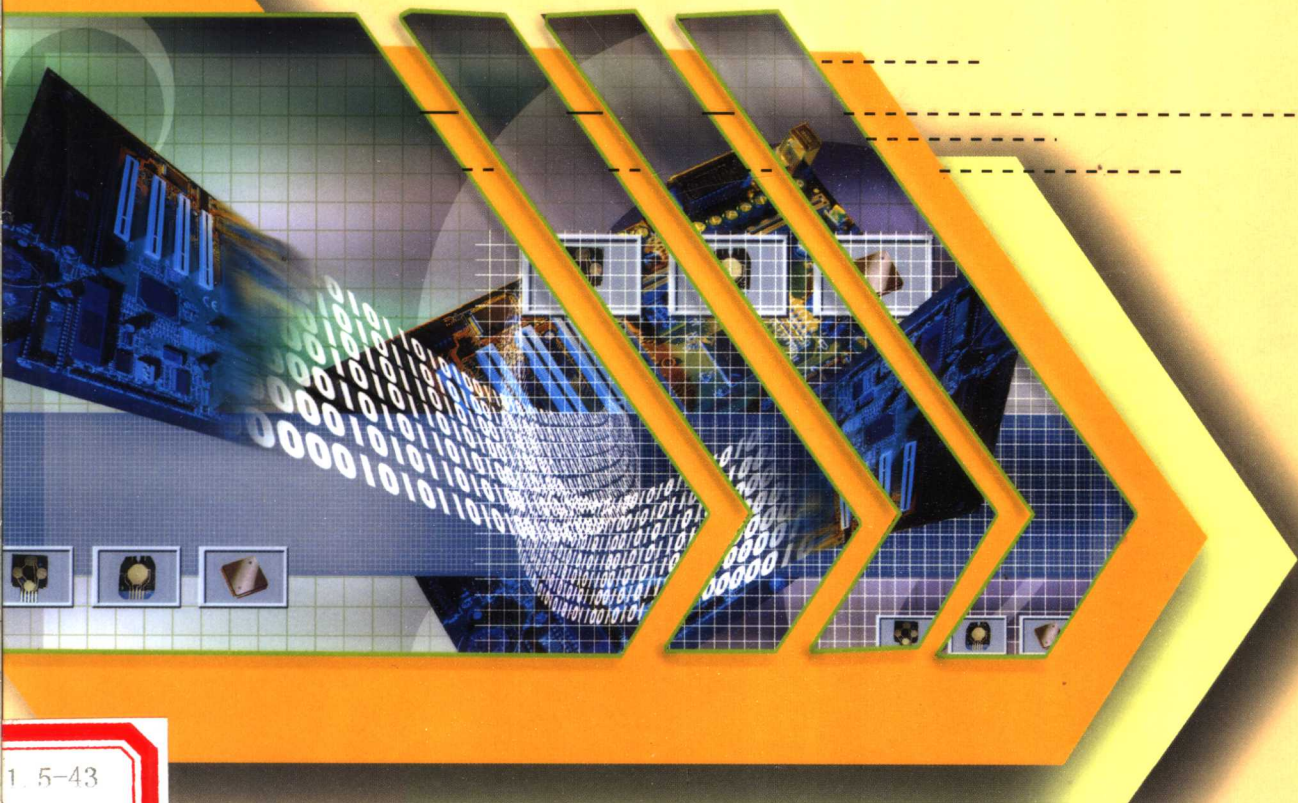


高等院校计算机应用技术规划教材

主编 谭浩强

软件工程 实用教程

陈明 编著



1.5-43

清华大学出版社



高等院校计算机应用技术规划教材

主编 谭浩强

软件工程 实用教程

陈 明 编著

清华大学出版社

北京

内 容 简 介

本书主要介绍了软件工程学及其应用,内容包括可行性研究、需求分析、概要设计、详细设计、编码、软件质量与质量保证、项目计划与管理、软件开发工具与环境方面的内容。为了保持教材内容的先进性,也包含了面向对象软件工程学。本书可作为高等院校的教材,也可作为从事软件开发与应用的工程人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

软件工程实用教程/陈明编著. —北京:清华大学出版社,2005.4

(高等院校计算机应用技术规划教材)

ISBN 7-302-10287-2

I. 软… II. 陈… 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(2004)第 143572 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客 户 服 务:010-62776969

