



普通高等教育“十三五”规划教材



高等学校规划教材

软件工程——方法与实践

(第3版)

◎ 主编 许家玲



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十三五”规划教材
高等学校规划教材

软件工程——方法与实践

(第3版)

许家珩 主编

许家珩 白忠建 吴磊 许卡 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书第3版在继承第2版特点的基础上做了较大修改,内容涵盖了IEEE新发布的软件工程知识体系指南SWEBOK V3.0版的知识域,删除了较陈旧的内容,新增了如云计算模型、敏捷开发测试等国内外软件工程发展的新技术。在系统介绍软件工程基础知识的基础上,重点介绍了软件需求和软件设计两个重要开发阶段,并将面向对象的方法及UML统一建模技术贯穿于各章,对面向对象的分析、设计、测试的方法做了详细介绍。同时还对CMM软件成熟度模型、风险管理及团队建设等先进的软件管理技术进行了介绍。

本书内容注重科学性、先进性,强调实践性,理论紧密联系实际是本书的一大特色;本书不仅提供了丰富的软件开发实例和素材,还用一章专门讨论了如何进行综合性、设计型的软件工程课程设计。

本书可作为高等院校计算机、软件工程及信息类专业本科生及研究生“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. —3版. —北京:电子工业出版社,2019.1

ISBN 978-7-121-35099-3

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(2018)第218738号

责任编辑:韩同平 特约编辑:李佩乾

印 刷:涿州市京南印刷厂

装 订:涿州市京南印刷厂

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:18.25 字数:584千字

版 次:2007年9月第1版

2019年1月第3版

印 次:2019年1月第1次印刷

定 价:49.90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, 88258888。

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

本书咨询联系方式:(010) 88254525, hantp@phei.com.cn。

第3版前言

随着互联网技术的应用和发展,网络软件及安全性已成为全球关注的热点,尤其近年继云计算、物联网后出现的大数据热潮,其潜在的巨大价值受到各界的广泛关注。软件工程作为研究、应用工程化方法创造、构建和维护高质量软件的学科,在大数据时代对培养软件工程人才起着不可替代的作用。

本书第3版在继承第2版特点的基础上,根据IEEE最新发布的、被称为软件工程发展史里程碑的软件工程知识体系指南SWEBOK V3,以及近年国内外软件工程技术的新发展,对内容和结构都做了较大的调整和修改。删除了较陈旧的内容,新增了如云计算模型,敏捷开发及测试等软件工程新技术。第3版的主要特点为:

1. 内容丰富、具有先进性。本版的内容涵盖了SWEBOK V3的15个知识域,保证了内容的科学性和先进性。在系统介绍软件工程的基本知识的基础上,以软件建模思想为主线,重点对软件需求工程和软件设计这两个软件开发的重要阶段的内容重新进行了修订。

2. 注重实践、案例导向。本书特别注重理论与实践相结合,在系统介绍软件工程基本理论的同时,不仅提供了丰富的软件开发案例,还引入了“Learning by doing”这一行之有效的先进教学理念,专题介绍与课堂教学同步进行的综合性、设计型的软件工程课程设计,让学生在“做中学”,在软件项目的开发实践中学习、深化、应用软件工程理论。

3. 提供了丰富的优质教学资源。本课程是教育部-微软“软件工程精品课程”、教育部新世纪网络课程、四川省精品课程,四川省精品资源共享课程。教学资源包括高质量的“软件工程”电子课件、中英文两个版本的“软件工程网络课程”、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,为课程的学习提供了丰富的多媒体网络教学资源。

