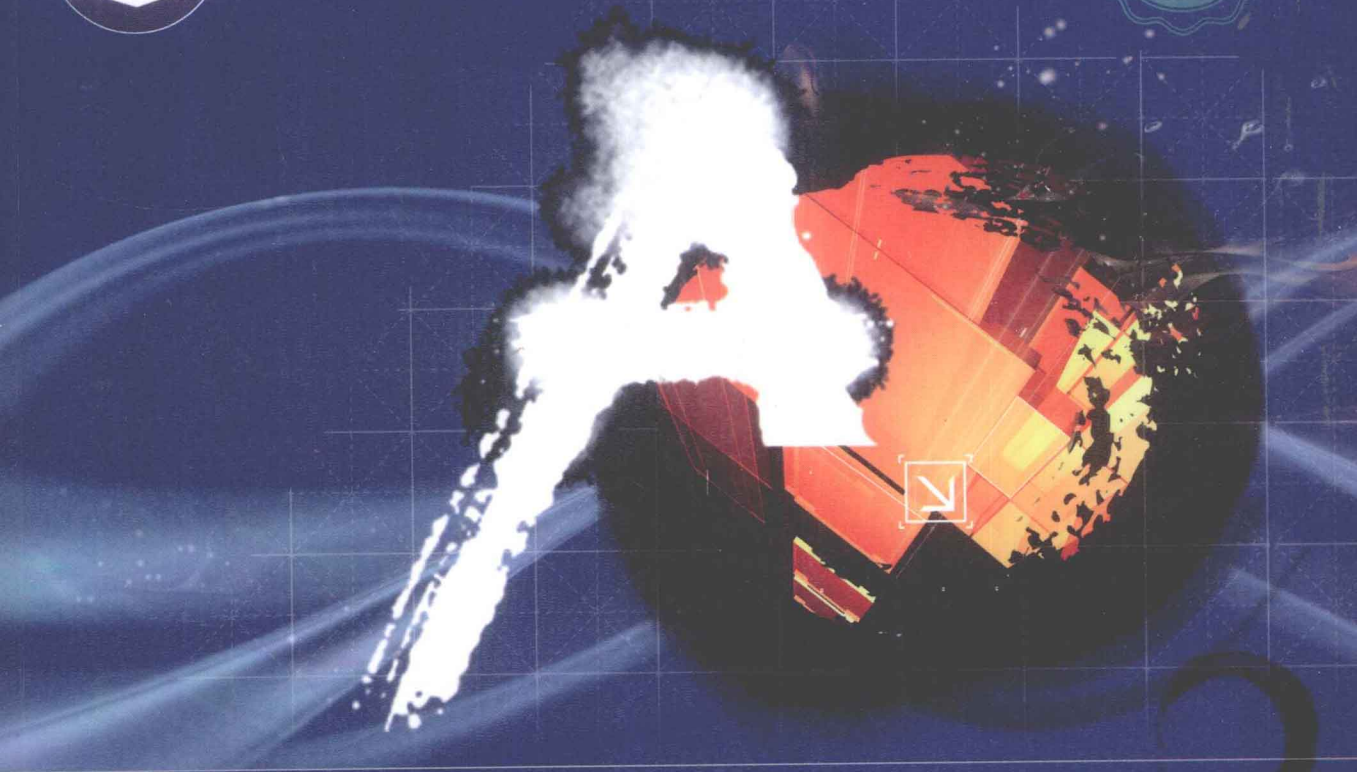




普通高等教育“十二五”规划教材



高等学校规划教材

软件工程——方法与实践

(第2版)

◎ 主编 许家珩

电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

高等学校规划教材

软件工程——方法与实践

(第2版)

许家珩 主编

许家珩 白忠建 吴磊 编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是在吸取了国内外有关教材的精华,并结合作者多年从事软件工程教学及软件开发的实践经验、体会的基础上编写的。内容注重科学性、先进性,强调实践性。在介绍软件工程基础知识的基础上,重点介绍面向对象的分析、设计、测试及UML统一建模语言,以及CMM软件成熟度模型、ERP企业资源规划等先进管理技术。提供了丰富的软件开发实例和素材,以及“软件工程课程网站”。

本书可作为高等院校计算机及信息类专业本科生及研究生的“软件工程”课程的教材,也可作为广大工程技术人员和科研人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

