



免费提供
电子教案

高等院校规划教材
软件工程系列

软件工程

第2版

瞿中 吴渝 常庆丽 王永昆 编著



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·软件工程系列

软 件 工 程

第2版

瞿 中 吴 渝 常庆丽 王永昆 编著



机 械 工 业 出 版 社

本书从实用的角度出发,根据教育部高教司主持编审的《中国计算机科学与技术学科教程 2002》中对软件工程的要求组织编写,并参照美国 ACM 和 IEEE Computing Curricula 2005 教程关于软件工程的描述。本书详细介绍了软件工程、软件开发过程、软件计划、需求分析、总体设计、详细设计、编码、软件测试、软件维护、软件工程标准化和软件文档、软件工程质量、软件工程项目管理、开发实例等知识。每章配有习题,以指导读者深入地进行学习。

本书内容丰富,结构合理,既可作为高等院校计算机专业课程的教材或教学参考书,也可作为通信、电子信息、自动化等相关专业的计算机课程教材,还可供软件工程师、软件项目管理者 and 应用软件开发人员阅读参考。

图书在版编目 (CIP) 数据

软件工程/瞿中等编著. —2 版, —北京:机械工业出版社, 2011. 6
(高等院校规划教材·软件工程系列)
ISBN 978 - 7 - 111 - 33949 - 6

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

中国版本图书馆 CIP 数据核字 (2011) 第 054959 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:唐德凯

责任印制:乔宇

三河市宏达印刷有限公司印刷

2011 年 5 月第 2 版·第 1 次印刷

184mm × 260mm · 23.25 印张 · 576 千字

0001 - 3000 册

标准书号: ISBN 978 - 7 - 111 - 33949 - 6

定价: 42.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

门户网: <http://www.cmpbook.com>

销售一部: (010) 68326294

教材网: <http://www.cmpedu.com>

销售二部: (010) 88379649

读者购书热线: (010) 88379203

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等教材系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

当今，软件工程已成为计算机科学中的一个重要分支，并且一直是一个非常热门的研究领域。软件工程是指导计算机软件开发与维护的工程学科，它采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间验证过的管理技术和当前能够得到的最好技术方法结合起来，以便经济地开发高质量的软件并有效地维护它。软件工程学是计算机专业一门非常重要的专业课程，它的研究范围非常广泛，包括技术、方法、工具和管理等许多方面。软件工程学也是一门迅速发展新兴学科，有很多新的技术和方法。严格遵循软件工程的方法，可以大大提高软件的开发效率和成功率，减少软件开发和维护中的问题。

本书共 11 章，从实用角度出发，讲述了软件工程的基本原理、概念、技术和方法。本书的主要内容有软件工程、软件开发过程、软件计划、需求分析、总体设计、详细设计、编码、软件测试、软件维护、软件工程标准化和软件文档、软件工程质量、软件工程项目管理、开发实例等。在编写的过程中注重理论与应用相结合，不仅对软件的分析、设计、开发及维护过程进行了全面地讲述，而且配有丰富的实例，每章还提供典型习题。

本书由有着丰富的教学经验和雄厚的软件开发能力的教师在参阅大量国内外有关软件工程的教材和资料后编写而成。全书由瞿中、吴渝、常庆丽、王永昆编写，研究生马庆伟、张庆庆、高腾飞、李梦露、关兴等参与了文字录入，并对书中的实例及图表做了大量的工作。本书的顺利出版，要感谢领导和老师给予的大力支持和帮助，也得到了计算机教育界许多同行的关心，在此一并致谢。

书中文字通俗易懂、概念清晰、深入浅出、实例丰富、实用性强，既可作为高等学校计算机专业课程的教材或教学参考书，也可作为通信、电子信息、自动化等相关专业的计算机课程教材，还可供软件工程师、软件项目管理者 and 应用软件开发人员阅读参考。为了便于读者学习，本书提供配套的电子教案和习题答案，如有需要可从机械工业出版社教材服务网（www.cmpedu.com）下载或与编者联系，编者的联系方式：quzhong@hotmail.com。