第3版的结构也以便于学习、掌握UML建模技术为指导思想进行了较大的调整。全书共分10章,第1章系统介绍软件工程的基本概念,第2章介绍UML统一建模语言和应用及RUP统一过程。第3、4章讨论了需求和设计这两个软件开发重要阶段的任务和原则,并对需求获取、需求分析建模、软件体系结构的常用模式、用户界面设计等关键技术进行了详细的讨论。第5、6章讨论软件构造和实现,以及测试技术。第7、8章介绍了软件维护及软件项目管理,第9章对CMM软件能力成熟度模型及应用进行了讨论。第10章详细介绍了开设软件工程课程设计的目的、要求、设计步骤,并介绍了可视化建模工具Rational Rose的使用,还提供了多个采用UML面向对象建模的软件开发实例。

教学过程中,可先只学习第2章的部分基础,在后续章节的学习中再进一步学习、研究和应用UML建模技术。

许家珩教授编写第1、3、6章,并负责全书统稿,白忠建副教授编写第4、5、8章、吴磊副教授编写第7、9、10章,许卡高工编写第2章。

电子科技大学黄迪明教授、四川大学彭德中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，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。

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

课程网站: <http://222.197.183.243/se/web/soft/default.aspx>

作 者
于电子科技大学

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 软件危机与软件工程	(1)
1.1.2 软件工程的定义及基本原则	(3)
1.1.3 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 软件工程过程及产品	(6)
1.3 软件过程模型	(6)
1.3.1 瀑布模型	(6)
1.3.2 增量模型	(7)
1.3.3 螺旋模型	(7)
1.3.4 喷泉模型	(8)
1.3.5 原型模型	(8)
1.3.6 智能模型	(9)
1.4 软件开发方法	(9)
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)
1.4.4 敏捷软件的开发	(12)
1.5 软件工具与集成化开发环境	(15)
1.5.1 软件工具的发展过程	(15)
1.5.2 软件工具	(16)
1.5.3 集成化 CASE 环境	(18)
小结	(19)
习题 1	(19)
第 2 章 面向对象方法与 UML 建模语言	(21)
2.1 面向对象方法概述	(21)
2.1.1 面向对象方法的特点	(21)
2.1.2 几种典型的面向对象方法	(22)
2.2 UML 概述	(23)
2.2.1 UML 的基本概念	(24)
2.2.2 UML 的图形表示	(25)
2.3 建立用例模型	(28)
2.3.1 需求分析与用例建模	(28)
2.3.2 确定执行者和用例	(29)
2.3.3 用例建模实例	(31)

2.4	建立静态模型	(33)
2.4.1	类图	(33)
2.4.2	包图	(41)
2.5	建立动态模型	(42)
2.5.1	消息	(42)
2.5.2	状态图	(43)
2.5.3	顺序图	(46)
2.5.4	合作图	(47)
2.5.5	活动图	(50)
2.6	建立实现模型	(52)
2.6.1	组件图	(52)
2.6.2	部署图	(53)
2.7	RUP 统一过程及其应用	(54)
2.7.1	UML 与 RUP 统一过程	(54)
2.7.2	RUP 的二维开发模型	(56)
2.7.3	RUP 的迭代开发模式	(58)
	小结	(59)
	习题二	(59)
第3章	软件需求工程	(63)
3.1	软件需求的基本概念	(63)
3.1.1	软件需求的任务	(63)
3.1.2	功能需求与非功能需求	(64)
3.2	需求工程过程	(66)
3.2.1	需求工程的基本活动	(66)
3.2.2	获取需求	(66)
3.2.3	需求分析与建模	(67)
3.2.4	需求的有效性验证	(68)
3.2.5	需求管理	(69)
3.3	软件需求获取技术	(70)
3.3.1	面谈法	(70)
3.3.2	问卷调查法	(70)
3.3.3	需求专题讨论会	(71)
3.3.4	原型法获取需求	(71)
3.3.5	面向用例的方法	(72)
3.4	需求分析与建模技术	(73)
3.4.1	需求分析方法	(73)
3.4.2	结构化分析(SA)方法	(74)
3.4.3	面向对象的分析(OOA)方法	(79)
3.5	软件需求案例分析	(82)
3.5.1	案例1——用SA法建立需求模型	(82)
3.5.2	案例2——用OOA法建立需求模型	(84)
	小结	(87)