责任编辑:谢 琛 xiech@tup.tsinghua.edu.cn

封面设计:常雪影

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市李旗庄少明装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:15.75 字数:361千字

版 次:2005年4月第1版 2005年4月第1次印刷

书 号:ISBN 7-302-10287-2/TP·7006

印 数:1~5000

定 价:21.00元

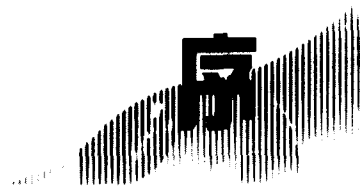
编辑委员会

《高等院校计算机应用技术规划教材》

主 任 谭浩强
副 主 任 陈 明 丁桂芝

委 员 (按姓氏笔画排序)

王智广	刘 星	刘荫铭	安志远
安淑芝	孙 慧	李文英	李叶紫
李 琳	李雁翎	宋 红	秦建中
陈 强	邵丽萍	尚晓航	张 玲
侯冬梅	郝 玲	赵丰年	莫治雄
袁 玫	訾秀玲	薛淑斌	谢树煜
谢 琛			



《高等院校计算机应用技术规划教材》

进入 21 世纪,计算机成为人类常用的现代工具,每一个有文化的人都应当了解计算机,学会使用计算机来处理面临的事务。

学习计算机知识有两种不同的方法:一种是侧重知识的学法,从原理入手,注重理论和概念;另一种是侧重于应用,从实际入手,注重掌握其应用的方法和技能。不同的人应根据其具体情况选择不同的学习方法。对大多数人来说,计算机是作为一种工具来使用的,应当以应用为目的,以应用为出发点。对于应用型人才来说,显然应当采用后一种学习方法。

传统的理论课程采用以下的三部曲:提出概念——解释概念——举例说明,这适合前面第一种侧重知识的学习方法。对于侧重应用的学习者,我们提倡新的三部曲:提出问题——解决问题——归纳分析。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。这种教学方法更适合于应用型人才。

应当指出,检查学习好坏的标准,不是“知道不知道”,而是“会用不会用”,学习的目的主要在于应用。因此希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有一些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而可以指定学生通过上机掌握这些内容。这样做可以培养学生的自学能力,启发学生的求知欲望。

1999 年我主编了一套《高职高专计算机系列教材》,由清华大学出版社出版,包括了 30 余种教材学习用书,受到全国各高校广大师生的欢迎,不仅许多高职高专学校选用它为教材,而且不少培养应用性人才的本科院校也认为很适合他们的教学特点、对提高教学质量起到了积极的作用。为了扩大这套教材的使用面,我们将这套教材改名为《高等院校计算机应用技术规划教材》,并对教材的内容进行了重新安排,包括 3 个子系列:

(1) 适合于培养应用性人才的院校和基础较好、要求较高的高职高专学校的教材和参考书。封面颜色为黄色。



(2) 组织编写出版一批“实用教程”,内容主要是引导学生直接学习和掌握计算机应用技能,对象是理论要求相对低一些或学时较少的学校(尤其是2年制高职高专)的需要。封面颜色为橘色。

(3) 组织若干种“实训教材”,其特点是侧重实践环节,引导学生通过自己的实践(而不是通过理论讲授)去获取知识,掌握应用。我们认为这应该是教学改革的一个重要方面,各类学校都可以选用这些实训教材。封面颜色为蓝色。

本套教材是针对培养应用性人才的院校和高职高专院校的教学特点组织编写的,包括了计算机专业和非计算机专业的教材和参考书。不同专业可以从中选择所需要的部分。本套教材包含的内容比较广,除了可作为正式教材外,还可作为某些专业的选修课或指定自学的教材。