目前，国内外有关软件工程技术与设计方面的资料很多，新理论、新技术层出不穷。由于时间仓促，加上软件工程发展迅速和编者水平有限，书中难免存在不妥之处，恳请读者批评指正，以便进一步完善。

作 者

目 录

出版说明

前言

第1章 概论	1
1.1 软件的概念	1
1.1.1 软件的定义以及特点	1
1.1.2 软件技术的发展阶段	2
1.1.3 软件的分类	3
1.2 软件危机	6
1.2.1 软件危机的定义	6
1.2.2 软件危机产生的原因	7
1.2.3 解决软件危机的途径	8
1.3 软件工程	8
1.3.1 软件工程的定义和研究对象	8
1.3.2 软件工程的基本原理	9
1.3.3 软件工程项目的目标	10
1.3.4 软件工程的基本原则	10
1.4 软件生存周期	11
1.4.1 软件生存周期的概念	11
1.4.2 软件开发工具	12
1.5 软件开发过程模型	13
1.6 软件开发方法简述	20
1.7 软件工程的最新发展动向	21
1.8 经典例题讲解	22
小结	24
习题	24
第2章 结构化分析	26
2.1 可行性研究	26
2.1.1 问题定义	26
2.1.2 可行性研究的任务	26
2.1.3 可行性研究的步骤	29
2.2 系统流程图	30
2.2.1 系统流程图的符号	30
2.2.2 系统流程图举例	31
2.2.3 分层	32
2.3 制订软件计划	32
2.3.1 确定软件计划	32

2.3.2 复审软件计划	34
2.4 成本/效益分析	34
2.4.1 成本估算技术	34
2.4.2 成本/效益分析的方法	37
2.5 需求分析的概念和任务	38
2.5.1 需求分析的概念	38
2.5.2 需求分析的层次	38
2.5.3 需求分析的目标和任务	39
2.5.4 需求分析的原则	40
2.5.5 需求规格说明书	41
2.5.6 评审	43
2.6 获取需求的方法	45
2.6.1 存在问题	45
2.6.2 常用方法	45
2.6.3 需求分析的过程	47
2.6.4 结构化需求分析方法	47
2.7 传统的软件建模	48
2.7.1 软件建模	48
2.7.2 数据模型的实体-联系图建立	48
2.7.3 功能模型、行为模型的建立及数据字典	49
2.7.4 构建数据流图实例	51
2.7.5 快速原型法分析实例	53
2.8 经典例题讲解	54
小结	61
习题	61
第3章 结构化设计	63
3.1 总体设计的任务及过程	63
3.1.1 总体设计的任务	63
3.1.2 总体设计的过程	63
3.2 总体设计的原理	65
3.2.1 软件结构和过程	65
3.2.2 模块设计	65
3.2.3 结构设计	72
3.3 设计准则	74
3.4 总体设计的常用方法及工具	75
3.4.1 面向数据流的设计方法	75
3.4.2 总体设计中的工具	79
3.5 模块结构设计	82
3.6 数据存储设计	82
3.7 模型-视图-控制器框架	83

3.7.1 MVC 模式	83
3.7.2 MVC 中的模型类、视图类和控制类	83
3.7.3 MVC 的实现	84
3.8 总体设计说明书编写规范	85
3.9 详细设计阶段的任务	87
3.10 结构化详细设计的原则	88
3.11 结构化详细设计的方法和工具	88
3.11.1 详细设计的方法	88
3.11.2 详细设计的工具	89
3.11.3 详细设计工具的选择	97
3.12 详细设计规格说明与复审	98
3.12.1 详细设计说明	98
3.12.2 设计复审	99
3.13 面向数据结构的结构化设计方法	99
3.14 Jackson 程序设计方法	99
3.14.1 Jackson 方法的基本思想	100
3.14.2 Jackson 结构图	100
3.14.3 Jackson 方法的设计技术	100
3.15 Warnier 程序设计方法	100
3.15.1 Warnier 方法的基本思想	100
3.15.2 Warnier 方法的设计技术	101
3.16 基于组件的设计方法	101
3.17 界面设计	102
3.17.1 用户界面设计的一般原则和步骤	102
3.17.2 字符界面设计	104
3.17.3 菜单设计	105
3.17.4 对话框设计	106
3.17.5 多窗口界面设计	107
3.18 软件体系结构	107
3.18.1 软件体系结构的兴起	107
3.18.2 软件体系结构的概念	108
3.18.3 软件体系结构的现状及发展方向	109
3.18.4 软件体系结构的描述方法	111
3.19 软件体系结构与操作系统	113
3.19.1 分层结构	113
3.19.2 微内核结构	115
3.20 经典例题讲解	117
小结	128
习题	128