习题二	(88)
第4章 软件设计	(91)
4.1 软件设计概述	(91)
4.1.1 软件设计在开发阶段中的重要性	(91)
4.1.2 软件设计阶段的任务	(92)
4.1.3 软件设计的目标	(92)
4.1.4 软件设计过程	(92)
4.2 软件体系结构设计	(93)
4.2.1 系统结构性模型——集中式系统模型	(93)
4.2.2 系统结构性模型——层次结构模型	(94)
4.2.3 系统结构性模型——分布式系统模型	(95)
4.2.4 控制模型	(100)
4.2.5 模块分解	(101)
4.3 详细设计描述工具	(104)
4.4 面向对象的设计	(105)
4.4.1 面向对象设计的准则	(106)
4.4.2 面向对象设计的基本任务	(106)
4.5 用户界面设计	(108)
4.5.1 用户界面设计的特性与设计任务	(108)
4.5.2 用户界面设计的基本原则	(109)
4.5.3 用户界面的基本类型	(110)
4.5.4 用户交互设计	(111)
4.6 MVC 设计模式	(114)
4.7 软件设计实例	(115)
小结	(118)
习题四	(118)
第5章 软件构造	(121)
5.1 程序设计语言的选择	(121)
5.1.1 程序设计语言的分类	(121)
5.1.2 高级程序设计语言的基本组成	(122)
5.1.3 程序设计语言选择准则	(123)
5.2 程序设计方法	(123)
5.2.1 结构化程序设计	(123)
5.2.2 面向对象程序设计	(125)
5.3 程序设计风格	(127)
5.3.1 源程序文件	(128)
5.3.2 语句构造方法	(129)
5.3.3 数据说明方法	(129)
5.3.4 输入/输出技术	(130)
5.4 算法与程序效率	(130)
5.4.1 算法转换过程中的指导原则	(130)
5.4.2 影响效率的因素	(131)

5.5	软件代码审查	(133)
5.6	软件复用	(134)
5.6.1	软件复用的级别	(134)
5.6.2	软件复用过程	(135)
5.6.3	可复用构件	(135)
5.6.4	基于复用的开发过程	(136)
5.6.5	构件的组装和复用	(137)
5.7	软件构造实例	(137)
5.7.1	实现环境	(137)
5.7.2	系统编码实现	(138)
5.7.3	用户界面	(143)
	小结	(143)
	习题五	(143)
第6章	软件测试	(145)
6.1	软件测试概述	(145)
6.1.1	软件测试的基本概念	(145)
6.1.2	软件测试的特点和基本原则	(147)
6.1.3	软件测试过程	(149)
6.1.4	静态分析与动态测试	(151)
6.2	白盒法测试	(153)
6.2.1	语句覆盖	(153)
6.2.2	判定覆盖	(154)
6.2.3	条件覆盖	(155)
6.2.4	判定-条件覆盖	(155)
6.2.5	条件组合覆盖	(156)
6.3	黑盒法测试	(157)
6.3.1	等价分类法	(157)
6.3.2	边界值分析法	(158)
6.3.3	错误推测法	(158)
6.3.4	因果图法	(159)
6.4	软件测试的策略	(160)
6.4.1	单元测试	(160)
6.4.2	集成测试	(162)
6.4.3	确认测试	(164)
6.4.4	系统测试	(165)
6.4.5	α 测试和 β 测试	(166)
6.4.6	综合测试策略	(167)
6.5	软件调试	(167)
6.5.1	软件调试过程	(167)
6.5.2	软件调试策略	(168)
6.6	面向对象的测试	(170)
6.6.1	面向对象测试的特点	(170)

