



中国高等职业技术教育研究会推荐
高职高专计算机专业规划教材

软件工程与 项目管理

COMPUTER
COMPUTER

主编 王素芬
参编 栾好利 朱克敌
杨政 张楠



西安电子科技大学出版社
<http://www.xduph.com>

中国高等职业技术教育研究会推荐

高职高专计算机专业规划教材

软件工程与项目管理

主 编 王素芬

副主编 栾好利

参 编 朱克敌 杨 政 张 楠



西安电子科技大学出版社

2010

内 容 简 介

本书全面、系统地介绍了软件工程的观念、原理和典型的技术方法。

全书共分 12 章。第 1 章概括介绍了软件工程的发展和基本原理,还对具有代表性的 CASE 工具进行了简要介绍,讨论了软件工程职业道德规范,并对软件工程的发展趋势进行了展望;第 2 章讨论了软件过程的基本活动和常用的软件开发方法,并介绍了典型的软件过程模型和微软公司的软件开发过程实例;第 3 章主要介绍了常用软件项目立项的方法、可行性分析、软件项目团队的建立以及软件项目立项文档的编写;第 4 章主要介绍了需求分析的过程、方法和软件需求分析文档的编写;第 5 章、第 6 章分别介绍了软件的总体设计和详细设计;第 7~9 章分别介绍了软件编码、测试与调试、维护;第 10 章介绍了面向对象的方法学;第 11 章介绍了统一建模语言(UML);第 12 章介绍了软件项目管理。为加强对学生实践能力的培养,各章最后大都设置了“实战训练”和“讨论”环节。

本书可作为高职高专学校软件工程课程的教材,也可作为软件开发人员的参考书。

图书在版编目(CIP)数据

软件工程与项目管理/王素芬主编. —西安:西安电子科技大学出版社,2010.3

高职高专计算机专业规划教材

ISBN 978-7-5606-2360-3

I. 软… II. 王… III. ① 软件工程—高等学校:技术学校—教材

② 软件开发—项目管理—高等学校:技术学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2009)第 208762 号

策 划 杨 璠

责任编辑 阎 彬 杨 璠

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2010 年 3 月第 1 版 2010 年 3 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 19.25

字 数 449 千字

印 数 1~3000 册

定 价 27.00 元

ISBN 978 - 7 - 5606 - 2360 - 3/TP • 1191

XDUP 2652001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

序

进入 21 世纪以来,高等职业教育呈现出快速发展的形势。高等职业教育的发展,丰富了高等教育的体系结构,突出了高等职业教育的类型特色,顺应了人民群众接受高等教育的强烈需求,为现代化建设培养了大量高素质技能型专门人才,对高等教育大众化作出了重要贡献。目前,高等职业教育在我国社会主义现代化建设事业中发挥着越来越重要的作用。

教育部 2006 年下发了《关于全面提高高等职业教育教学质量的若干意见》,其中提出了深化教育教学改革,重视内涵建设,促进“工学结合”人才培养模式改革,推进整体办学水平提升,形成结构合理、功能完善、质量优良、特色鲜明的高等职业教育体系的任务要求。

根据新的发展要求,高等职业院校积极与行业企业合作开发课程,根据技术领域和就业岗位群任职要求,参照相关职业资格标准,改革课程体系和教学内容,建立突出职业能力培养的课程标准,规范课程教学的基本要求,提高课程教学质量,不断更新教学内容,而实施具有工学结合特色的教材建设是推进高等职业教育改革发展的重要任务。

为配合教育部实施质量工程,解决当前高职高专精品教材不足的问题,西安电子科技大学出版社与中国高等职业技术教育研究会在前三轮联合策划、组织编写“计算机、通信电子、机电及汽车类专业”系列高职高专教材共 160 余种的基础上,又联合策划、组织编写了新一轮“计算机、通信、电子类”专业系列高职高专教材共 120 余种。这些教材的选题是在全国范围内近 30 所高职高专院校中,对教学计划和课程设置进行充分调研的基础上策划产生的。教材的编写采取在教育部精品专业或示范性专业的高职高专院校中公开招标的形式,以吸收尽可能多的优秀作者参与投标和编写。在此基础上,召开系列教材专家编委会,评审教材编写大纲,并对中标大纲提出修改、完善意见,确定主编、主审人选。该系列教材以满足职业岗位需求为目标,以培养学生的应用技能为着力点,在教材的编写中结合任务驱动、项目导向的教学方式,力求在新颖性、实用性、可读性三个方面有所突破,体现高职高专教材的特点。已出版的第一轮教材共 36 种,2001 年全部出齐,从使用情况看,比较适合高等职业院校的需要,普遍受到各学校的欢迎,一再重印,其中《互联网实用技术与网页制作》在短短两年多的时间里先后重印 6 次,并获教育部 2002 年普通高校优秀教材奖。第二轮教材共 60 余种,在 2004 年已全部出齐,有的教材出版一年多的时间里就重印 4 次,反映了市场对优秀专业教材的需求。前两轮教材中有十几种入选国家“十一五”规划教材。第三轮教材 2007 年 8 月之前全部出齐。本轮教材预计 2009 年全部出齐,相信也会成为系列精品教材。