第4章 编码及测试	131
4.1 程序设计语言	131
4.1.1 程序设计语言的发展及分类	131
4.1.2 选择程序设计语言的标准	134
4.2 程序设计风格	135
4.2.1 源程序文档化	135
4.2.2 数据说明	136
4.2.3 表达式和语句结构	136
4.2.4 输入和输出	137
4.3 程序效率	138
4.4 编程安全	139
4.5 结构化程序设计方法	141
4.6 程序的复杂性及度量	143
4.6.1 代码行度量法	143
4.6.2 McCabe 度量法	143
4.6.3 Halstead 度量法	145
4.7 软件测试基础	146
4.7.1 软件测试的意义	146
4.7.2 基本概念	146
4.7.3 软件测试的目的、任务、原则和研究对象	148
4.7.4 软件测试的发展历史及趋势	150
4.8 软件测试的方法	150
4.8.1 静态测试和动态测试	150
4.8.2 黑盒测试和白盒测试	151
4.9 软件测试的步骤	156
4.9.1 单元测试	156
4.9.2 集成测试	160
4.9.3 确认测试	164
4.9.4 系统测试	165
4.9.5 验收测试	168
4.10 软件测试	171
4.10.1 软件测试角色	171
4.10.2 软件测试的需求规格说明	172
4.10.3 软件测试设计说明	177
4.11 测试设计和管理	178
4.11.1 错误曲线	178
4.11.2 测试用例设计	179
4.12 软件测试工具	187
4.12.1 自动软件测试的优点	187
4.12.2 测试工具分类	188

4.12.3 自动测试的相关问题	190
4.13 经典例题讲解	191
小结	194
习题	195
第5章 软件维护及软件再工程	198
5.1 软件维护的概念	198
5.1.1 软件维护的定义	198
5.1.2 软件维护的分类	198
5.1.3 软件维护成本	200
5.2 软件维护的特点	201
5.3 软件维护过程	202
5.4 软件维护的步骤	205
5.5 软件的可维护性	206
5.5.1 影响软件可维护性的因素	206
5.5.2 软件可维护性度量	208
5.5.3 提高软件的可维护性方法	208
5.6 逆向工程和再工程	210
5.6.1 预防性维护	210
5.6.2 软件的逆向工程和再工程	211
5.6.3 软件再工程的过程	211
5.6.4 软件再工程的方法	212
小结	213
习题	213
第6章 面向对象方法学	215
6.1 面向对象方法学概述	215
6.1.1 面向对象方法学的要点	215
6.1.2 面向对象方法学的优点	216
6.2 面向对象的概念	218
6.2.1 对象	218
6.2.2 其他概念	220
6.3 面向对象建模	222
6.4 对象模型	223
6.4.1 类图的基本符号	223
6.4.2 表示关系的符号	224
6.5 功能模型	228
6.5.1 用例图	228
6.5.2 用例建模	229
6.6 3种模型之间的关系	230
6.7 经典例题讲解	231
小结	232