6.6.2	面向对象测试模型	(171)
6.6.3	面向对象测试类型	(171)
6.6.4	分析模型测试	(173)
6.6.5	面向对象的测试用例	(177)
6.7	自动化测试	(177)
6.7.1	自动化测试概述	(178)
6.7.2	实施自动化测试的前提条件	(179)
6.7.3	自动化测试过程	(180)
6.7.4	自动化测试的原则	(182)
6.7.5	敏捷测试	(182)
	小结	(183)
	习题六	(184)
第7章	软件维护	(186)
7.1	软件维护的基本概念	(186)
7.1.1	软件维护的目的	(186)
7.1.2	软件维护的分类	(186)
7.1.3	软件维护的特性	(187)
7.1.4	软件维护的代价	(188)
7.2	软件维护的过程	(189)
7.2.1	软件维护的组织和维护过程管理	(189)
7.2.2	维护工作的流程	(190)
7.3	软件维护技术	(191)
7.4	软件可维护性	(192)
7.4.1	软件可维护性的定义	(192)
7.4.2	提高可维护性的方法	(193)
7.5	软件再工程技术	(197)
7.5.1	逆向工程	(197)
7.5.2	软件再工程过程	(197)
7.5.3	再工程的成本及效益分析	(198)
7.5.4	再工程的风险分析	(199)
	小结	(199)
	习题七	(199)
第8章	软件项目管理	(201)
8.1	软件项目管理概述	(201)
8.1.1	软件项目管理的特点	(201)
8.1.2	软件项目管理的主要活动	(202)
8.2	软件项目可行性研究	(203)
8.2.1	可行性研究的任务	(203)
8.2.2	可行性研究报告	(203)
8.3	软件项目计划	(204)
8.3.1	软件项目计划内容	(204)
8.3.2	软件开发进度计划	(205)

8.4	软件项目成本估算技术	(207)
8.4.1	影响成本估算的因素	(207)
8.4.2	成本估算模型	(208)
8.4.3	成本/效益分析	(211)
8.5	软件项目人力资源管理	(211)
8.5.1	软件团队建设	(212)
8.5.2	团队人员的选择	(213)
8.6	软件项目风险管理	(213)
8.6.1	软件项目风险管理概述	(213)
8.6.2	软件项目风险管理过程	(214)
8.6.3	风险管理的理论和模型	(216)
8.7	软件质量保证	(218)
8.8	软件配置管理	(219)
8.8.1	软件配置管理的基本概念	(219)
8.8.2	软件配置管理的活动	(219)
	小结	(222)
	习题八	(222)
第9章	软件能力成熟度模型	(223)
9.1	CMM 概述	(223)
9.1.1	软件过程成熟度的基本概念	(223)
9.1.2	软件过程的成熟度等级	(224)
9.2	CMM 的结构	(225)
9.2.1	关键过程域	(226)
9.2.2	关键实践	(227)
9.2.3	共同特性	(227)
9.3	CMM 的实施与评估	(227)
9.3.1	软件过程评估的必要性	(228)
9.3.2	软件过程评估及参考模型	(229)
9.3.3	软件过程改进	(231)
9.3.4	CMM 评估的执行步骤	(231)
9.3.5	软件企业如何实施 CMM	(234)
9.3.6	CMM 与 ISO 9000 标准	(236)
9.4	软件能力成熟度模型集成	(237)
9.4.1	CMMI 的产生与发展	(237)
9.4.2	CMMI 的模型	(237)
9.4.3	CMMI 的评估	(239)
9.4.4	CMMI 与 CMM 的比较	(240)
	小结	(241)
	习题九	(241)
第10章	软件工程课程设计	(243)
10.1	课程设计的目的和要求	(243)
10.2	课程设计步骤及安排	(243)

10.3 可视化建模工具 Rational Rose..... (245)

10.3.1 Rose 工具简介..... (245)

10.3.2 绘制业务用例图..... (246)

10.3.3 绘制用例图..... (247)

10.3.4 绘制类图..... (248)

10.3.5 绘制协作图与顺序图..... (249)