软件工程:方法与实践 / 许家珩主编. —2版. —北京:电子工业出版社,2011.12

高等学校规划教材

ISBN 978-7-121-15343-3

I. ①软… II. ①许… 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(2011)第247057号

责任编辑:韩同平 特约编辑:李佩乾

印 刷:北京季蜂印刷有限公司

装 订:三河市鹏成印业有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×1092 1/16 印张:20.75 字数:531.2千字

印 次:2011年12月第1次印刷

印 数:3000册 定价:36.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

第2版前言

当前,全球软件产业呈现出网络化、服务化和全球化的三大发展趋势。软件产业将成为全球高科技产业发展最主要的推动力,这对中国软件产业的发展和国际化软件人才培养提出了新的挑战。软件工程作为一门指导计算机软件系统开发和维护的工程学科,近年对软件产业的支撑作用凸现,软件工程的应用水平和普及程度关系到我国的国际竞争力。

本书为电子科技大学“软件工程”课程(教育部优秀“软件工程网络课程”、教育部-微软精品课程、四川省精品课程)的配套教材,是作者20多年从事“软件工程”课程教学和软件开发的实践经验的总结。

第2版在继承第1版“内容先进、注重实践、案例导向”特点的基础上,根据软件工程的最新发展,对全书的结构和内容进行了较大的调整:(1)增加了软件自动测试、软件风险管理、面向对象的程序设计等内容;(2)为便于读者更好地掌握面向对象的方法,将原来的第4、5章合并,将OOA和OOD分放在第2、3章介绍,并提供了更加丰富的OOSE的案例。(3)订正了第1版中错漏之处,提高了可读性,使语言更简练,深入浅出,通俗易懂。

全书共12章,在系统介绍软件工程的基本内容、开发及管理技术的基础上,重点介绍面向对象的方法及UML统一建模语言,以及CMM软件成熟度模型、ERP企业资源规划等先进管理技术。内容涵盖了ACM/IEEE联合工作组制定的软件工程教育知识体系,保证了内容的科学性和先进性。其中,第12章软件工程课程设计是一个综合性的设计型实验,旨在培养学生的实践能力及创新能力,并提供了4个采用UML面向对象建模的软件开发实例。

为便于广大读者自学,我们还提供了“软件工程精品课程网站”,包括高质量的全程授课“软件工程”电子课件、中英文两个版本的“软件工程网络课件”、在线自测、电子教材、案例分析、资料查阅、在线讨论及“网上课程设计平台”等丰富的网络资源。建议在学习第2章时,即可开始“软件工程课程设计”,课程设计与课堂教学内容同步进行。以“Learning by doing(做中学)”的先进教学理念为指导,在软件项目的开发实践中学习、深化、应用软件工程理论。

许家珩教授编写第1、2、4、7章,并负责全书统稿,白忠建博士编写第3、5、6、10章,吴磊博士编写第8、9、11、12章。

电子科技大学黄迪明教授认真审阅了全书,并提出了十分宝贵的修改意见,在此表示诚挚的感谢。本书在编写过程中,得到了许多教师和学生的鼓励和支持,曾翎教授、彭德中教授对本书的编写提出了许多有益的意见和建议,在此对所有支持和帮助本书编写的人们表示衷心的感谢。最后,特别要感谢电子工业出版社的韩同平编辑,本书从策划、编写到出版,没有他的信任、鼓励、支持和帮助,是不可能顺利完成的。

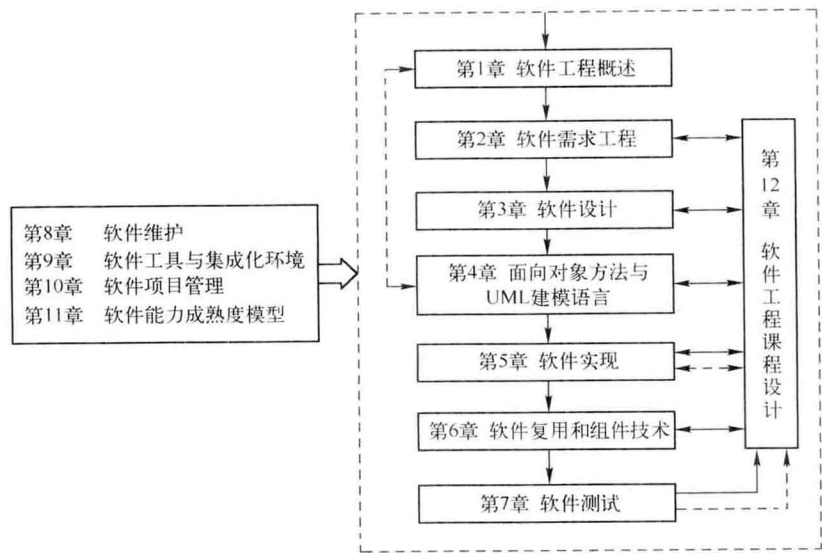
由于作者水平有限,诚恳希望广大读者批评指正!