习题	233
第7章 面向对象分析	234
7.1 面向对象分析的基本过程	234
7.1.1 概述	234
7.1.2 三个子模型与五个层次	234
7.2 需求陈述的书写	235
7.3 建立对象模型	236
7.3.1 确定类与对象	236
7.3.2 确定关联	237
7.3.3 划分主题	237
7.3.4 确定属性	237
7.3.5 识别继承关系	238
7.3.6 反复修改	238
7.4 建立动态模型	238
7.5 建立功能模型	240
7.5.1 画出基本系统模型图	240
7.5.2 画出功能级数据流图	240
7.5.3 描述处理框功能	240
7.6 定义服务	240
7.7 经典例题讲解	241
小结	244
习题	244
第8章 面向对象设计	246
8.1 面向对象设计的准则	246
8.1.1 模块化	246
8.1.2 抽象化	246
8.1.3 信息隐藏和封装	247
8.1.4 对象的高内聚和弱耦合	247
8.1.5 可扩充性	247
8.1.6 可重用性	247
8.2 启发规则	247
8.2.1 设计结果应该清晰易懂	247
8.2.2 一般——特殊结构的深度应适当	248
8.2.3 设计简单的类	248
8.2.4 使用简单的协议	248
8.2.5 使用简单的服务	249
8.2.6 把设计变动减至最小	249
8.3 软件重用	249
8.3.1 概述	249
8.3.2 类构件	250

8.3.3 软件重用的效益	250
8.4 系统分解	251
8.4.1 子系统之间的两种交互方式	251
8.4.2 组织系统的两种方案	252
8.4.3 设计系统的拓扑结构	252
8.5 设计问题域子系统	253
8.6 设计人一机交互子系统	254
8.6.1 设计人一机交互界面的准则	254
8.6.2 设计人一机交互子系统的策略	255
8.7 设计任务管理子系统	256
8.7.1 确定事件驱动型任务	256
8.7.2 确定时钟驱动型任务	256
8.7.3 确定优先任务和关键任务	256
8.7.4 确定协调任务	257
8.7.5 确定资源需求	257
8.8 设计数据管理子系统	257
8.8.1 选择数据存储管理模式	257
8.8.2 设计数据管理子系统	258
8.9 设计类中的服务	259
8.9.1 确定类中应有的服务	259
8.9.2 涉及实现服务的方法	260
8.10 设计关联	260
8.10.1 关联的遍历	260
8.10.2 实现单向关联	260
8.10.3 实现双向关联	261
8.10.4 链属性的实现	262
8.11 设计优化	262
8.11.1 确定优先级	262
8.11.2 提高效率的几项技术	262
8.11.3 调整继承关系	263
8.12 经典例题讲解	263
小结	265
习题	265
第9章 面向对象实现	266
9.1 面向对象语言	266
9.1.1 面向对象语言的优点	266
9.1.2 面向对象语言的技术特点	266
9.1.3 选择面向对象语言的原则	267
9.2 面向对象程序设计风格	268
9.2.1 提高可重用性	268

9.2.2 提高可扩充性	269
9.2.3 提高健壮性	269
9.3 测试策略	270
9.3.1 面向对象测试模型	270
9.3.2 面向对象分析的测试	270
9.3.3 面向对象设计的测试	270
9.3.4 面向对象编程的测试	271
9.3.5 面向对象的单元测试	271
9.3.6 面向对象的集成测试	271
9.3.7 面向对象的系统测试	272
9.4 经典例题讲解	272
小结	274
习题	274
第10章 软件工程标准化和软件文档	276
10.1 软件工程标准化的概念	276
10.1.1 什么是软件工程标准化	276
10.1.2 软件工程标准化的意义	276
10.1.3 软件工程标准化的类型	277
10.2 软件工程标准的制定与推行	278
10.3 软件工程标准的层次和体系框架	279
10.3.1 软件工程标准的层次	279
10.3.2 软件工程过程中版本控制与变更控制处理过程	280
10.3.3 中国的软件工程标准化工作	281
10.4 ISO 9000 国际标准简介	282
10.5 软件文档	285
10.5.1 软件文档的作用和分类	285
10.5.2 对软件文档编制的质量要求	287
10.5.3 软件文档的管理和维护	288
10.6 经典例题讲解	289
小结	290
习题	291
第11章 软件工程质量及项目管理	293
11.1 软件质量特性	293
11.1.1 软件质量的定义	293
11.1.2 软件质量的特性	294
11.2 软件质量的度量模型	295
11.2.1 软件度量和软件质量的度量	295
11.2.2 软件质量的度量模型	298
11.3 软件质量保证	301
11.3.1 什么是软件质量保证	301