本套教材是由一些普通高校和高职院校的老师编写的,他们对应用型的教学特点有较多的了解,也有较多的实践经验,保证了教材的质量。

由于我国的计算机应用技术教育正在蓬勃发展,许多问题有待深入讨论,新的经验也会层出不穷,本套教材的内容将会根据新的形势不断丰富和调整。

本套教材和参考书中肯定会有不足之处,请专家和读者不吝指正。

全国高等院校计算机基础教育研究会会长

谭浩强

2005年1月

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。软件产业是创新的、充满活力的和成功的产业。

计算机软件现已成为一种新的驱动力,是进行决策的引擎,是现代工程研究和问题解决的基础。在各种类型的应用系统中无所不在、广泛应用。

软件危机是指软件开发和维护过程中遇到的一系列严重问题。例如,如何开发软件,如何满足对软件的日益增长的需求,如何维护数量不断膨胀的已有软件等一系列问题。

为了克服与摆脱软件危机,诞生了**软件工程学**。从 1969 年提出软件工程概念以来,历经 30 多年飞速地发展,软件工程学逐渐成熟,现已成为计算机科学与技术领域中一门重要的学科。软件工程学的目标是以解决软件生产的质量和效率为宗旨,研究一套科学的工程方法,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,对软件的开发与研究起到重要的技术保障与促进作用。

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,它们组织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。

本书较系统地介绍了软件工程学的内容,主要包括:软件工程概述、可行性分析、需求分析、概要设计、详细设计、面向对象的分析与设计方法、编码、软件质量与质量保证、项目计划与管理、软件开发工具与环境等。

软件工程是一个实践性极强的实用学科,在学习中,不仅要能掌握其理论原则与方法,更重要的是能学会熟练的应用。计算机应用专业和相近专业的学生毕业后,有相当部分的同学要从事计算机软件开发和应用工作,通过软件工程理论的学习与实践,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。

本书的内容较为系统,注重基本概念的解释和方法的说明,各章都附有内

容小结、习题,便于学习总结和练习。

在结构上呈积木状和模块化,各章内容相对独立,但在逻辑上是一整体,便于选择性学习。

在取材上注重实用,避开了过多的理论说明和推演,有助于快速掌握原则和方法。

由于作者水平有限,书中不足之处在所难免,敬请批评指正。

陈 明

2005年3月

目录

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 软件定义	3
1.1.3 软件的特点	3
1.1.4 软件分类	4
1.2 软件工程概念	6
1.2.1 软件危机与软件工程的定义	6
1.2.2 软件工程的原则	8
1.3 软件生存周期与软件开发模型	9
1.3.1 软件生存周期	9
1.3.2 软件开发模型	10
1.4 小结	16
习题	16
第 2 章 可行性研究	18
2.1 可行性研究的任务	18
2.2 可行性研究的步骤	19
2.3 系统流程图	21
2.4 成本-效益分析	22
2.4.1 成本估计	23
2.4.2 费用估计	24
2.4.3 几种度量效益的方法	25
2.5 小结	26
习题	26

第 3 章 需求分析	27
3.1 需求分析的任务与步骤	27
3.1.1 需求分析的任务	27
3.1.2 需求分析的步骤	29
3.1.3 需求分析的原则	31
3.2 需求分析的方法	32
3.2.1 面向数据流的需求分析方法	32
3.2.2 数据流图	33
3.2.3 数据字典	39
3.3 图形	44
3.3.1 层次方框图	44
3.3.2 Warnier 图	44
3.3.3 IPO 图	45
3.4 需求规格说明与评审	46
3.4.1 需求规格说明的主要内容	46
3.4.2 需求分析的评审	47
3.4.3 需求分析评审的方法	48
3.5 小结	49
习题	49
第 4 章 概要设计	50
4.1 概要设计的任务与步骤	50
4.1.1 概要设计任务	50
4.1.2 概要设计步骤	50
4.2 软件设计的原则	52
4.2.1 模块化	52
4.2.2 抽象与逐步求精	53
4.2.3 信息隐蔽和局部化	53
4.2.4 模块独立性	53
4.2.5 结构设计原则	57
4.3 面向数据流的设计方法	60
4.3.1 基本概念	61
4.3.2 系统结构图(SC 图)的组成	62
4.3.3 变换分析	64
4.3.4 事务分析	68

4.4	面向数据结构的设计方法	69
4.5	概要设计文档评审	78
4.6	小结	80
	习题	80
第 5 章	详细设计	81
5.1	详细设计的任务与原则	81
5.1.1	详细设计的任务	81
5.1.2	详细设计的原则	82
5.2	详细设计的方法	82
5.2.1	程序流程图	82
5.2.2	N-S 图	84
5.2.3	PAD 图	85
5.2.4	PDL	87
5.3	详细设计规格说明与复审	89
5.3.1	详细设计说明书	89
5.3.2	设计复审	90
5.4	小结	91
	习题	91
第 6 章	面向对象的分析和设计方法	92
6.1	面向对象方法的基本概念和特征	92
6.1.1	面向对象方法概述	92
6.1.2	面向对象的软件工程	94
6.1.3	面向对象的基本概念和特征	98
6.2	面向对象的分析	105
6.2.1	面向对象分析基本过程及原则	105
6.2.2	对象与类的确定	109
6.2.3	确定属性	112
6.2.4	定义服务	113
6.2.5	对象间通信	116
6.3	面向对象设计	122
6.3.1	面向对象设计的基本概念	122
6.3.2	面向对象设计的方法	124
6.4	UML	126
6.4.1	UML 的产生与发展	126