作者 Email: jiayixu@uestc.edu.cn

作者
于电子科技大学

软件工程课程学习指导

本书的结构如下：



建议学习路径：

1. 学习第 2 章时，即可开始“软件工程课程设计”，课程设计与基本教学内容同步进行。上图中实线箭头表示学习顺序，双向箭头表示执行过程的交叠，即在进行课程设计时，可随时返回进行基本教学内容的学习。

2. 对软件工程方法有一定基础的读者，也可以跳过部分章节，建议按照虚线箭头的方向执行，即在学习第 1 章后，学习第 4 章，并开始课程设计。

3. 第 8 章至第 11 章的部分内容可安排自学，并在“软件工程课程设计”过程中实施。

4. 第 6 章为选学内容。

课程网站：

教育网：<http://222.197.165.195/wlxt/ncourse/se/web/soft/default.aspx>

公网：<http://125.71.228.222/wlxt/ncourse/se/web/soft/default.aspx>

目 录

第 1 章 软件工程概述	1
1.1 软件工程的产生和发展	1
1.1.1 软件工程的发展过程	1
1.1.2 软件危机	2
1.1.3 软件工程的定义及基本原则	3
1.1.4 软件工程研究的内容	3
1.2 软件与软件过程	4
1.2.1 软件的概念和特点	5
1.2.2 软件工程过程	6
1.3 软件过程模型	6
1.4 软件开发方法	10
1.4.1 结构化开发方法	10
1.4.2 原型化开发方法	10
1.4.3 面向对象的开发方法	11
1.5 软件工具与软件开发环境	13
小结	14
习题一	15
第 2 章 软件需求工程	17
2.1 软件需求的基本概念	17
2.1.1 软件需求的任务	17
2.1.2 功能需求与非功能需求	18
2.2 需求工程过程	20
2.3 软件需求获取技术	23
2.4 需求分析与建模	25
2.4.1 结构化分析 (SA) 方法	26
2.4.2 面向对象的分析方法	31
2.5 软件需求案例分析	35
2.5.1 案例 1——医院病房监护系统	35
2.5.2 案例 2——网上拍卖系统	37
小结	39
习题二	39
第 3 章 软件设计	42
3.1 软件设计概述	42
3.2 软件体系结构设计	44
3.2.1 仓库模型	44

3.2.2 分布式结构	45
3.2.3 其他体系结构	52
3.3 模块分解	52
3.3.1 模块分解的目的	53
3.3.2 模块间的关系	53
3.3.3 模块的独立性	54
3.3.4 信息隐蔽	56
3.4 详细设计描述工具	57
3.5 面向对象的设计	58
3.6 用户界面设计	60
3.6.1 用户界面设计的特性与设计任务	60
3.6.2 用户界面设计的基本原则	61
3.6.3 用户界面的基本类型	62
3.6.4 信息输入/输出界面	63
3.6.5 MVC 模式	67
小结	69
习题三	69
第4章 面向对象方法与 UML 建模语言	71
4.1 面向对象方法概述	71
4.1.1 面向对象方法的特点	71
4.1.2 几种典型的面向对象方法	72
4.2 UML 概述	78
4.2.1 UML 的基本概念	79
4.2.2 UML 的图形表示	81
4.3 建立用例模型	83
4.3.1 需求分析与用例建模	84
4.3.2 确定执行者	85
4.3.3 确定用例	86
4.3.4 建立用例之间的关系	88
4.3.5 用例建模实例	89
4.4 建立静态模型	92
4.4.1 类图	92
4.4.2 包图	101
4.5 建立动态模型	102
4.5.1 消息	102
4.5.2 状态图	103
4.5.3 顺序图	106
4.5.4 合作图	108
4.5.5 活动图	110
4.6 建立实现模型	112