11.3.2	软件质量保证的主要任务	301
11.3.3	软件质量保证策略	302
11.4	技术评审	302
11.5	软件质量管理体系	305
11.5.1	软件产品质量管理的特点	305
11.5.2	软件质量管理的指导思想	306
11.5.3	软件质量管理体系	307
11.6	软件项目管理	311
11.6.1	软件项目管理的特点	312
11.6.2	软件项目管理的主要职能	312
11.6.3	软件项目管理的主要内容	312
11.7	基于 CASE 技术的开发工具简介	313
11.8	软件项目管理活动	314
11.8.1	计划项目	314
11.8.2	项目组织	315
11.8.3	控制项目	317
11.8.4	终结项目	317
11.9	成本估算	317
11.10	计划和组织	318
11.10.1	项目计划的制定	318
11.10.2	项目组人员管理原则	319
11.10.3	人员组织与管理	320
11.11	进度计划	321
11.11.1	制定开发进度计划	321
11.11.2	Gantt 图与时间管理	322
11.11.3	工程网络与关键路径	322
11.11.4	项目进度跟踪与控制	325
11.12	风险管理	326
11.12.1	风险识别与分类	326
11.12.2	风险评估与分析	329
11.12.3	风险策划与管理	330
11.12.4	风险规避与监控	332
11.13	项目管理认证体系 IPMP 与 PMP	332
11.13.1	IPMP 概况	332
11.13.2	PMP 简介	333
11.13.3	我国目前的项目管理认证体系的发展状况	334
11.14	经典例题讲解	334
	小结	337
	习题	338

第 12 章 开发实例	340
12.1 项目论证和计划	340
12.1.1 系统调查	340
12.1.2 新系统的总体功能需求和性能要求	340
12.1.3 系统开发的框架	341
12.2 可行性分析	341
12.3 需求分析	342
12.3.1 数据流分析	342
12.3.2 系统流程图	343
12.3.3 数据字典	344
12.4 总体设计	345
12.4.1 功能模块图	345
12.4.2 层次方框图	345
12.4.3 IPO 图	346
12.4.4 系统的功能结构图	346
12.4.5 人事管理工作的工作流程模型图	346
12.4.6 系统数据库关系说明图	347
12.5 详细设计	347
12.6 系统实现	348
12.7 测试与维护	349
12.7.1 测试结果	350
12.7.2 系统维护	350
小结	353
习题	354
参考文献	356

第1章 概 论

本章要点

- 软件、软件工程的概念以及开发的主要原则
- 软件开发过程的模型以及开发方法
- 软件工程标准及开发文档的编制管理方法
- 软件工程发展的新动向

1.1 软件的概念

1.1.1 软件的定义以及特点

软件（Software）在计算机系统中与硬件（Hardware）相互依存，包括程序（Program）、相关数据（Data）及其说明文档（Document）。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；说明文档是与程序开发维护和使用有关的各种图文数据。软件的发展历史不过四五十年，人们对软件的认识也经历了一个由浅到深的过程。在计算机系统发展的初期，硬件通常用来执行一个单一的程序，而程序是为一个特定的目的编制的。早期硬件的通用性很强，软件的通用性却是很弱。大多数软件是由使用该软件的个人或机构研制的，软件往往有局限性。早期的软件开发也没有什么系统的方法可以遵循，软件设计是在某个开发者的头脑中完成的。而且，软件成品除了源代码外几乎没有软件说明书等文件。

随着计算机的普及，软件变得越来越复杂，规模变得越来越大，人与人、人与机器间的相互沟通变得更加困难。文档在软件开发与维护过程中体现出越来越高的价值，甚至超出了软件产品本身。因此，“软件就是程序”的观念已经落后，需要对其进行重新认识。