6.4.2	UML 的表示法	132
6.4.3	UML 软件开发过程概述	145
6.5	小结	148
	习题	148
第 7 章	编码	149
7.1	程序设计语言	149
7.1.1	程序设计语言分类	149
7.1.2	程序设计语言的特点	151
7.1.3	程序设计语言的选择	153
7.2	编码风格	154
7.2.1	源程序文档化	155
7.2.2	数据说明	157
7.2.3	语句结构	157
7.2.4	输入输出	159
7.3	程序效率及编程安全	160
7.3.1	有关程序效率的几条准则	160
7.3.2	算法对效率的影响	160
7.3.3	影响存储器效率的因素	160
7.3.4	影响输入输出的因素	161
7.3.5	编程安全	161
7.4	小结	162
	习题	162
第 8 章	软件质量与质量保证	163
8.1	软件质量的定义	163
8.2	影响软件质量的因素	163
8.3	软件质量保证策略	165
8.4	软件质量保证活动	166
8.5	软件评审	166
8.5.1	设计质量的评审内容	167
8.5.2	程序质量的评审内容	168
8.6	软件质量保证的标准	170
8.7	结构化的软件测试	171
8.7.1	软件测试的概念和原则	171
8.7.2	软件测试技术	173

8.7.3	测试的步骤	180
8.7.4	软件纠错技术	182
8.8	面向对象的软件测试	184
8.8.1	面向对象分析和面向对象设计的模型测试	184
8.8.2	面向对象的测试策略	186
8.8.3	面向对象软件测试集设计	187
8.9	软件测试计划与测试分析报告	189
8.10	软件维护	190
8.10.1	软件维护的定义、分类与特点	190
8.10.2	软件维护步骤及组织	193
8.10.3	软件的可维护性	197
8.10.4	软件维护的副作用	198
8.10.5	逆向工程和再生工程	199
8.11	小结	199
	习题	199

第9章 项目计划与管理 201

9.1	软件项目特点、管理的特殊性 & 软件管理功能	201
9.1.1	软件项目的特点	201
9.1.2	软件管理的功能	202
9.1.3	确定软件项目的工作范围	202
9.2	确定软硬件资源	202
9.2.1	人力资源	203
9.2.2	硬件	203
9.2.3	软件	203
9.3	人员的计划和组织	204
9.4	成本估计及控制	205
9.4.1	软件开发成本估计方法	205
9.4.2	专家判定技术	206
9.4.3	成本估算模型	207
9.5	进度计划	209
9.5.1	软件工作的特殊性	209
9.5.2	各阶段工作量的分配	210
9.5.3	制定开发进度	211
9.6	软件配置管理	211
9.6.1	基线	212
9.6.2	软件配置项	213

9.6.3 软件配置管理过程	214
9.7 软件管理方案	217
9.8 小结	219
习题	219
第 10 章 软件开发工具与环境概述	220
10.1 软件开发工具概述	220
10.2 软件开发工具的功能	221
10.3 软件开发工具的特性	222
10.4 软件开发工具的分类	223
10.5 软件开发环境	225
10.6 常用开发环境	226
10.6.1 Windows 操作系统	226
10.6.2 UNIX 程序开发环境	227
10.6.3 Linux 开发环境	232
10.7 软件开发工具与环境的应用	234
10.8 CASE 技术	235
10.9 小结	237
习题	237
参考文献	238

第1章

软件工程概述

1.1 软件

软件是一种逻辑产品,同时又是开发和运行产品的载体。作为一种逻辑产品,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,如只有一个 bit;也可以很复杂,如多媒体信息。作为开发运行产品的载体,软件是计算机工作的基础、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,软件处理数据,使得这些数据更为有用;软件管理商业信息增强了商业竞争力;它不仅提供了通往全球信息网络的途径,而且也提供了以各种形式获取信息的手段。