4.6.1	组件图	112
4.6.2	部署图	113
4.7	RUP 统一过程及其应用	115
4.7.1	UML 与 RUP 统一过程	115
4.7.2	RUP 的二维开发模型	117
4.7.3	RUP 的迭代开发模式	119
小结		120
习题四		120
第 5 章	软件实现	124
5.1	程序设计语言的选择	124
5.2	结构化程序设计	127
5.3	面向对象程序设计	128
5.4	程序设计风格	132
5.5	算法与程序效率	135
5.6	软件代码审查	137
小结		138
习题五		138
第 6 章	软件复用和组件技术	140
6.1	软件复用概述	140
6.2	软件复用的实施与过程	143
6.3	可复用组件与组件工程	144
6.4	领域工程分析和基于组件的开发	146
6.5	基于组件的软件开发特点	148
6.6	软件组件技术的技术规范	150
6.6.1	CORBA	150
6.6.2	COM	151
6.6.3	EJB	153
6.6.4	Web 服务	155
小结		157
习题六		157
第 7 章	软件测试	158
7.1	软件测试概述	158
7.1.1	软件测试的基本概念	158
7.1.2	软件测试的特点和基本原则	159
7.1.3	软件测试过程	162
7.1.4	静态分析与动态测试	164
7.2	白盒法测试	166
7.3	黑盒法测试	170
7.4	软件测试的策略	174
7.4.1	单元测试	174

7.4.2	集成测试	176
7.4.3	确认测试	179
7.4.4	系统测试	179
7.4.5	α 测试和 β 测试	180
7.4.6	综合测试策略	181
7.5	软件调试	181
7.5.1	软件调试过程	182
7.5.2	软件调试策略	182
7.6	面向对象的测试	184
7.6.1	面向对象测试的特点	185
7.6.2	面向对象测试类型	185
7.6.3	分析模型测试	187
7.6.4	面向对象的测试用例	191
7.7	自动测试	192
7.7.1	自动化测试概述	192
7.7.2	实施自动化测试的前提条件	193
7.7.3	自动化测试过程	195
7.7.4	自动化测试的原则	197
	小结	197
	习题七	198
第 8 章	软件维护	200
8.1	软件维护的基本概念	200
8.2	软件维护的过程	203
8.3	软件维护技术	205
8.4	软件可维护性	206
8.4.1	软件可维护性的定义	206
8.4.2	提高可维护性的方法	207
8.5	逆向工程和再工程	211
	小结	213
	习题八	213
第 9 章	软件工具与集成化环境	215
9.1	软件工具	215
9.1.1	软件开发工具	215
9.1.2	软件维护工具	218
9.1.3	软件管理与支持工具	219
9.2	集成化 CASE 环境	220
9.2.1	概述	220
9.2.2	集成化的 CASE 开发环境的体系结构	222
9.3	软件开发工具——Rational Rose	227
9.3.1	Rose 工具简介	227

9.3.2 业务用例图	228
9.3.3 用例图	229
9.3.4 类图	231
9.3.5 协作图与时序图	232
9.3.6 活动图	234
9.3.7 状态图	234
9.3.8 组件图和部署图	235
小结	237
习题九	237
第 10 章 软件项目管理	238
10.1 软件项目管理概述	238
10.2 软件项目可行性研究	240
10.3 软件项目成本估算技术	242
10.3.1 影响成本估算的因素	242
10.3.2 成本估算模型	244
10.3.3 成本/效益分析	247
10.4 软件项目组织与人员管理	248
10.4.1 软件开发进度计划	248
10.4.2 人员配备与组织	250
10.4.3 软件开发小组与软件生产率	250
10.5 项目风险管理	251
10.5.1 软件项目风险管理概述	251
10.5.2 软件项目风险管理过程	252
10.5.3 风险管理的理论和模型	254
10.6 软件质量保证	256
10.7 企业资源规划	257
10.7.1 资源管理发展过程	257
10.7.2 ERP 系统的管理思想	258
10.7.3 应用 ERP 与企业的关系	259
10.7.4 ERP 应用成功的标志	261
10.7.5 SAP ERP 简介	261
小结	263
习题十	263
第 11 章 软件能力成熟度模型	265
11.1 CMM 概述	265
11.2 CMM 的内部结构	268
11.3 CMM 的应用	270
11.4 CMM 的实施与评估	272
11.4.1 软件过程评估的必要性	272
11.4.2 软件过程评估参考模型	274