软件的特点如下。

1) 软件是一种抽象的逻辑实体。人们无法看到其具体形态，只能通过观察、分析、思考、判断等方式了解它的特性和功能。

2) 软件是一种通过人们的智力活动，把知识与技术转化为信息的一种产品，是在研制、开发中被创造出来的。在软件的开发过程中没有明显的制造过程，不像硬件那样，一旦研制成功，可以重复制造，在制造过程中进行质量控制，以保证产品质量。软件研制成功后只是大量复制同一内容的副本，所以软件的质量主要取决于软件的开发过程，在软件的制造过程中几乎不会引入新的质量问题。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，所以要保证软件质量，必须在软件开发的过程中加强管理。同时由于软件的复制是非常容易的事情，必须在技术上和法律上采取有力的措施，严格控制任意复制软件的行为。

3) 在软件的运行和使用期间，没有硬件那样的机器磨损、老化问题。但是软件存在退

化问题，需要维护。软件的退化问题主要是因为在软件的生存周期中，为了使它能够处理之前没有发现的故障，能够适应硬件、软件环境的变化以及用户的新要求，必须多次修改软件，每次修改都会引入新的不可知的错误，连续多次后，软件失效率就会提高，图 1-1 描述了硬件与软件使用过程中产生的失效率曲线。

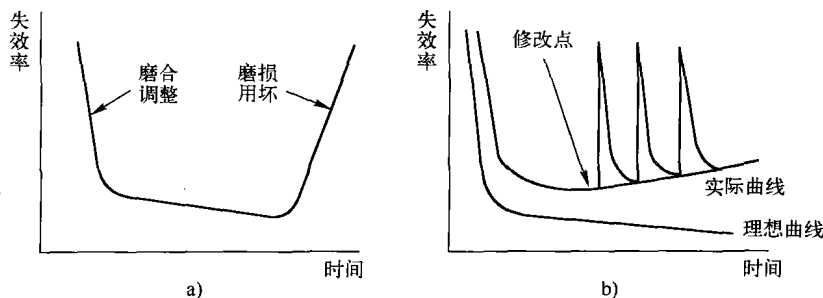


图 1-1 失效率曲线

a) 硬件失效率曲线 b) 软件失效率曲线

4) 软件的开发和运行受到计算机硬件系统的限制。在软件的开发和运行中必须以硬件提供的条件为基础。有的软件依赖于某种硬件系统，有的依赖于某种操作系统，这给软件的使用带来很多不便。为了消除这种依赖关系，在软件开发中提出了软件移植问题，并将软件的可移植性作为衡量软件质量的一个重要因素。

5) 软件开发至今尚未摆脱手工开发方式，很多软件仍然是“定制”的。这使得软件的开发效率受到很大限制。近年来出现的软件复用技术、自动生成技术和其他一些有效的软件开发工具或软件开发环境，在一定程度上提高了软件的开发效率，但在软件项目中采用的比率较低。对软件工作而言，开发工作是一种高强度的脑力工作，并不轻松。

6) 软件的开发是一个复杂的过程。软件的复杂性可能来自于它所反映的实际问题的复杂性，也可能来自于程序逻辑结构的复杂性。因而，管理是软件开发过程中必不可少的内容。

7) 软件的成本相当高。软件开发需要投入大量的、高强度的脑力劳动，成本很高，风险也非常大，现在软件的开销已大大超过了硬件的开销。

8) 相当多软件的开发涉及社会因素。如许多软件的开发和运行涉及机构、体制、管理方式等方面的问题，同时也涉及人们的观念和心里等问题。

1.1.2 软件技术的发展阶段

17 世纪 60 年代，Augusta Ada 为 Lovelace Charles Babbage 的分析机 (Analytic Machine) 编写流程，其中包括计算三角函数、级数相乘、伯努利函数等。在 20 世纪 40 年代末，随着 ENIAC (Electronic Numerical Integrator and Calculator) 的问世，以写软件为职业的人开始出现，他们多是经过训练的数学家和电子工程师。到了 20 世纪 60 年代，美国大学里开始出现专门教授人们编写软件的专业，并且对该专业毕业的大学生、研究生授予计算机专业的学位。伴随着信息产业的迅速发展，软件对人类社会的作用显得越来越重要，人们对软件的认识也更为深刻。

在发展过程中，软件技术主要经历了以下 4 个发展阶段。