1.1.1 软件的发展

计算机硬件的发展遵循摩尔定律,每 18 个月芯片的性能与速度均提高一倍。软件的发展速度也很快。体系结构经历了四个阶段的变化:主机结构、文件结构、客户/服务器结构、浏览器/服务器结构。编程语言经历了机器语言、汇编语言、高级语言和第四代语言的变化。开发工具经历了分离工具到集成可视化开发工具的变化。归纳起来,软件发展分为下述几阶段。

1. 程序设计阶段

在计算机发展早期阶段(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标准方法可遵循。对软件的开发没有任何管理方法,一旦任务超时或者成本提高,程序员才开始修补。在通用的硬件已经非常普遍的时候,软件却相反,对每一类应用均需自行再设计,应用范围很有限。软件产品处在初级阶段,大多数软件都是由使用者自己开发,例如书写软件,使其运行,有问题则自行寻求解决的办法等。因为是个人的软件环境,设计往往仅是人们头脑中的一种模糊想法,并且根本就不存在文档。

2. 程序系统阶段

计算机系统发展的第二阶段(20世纪60年代中期到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和硬件与软件配合的新层次,出现了实时系统和第一代数据库管理系统。这个阶段另一个特点就是软件产品的使用和软件作坊的出现。开发出软件可以在较广泛的范围中应用,主机和微机上的程序能够拥有数百甚至上千的用户。

在软件的使用中,当发现错误时需要纠正程序源代码;当用户需求发生变化时需要修改程序源代码;当硬件环境变化时需要适应,这些活动统称为软件维护。但在软件维护上要消耗大量资源,更为严重的是,许多程序的个性化特性使其根本不能维护。于是“软件危机”出现了。

3. 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十年,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,有了软件工程化的设计原则、方法、标准可以遵循。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系统和程序,而是面向计算机和软件的综合影响。由复杂的操作系统控制的强大的桌面机、广域网络和局域网,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/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。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。随着第四阶段的进展,一些新技术开始出现。面向对象技术将在许多领域中迅速取代传统结构化软件开发方法。

表 1-1 给出了四个阶段典型技术的比较。

表 1-1 四个阶段典型技术

阶段	第一阶段	第二阶段	第三阶段	第四阶段
典型技术	<ul style="list-style-type: none">• 面向批处理• 有限的分布• 自定义软件	<ul style="list-style-type: none">• 多用户• 实时• 数据库• 软件产品	<ul style="list-style-type: none">• 分布式系统• 嵌入“智能”• 低成本硬件• 消费者的影响	<ul style="list-style-type: none">• 强大的桌面系统• 面向对象技术• 专家系统• 人工神经网络• 并行计算• 网络计算

1.1.2 软件定义

计算机系统是通过运行程序来实现各种不同的应用。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、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

在计算机系统中,软件是一个逻辑部件,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。虽然可以把它记录在介质上,但却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。即其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被发现的问题。所以要保证软件的质量,必须着重于软件开发过程。

(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。但软件维护比硬件维护要复杂得多,与硬件的维护有着本质的差别,参阅图 1-1、图 1-2 和图 1-3。图 1-1 所示的是硬件的故障率随时间变化的曲线,图 1-2 所示的是在理想情况下软件故障率随时间变化的曲线,图 1-3 所示的是软件的实际故障率曲线。

(4) 软件的开发和运行经常受到计算机系统的限制,对计算机系统有着不同程度的依赖关系。在软件的开发和运行中必须以硬件提供的条件为基础。为了消除这种依赖关系,在软件开发中提出了软件移植的问题,并且把软件的可移植性作为衡量软件质量的因素之一。

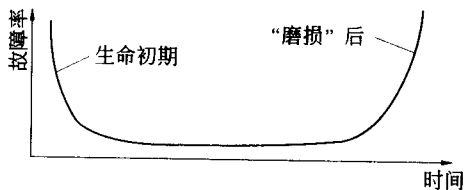


图 1-1 硬件的故障率随时间变化的曲线

(5) 软件的开发尚未完全摆脱手工的开发方式。由于传统的手工开发方式仍然占据统治地位,软件开发的效率受到很大的限制。因此,应促进软件技术发展,提出和采用新