11.4.3 CMM 评估的执行步骤	275
11.4.4 软件企业如何实施 CMM	278
11.4.5 CMM 与 ISO 9000 标准	280
11.5 软件能力成熟度模型集成	281
小结	284
习题十一	284
第 12 章 软件工程课程设计	286
12.1 课程设计的目的和要求	286
12.2 课程设计步骤及安排	286
12.3 案例分析	288
12.3.1 案例一: ATM 系统	288
12.3.2 案例二: 网上拍卖系统	294
12.3.3 案例三: 会议管理系统	300
12.3.4 案例四: 仓库信息管理系统	310
参考文献	319

第 1 章 软件工程概述

1.1 软件工程的产生和发展

软件工程 (Software Engineering) 是在克服 20 世纪 60 年代末所出现的“软件危机”的过程中逐渐形成与发展的。自 1968 年在北大西洋公约组织举行的软件可靠性学术会议 (NATO) 上正式提出“软件工程”的概念以来, 在 40 多年的时间里, 软件工程在理论和实践两方面都取得了长足的进步。

软件工程是一门指导计算机软件系统开发和维护的工程学科, 是一门新兴的边缘学科, 它涉及计算机科学、工程学科、管理学、数学等多个学科, 其研究范围广, 不仅包括软件系统的开发方法和技术、管理技术, 还包括软件工具、环境及软件开发的规范。

软件是信息化的核心, 国民经济、国防建设、社会发展及人民生活都离不开软件。软件产业关系到国家经济发展和文化安全, 体现了国家综合实力, 是决定 21 世纪国际竞争地位的战略产业。随着我国信息化建设的深入发展, 软件工程对促进信息产业发展和信息化建设的作用凸现。

因此, 大力推广应用软件工程的开发技术及管理技术, 提高软件工程的应用水平, 对促进我国软件产业与国际接轨, 推动软件产业的迅速发展将起着十分重要的作用。

1.1.1 软件工程的发展过程

软件工程的产生和发展是与软件的发展过程紧密相关的。自从第一台电子计算机诞生以来, 就开始了软件的生产。从“软件工程”的概念提出至今, 它的发展已经历了四个重要阶段:

(1) 第一代软件工程

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、规模及复杂度的迅速增大, 生产作坊的方式已不能够适应软件生产的需要, 出现了所谓“软件危机 (Software Crisis)”, 其表现为软件生产效率低, 大量质量低劣的软件涌入市场甚至在软件开发过程中夭折。由于“软件危机”的不断扩大, 软件产业濒临瘫痪。

为了克服“软件危机”, 在著名的 NATO 会议上第一次提出了“软件工程”的术语, 将软件开发纳入了工程化的轨道, 基本形成了软件工程的概念、框架、技术和方法。这一阶段又称为传统的软件工程。

(2) 第二代软件工程

20 世纪 80 年代中期, 以 Smalltalk 为代表的面向对象的程序设计语言相继推出, 面向对象的方法与技术得到发展; 从 20 世纪 90 年代起, 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, 出现了许多面向对象的开发方法的流派, 面向对象的方法逐渐成为软件开发的主流方法。所以这一阶段又称为对象工程。

(3) 第三代软件工程

随着软件规模和复杂度的不断增大, 开发人员也随之增多, 开发周期也相应增长, 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，是软件开发和维护中的管理和支持能力。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) 第四代软件工程

20 世纪 90 年代起，基于组件（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存的可复用组件组装完成，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。软件复用技术及组件技术的发展，对克服软件危机提供了一条有效途径，将这一阶段称为组件工程。

1.1.2 软件危机

1. 软件危机的产生

软件危机的出现是由于软件的规模越来越大，复杂度不断增加，而软件需求量也不断增大，生产作坊式的软件开发模式及技术已不能满足软件发展的需要。

软件开发过程是一种高密集度的脑力劳动，需要投入大量的人力、物力和财力；由于软件开发的模式及技术不能适应软件发展的需要，致使大量质量低劣的软件产品涌向市场，有的甚至在开发过程中就夭折了。国外在开发一些大型软件系统时，遇到了许多困难，有的系统最终彻底失败了；有的系统则比原计划推迟了好多年，而且费用大大超过了预算；或者系统功能不符合用户的需求；也无法进行修改维护。典型的例子有：