10.3.6 绘制活动图..... (250)

10.3.7 绘制状态图..... (251)

10.3.8 绘制构件图和部署图..... (251)

10.4 案例分析..... (252)

10.4.1 案例一：ATM 系统..... (252)

10.4.2 案例二：网上拍卖系统..... (257)

10.4.3 案例三：会议管理系统..... (263)

10.4.4 案例四：仓库信息管理系统..... (272)

参考文献..... (278)

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 年代末国际上所出现的“软件危机”的过程中逐渐形成与发展的。自 1968 年北大西洋公约组织 (NATO) 所举行的软件可靠性学术会议上为克服软件危机, 正式提出“软件工程”的概念以来, 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、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, 体现了一个国家的综合实力, 是决定 21 世纪国际竞争地位的战略产业。尤其是随着互联网技术的迅速发展, 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 软件危机与软件工程

1. 软件危机

20 世纪 60 年代末期, 随着软件的规模越来越大, 复杂度不断增加, 软件需求量也不断增大, 而当时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: “像巨兽在泥潭中做垂死挣扎, 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

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。

3. 软件工程的观念

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。

1968年,在北大西洋公约组织所召开的可靠性会议上,首次提出了“软件工程”的概念,即借鉴工程化的方法来开发软件。自该会议以来,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。

从“软件工程”的概念提出至今,软件工程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年代末所出现的“软件危机”,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,使软件产业濒临瘫痪。

为克服“软件危机”,在著名的NATO软件可靠性会议上第一次提出了“软件工程”的术语以来,将软件开发纳入了工程化的轨道,基本形成了软件工程的观念、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,演化成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,即是对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年代起至今,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, 软件工程一直以来都缺乏统一的定义, 很多学者、组织机构都分别给出了自己认可的定义。

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、可定量的方法之应用, 即是对软件的工程化应用。

ISO 9000 对软件工程过程的定义是: 软件工程过程是输入转化为输出的一组彼此相关的资源和活动。

BarryBoehm 则定义为: 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。

对于软件工程的各种各样的定义, 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。

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。即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

软件工程的基本原则是随着软件工程的发展而变化的。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, 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。

1.1.3 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几方面: 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, 包括具体实现的技术。

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、软件质量管理等。

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。

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。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过程。
- 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 是硬件失效率的“U 形”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 形”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。

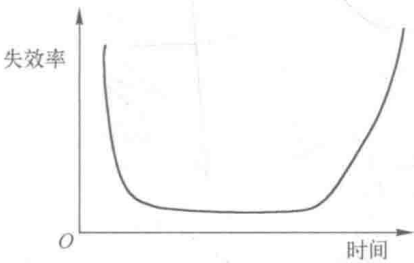


图 1.1 硬件失效率曲线

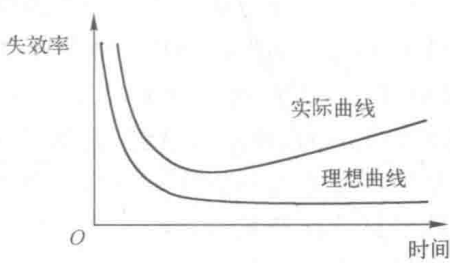


图 1.2 软件失效率曲线

2. 软件生存期

软件生存期（Life Cycle），又称生命周期（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。

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“软件系统要解决的问题是什么”。
- ② 需求分析。该阶段的任务是，通过分析准确地确定“软件系统必须做什么”，即软件系统必须具备哪些功能。
- ③ 概要设计。也称总体设计。主要任务是确定软件体系结构，划分子系统模块及确定模块之间的关系。并确定系统的数据结构和进行界面设计。
- ④ 详细设计。即对每个模块完成的功能、算法进行具体描述，要把功能描述变为精确的、结构化的过程描述。
- ⑤ 软件构造。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，即编写以某特定程序设计语言表示的“源代码”。
- ⑥ 测试。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