IBM 公司的 OS/360，共约 100 万条指令，花费了 5000 个人年，经费达数亿美元，而结果却令人沮丧，错误多达 2000 个以上，系统根本无法正常运行。OS/360 系统的负责人 Brooks 这样描述开发过程的困难和混乱：“像巨兽在泥潭中作垂死挣扎，挣扎得越猛，泥浆就沾得越多，最后没有一个野兽能够逃脱淹没在泥潭中的命运……”。

1967 年前苏联“联盟一号”载人宇宙飞船，由于其软件忽略一个小数点的错误，导致返航时打不开降落伞，当进入大气层时因摩擦力太大而烧毁，造成机毁人亡的巨大损失。

还有，可以称为 20 世纪世界上最精心设计，并花费了巨额投资的美国阿波罗登月飞行计划的软件，也仍然没有避免出错。例如，阿波罗 8 号太空飞船由于计算机软件的一个错误，造成存储器的一部分信息丢失；阿波罗 14 号在飞行的 10 天中，出现了 18 个软件错误。

2. 软件危机的表现

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、程序人员无限度地增加等各个方面，以至于形成人们难以控制软件开发的局面。

软件危机主要表现在两个方面：

- ① 软件产品质量低劣，甚至在开发过程中就夭折。
- ② 软件生产效率低，不能满足需要。

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，克服软件危机刻不容缓。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。

但距离彻底克服软件危机这个软件工程的最终目标，任重道远，还需要软件工作者付出长期艰苦的努力。

1.1.3 软件工程的定义及基本原则

1. 软件工程的定义

自 1968 年提出软件工程这个术语以来，对于软件工程就有了各种各样的定义，但是它们的基本思想都是强调在软件开发过程中应用工程化原则的重要性。

例如，1983 年，IEEE（国际电气与电子工程师协会）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、运作、维护的系统化的、有规范的、可量化的方法之应用，即是对软件的工程化应用。

从软件工程的定义可见，软件工程是一门指导软件开发的工程学科，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，把经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”。

2. 软件工程的基本原则

过去，软件工程的基本原则是抽象、模块化、清晰的结构、精确的设计规格说明。但今天的认识已经发生了很大的变化，现已提出的软件工程 4 条基本原则是：

① 必须认识软件需求的变动性，以便采取适当措施来保证产品能最好地满足用户要求。在软件设计中，通常要考虑模块化、抽象与信息隐蔽、局部化、一致性等原则。

② 稳妥的设计方法将大大方便软件开发，以达到软件工程的目标。软件工具与环境对软件设计的支持来说，颇为重要。

③ 软件工程项目的质量与经济开销取决于对它所提出的支撑质量与效用。

④ 只有在强调对软件过程进行有效管理的情况下，才能实现有效的软件工程。

近年来，印度的软件产业迅速发展，其成功的经验是，严格按照国际规范进行科学管理。在本教材中，虽然主要讨论软件开发技术，只在第 10 章讨论软件管理技术，但软件管理仍然是软件开发成功的关键，将重点介绍当前质量管理的国际规范 CMM（软件成熟度的度量）。

1.1.4 软件工程研究的内容

软件工程是一门新兴的边缘学科，涉及的学科多，研究的范围广。归结起来软件工程研究的主要内容有以下 4 个方面：方法与技术、工具及环境、管理技术、标准与规范。

① 软件开发方法，主要讨论软件开发的各种方法及其工作模型，它包括多方面的任务，如软件系统需求分析、总体设计，以及如何构建良好的软件结构、数据结构及算法设计等，同时讨论具体实现的技术。

② 软件工具为软件工程方法提供支持，研究计算机辅助软件工程，建立软件工程环境。

③ 软件工程管理，是指对软件工程全过程的控制和管理，包括计划安排、成本估算、项目管理、软件质量管理。

④ 软件工程标准化与规范化，使得各项工作有章可循，以保证软件生产效率和软件质量的提高。软件工程标准可分为 4 个层次：国际标准、行业标准、企业规范和项目规范。

必须要强调的是，随着人们对软件系统研究的逐渐深入，软件工程所研究的内容也不是一成不变的。

此外，按照 ACM 和 IEEE-CS 发布的软件工程知识体系 (SWEBOK) 定义的软件工程学科的内涵，软件工程研究的内容由 10 个知识域构成。

(1) 软件需求 (Software Requirements)。软件需求涉及需求抽取、需求分析、建立需求规格说明和确认等活动，还涉及建模、经济与时间可行性分析。

(2) 软件设计 (Software Design)。设计是软件工程最核心的内容。其主要活动有软件体系结构设计、软件详细设计。涉及软件体系结构、组件、接口，以及系统或组件的其他特征，还涉及软件设计质量分析和评估、软件设计的符号、软件设计策略和方法等。

(3) 软件构造 (Software Construction)。通过编码、单元测试、集成测试、调试、确认等活动，生成可用的、符合设计功能的软件。并要求控制和降低程序复杂性。

(4) 软件测试 (Software Testing)。测试是软件生存周期的重要部分，涉及测试的标准、测试技术、测试度量和测试过程。

(5) 软件维护 (Software Maintenance)。软件产品交付后，需要改正软件的缺陷，提高软件性能或其他属性，使软件产品适应新的环境。软件维护是软件进化的继续。基于服务的软件维护越来越受到重视。

(6) 软件配置管理 (Software Configuration Management)。为了系统地控制配置变更，维护整个系统生命周期中配置的一致性和可追踪性，必须按时间管理软件的不同配置，包括配置管理过程的管理、软件配置鉴别、配置管理控制、配置管理状态记录、配置管理审计、软件发布和交付管理等。

(7) 软件工程管理 (Software Engineering Management)。运用管理活动，如计划、协调、度量、监控、控制和报告，确保软件开发和维护是系统的、规范的、可度量的。它涉及基础设施管理、项目管理、度量和控制计划三个层次。

(8) 软件工程过程 (Software Engineering Process)。软件工程过程关注软件过程的定义、实现、评估、测量、管理、变更、改进，以及过程和产品的度量。

(9) 软件工程工具和方法 (Software Engineering Tools and Methods)。软件开发工具是以计算机为基础辅助软件生存周期过程的实施。软件工具的种类很多，如：需求工具、设计工具、构造工具、测试工具、维护工具、配置管理工具、工程管理工具、工程过程工具、软件质量工具等。

软件工程方法支持软件工程活动，典型的有结构化方法、面向数据方法、面向对象方法、原型化方法及基于数学的形式化方法等。

(10) 软件质量 (Software Quality)。软件质量管理贯穿整个软件生存周期，涉及软件质量需求、软件质量度量、软件属性检测、软件质量管理技术和过程等。

必须要强调的是，随着人们对软件系统研究的逐渐深入，软件工程所研究的内容也在不断更新和发展。

1.2 软件与软件过程

软件工程是在软件生产中采用工程化的方法，并采用一系列科学的、现代化的方法和技术来开发软件的。这种工程化的思想贯穿软件开发和维护的全过程。

为了进一步学习有关软件工程的方法和技术，先介绍软件、软件生存期及软件工程过程

这几个重要的概念。

1.2.1 软件的概念和特点

1. 软件及其特点

“软件就是程序，开发软件就是编写程序”是一个错误观点。这种错误观点的长期存在，影响了软件工程的正常发展。

事实上，正如 **Boehm** 指出的：软件是程序，以及开发、使用和维护程序所需的所有文档。它是由应用程序、系统程序、面向用户的文档及面向开发者的文档四部分构成的。

软件的特点如下。

- ① 软件是一种逻辑实体，不是具体的物理实体。
- ② 软件产品的生产主要是研制。
- ③ 软件具有“复杂性”，其开发和运行常受到计算机系统的限制。
- ④ 软件成本昂贵，其开发方式目前尚未完全摆脱手工生产方式。
- ⑤ 软件不存在磨损和老化问题，但存在退化问题。

图 1.1 是硬件的失效率曲线，它是一个“U 型”曲线（浴盆曲线），说明硬件随着使用时间的增加，失效率急剧上升。

图 1.2 所描述的软件失效率曲线，它没有“U 型”曲线的右半翼，表明软件随着使用时间的增加，失效率降低；因为软件不存在磨损和老化问题，但存在退化问题。

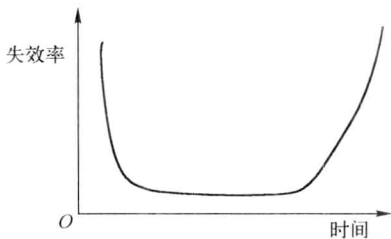


图 1.1 硬件失效率曲线

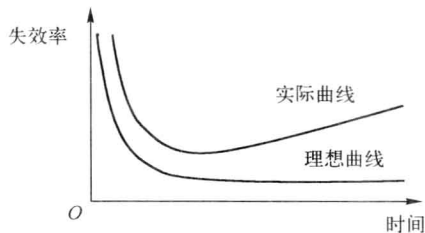


图 1.2 软件失效率曲线

2. 软件生存期

软件生命周期 (SDLD) 是指一个从用户需求开始，经过开发、交付使用，在使用中不断地增补修订，直至软件报废的全过程，亦称软件生存期 (Life Cycle)。

软件生命周期分为以下阶段：

- ① 可行性研究和项目开发计划。该阶段必须要回答的问题是“要解决的问题是什么”。
- ② 需求分析。该阶段的任务不是具体地解决问题，而是准确地确定“软件系统必须做什么”，确定软件系统必须具备哪些功能。
- ③ 概要设计。概要设计就是设计软件的结构，该结构由哪些模块组成，这些模块的层次结构是怎样的，这些模块的调用关系是怎样的，每个模块的功能是什么。同时还要设计该项目的应用系统的总体数据结构和数据库结构，即应用系统要存储什么数据，这些数据是什么样的结构，它们之间有什么关系等。
- ④ 详细设计。即对每个模块完成的功能进行具体描述，要把功能描述变为精确的、结构化的过程描述。

⑤ 编码。该阶段把每个模块的控制结构转换成计算机可接受的程序代码，即写成以某特定程序设计语言表示的“源程序”。

⑥ 测试。它是保证软件质量的重要手段，其主要方式是在设计测试用例的基础上检验软件的各个组成部分。测试分为模块测试、组装测试、确认测试等。

⑦ 维护。软件维护是软件生存期中时间最长的阶段。已交付的软件投入正式使用后，便进入软件维护阶段，它可以持续几年甚至几十年。

在大部分文献中将生存期划分为 5 个阶段，即要求定义、设计、编码、测试及维护。其中要求定义阶段包括可行性研究和项目开发计划及需求分析，设计阶段包括概要设计和详细设计。

为了描述软件生存期的活动，提出了多种生存期模型，如瀑布模型、循环模型、螺旋模型、喷泉模型、智能模型等。

1.2.2 软件工程过程

软件工程过程是指在软件工具的支持下，所进行的一系列软件工程活动。通常包括以下 4 类基本过程：

- ① 软件规格说明：规定软件的功能及其运行环境。
- ② 软件开发：产生满足规格说明的软件。
- ③ 软件确认：确认软件能够完成客户提出的要求。
- ④ 软件演进：为满足客户的变更要求，软件必须在使用的过程中演进。

软件工程过程具有可理解性、可见性（过程的进展和结果可见）、可靠性、可支持性（易于使用 CASE 工具支持）、可维护性、可接受性（为软件工程师接受）、开发效率和健壮性（抵御外部意外错误的能力）等特性。

软件工程有方法、工具和过程三个要素。软件工程方法是研究软件开发是“如何做”的；软件工具是研究支撑软件开发方法的工具，为方法的运用提供自动或者半自动的支撑环境。软件工具的集成环境，又称为计算机辅助软件工程（Computer Aided Software Engineering, CASE）；软件工程过程则是指将软件工程方法与软件工具相结合，实现合理、及时地进行软件开发的目的，为开发高质量软件规定各项工作任务的工作步骤。如图 1.3 所示，在软件工程的三要素中，软件过程将人员、方法与规范、工具和管理有机结合，形成一个能有效控制软件开发质量的运行机制。

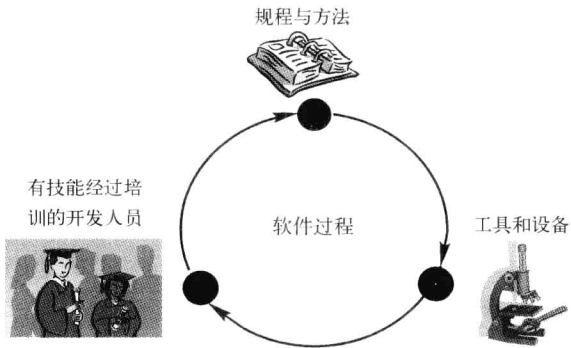


图 1.3 软件工程过程

1.3 软件过程模型

软件过程模型也称为软件生存期模型或软件开发模型，是描述软件过程中各种活动如何