教材建设是高职高专院校教学基本建设的一项重要工作。多年来,高职高专院校十分重视教材建设,组织教师参加教材编写,为高职高专教材从无到有,从有到优、到特而辛勤工作。但高职高专教材的建设起步时间不长,还需要与行业企业合作,通过共同努力,出版一大批符合培养高素质技能型专门人才要求的特色教材。

我们殷切希望广大从事高职高专教育的教师,面向市场,服务需求,为形成具有中国特色和高职教育特点的高职高专教材体系作出积极的贡献。

中国高等职业技术教育研究会会长

2007 年 6 月



高职高专计算机专业规划教材

编审专家委员会

- 主任：** 温希东 （深圳职业技术学院副校长，教授）
- 副主任：** 徐人凤 （深圳职业技术学院电子与通信工程学院副院长，高工）
刘中原 （上海第二工业大学计算机与信息学院副院长，副教授）
李卓玲 （沈阳工程学院信息工程系主任，教授）
- 委员：** （按姓氏笔画排列）
- 丁桂芝 （天津职业大学电子信息工程学院院长，教授）
马宏锋 （兰州工业高等专科学校计算机工程系副主任，副教授）
王 军 （武汉交通职业学院信息系副主任，副教授）
王 雷 （浙江机电职业技术学院计算机应用工程系主任，高工）
王养森 （南京信息职业技术学院计算机科学与技术系主任，高工）
王趾成 （石家庄职业技术学院计算机系主任，高工）
汤 勇 （成都职业技术学院国际软件学院副院长，副教授）
朱小平 （广东科学技术职业学院计算机学院副院长，副教授）
齐志儒 （东北大学东软信息学院计算机系主任，教授）
孙街亭 （安徽职业技术学院教务处处长，副教授）
张 军 （石家庄职业技术学院计算机系，高工）
李成大 （成都电子机械高等专科学校计算机工程系副主任，副教授）
苏传芳 （安徽电子信息职业技术学院计算机科学系主任，副教授）
苏国辉 （黎明职业大学计算机系副主任，讲师）
汪临伟 （九江职业技术学院电气工程系主任，副教授）
汪清明 （广东轻工职业技术学院计算机系副主任，副教授）
杨文元 （漳州职业技术学院计算机工程系副主任，副教授）
杨志茹 （株洲职业技术学院信息工程系副主任，副教授）
胡昌杰 （湖北职业技术学院计算机科学与技术系副主任，副教授）
聂 明 （南京信息职业技术学院软件学院院长，副教授）
章忠宪 （漳州职业技术学院计算机工程系主任，副教授）
睦碧霞 （常州信息职业技术学院软件学院院长，副教授）
董 武 （安徽职业技术学院电气工程系副主任，副教授）
蒋方纯 （深圳信息职业技术学院软件工程系主任，副教授）
鲍有文 （北京联合大学信息学院副院长，教授）

前 言

软件工程采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理方法和当前最先进的软件开发技术结合起来，应用到软件开发和维护过程中，以解决软件危机问题，并生产出无故障的、及时交付的、在预算之内的和满足用户需求的软件产品。软件工程是培养学生从单纯的程序设计型人才向软件开发型人才发展的重要课程，是计算机科学与技术及软件专业学生的必修课，同时也是软件从业人员从程序员向软件工程师和更高层次职位发展的必须具备的专业知识和方法。

软件工程的教育培养目标是让受教育者了解和掌握软件开发中的方法学和工程学知识，并应用于实践。软件工程是一门实践性很强的学科，很多知识是软件开发最佳实践的总结。初次学习软件工程的学生一般没有丰富的软件开发实践经验，如果仅学习书本上的理论知识点，将很难领略软件工程的思想精髓，达不到令人满意的教学效果。本书在讲授理论的同时，加强了对学生实践能力的培养，将一个完整的案例贯穿于全书，并在各章的最后大都增设了“实战训练”和“讨论”环节。

全书共分12章。第1章概括介绍了软件工程的发展和基本原理，对具有代表性的CASE工具进行了简要介绍，讨论了软件工程职业道德规范，并对软件工程的发展趋势进行了展望；第2章讨论了软件过程的基本活动和常用的软件开发方法，并介绍了典型的软件过程模型和微软公司的软件开发过程实例；第3章主要介绍了常用软件项目立项的方法、可行性分析、软件项目团队的建立以及软件项目立项文档的编写；第4章主要介绍了需求分析的过程、方法和软件需求分析文档的编写；第5章、第6章分别介绍了软件的总体设计和详细设计；第7~9章分别介绍了软件编码、测试与调试、维护；第10章介绍了面向对象的方法学；第11章介绍了统一建模语言(UML)；第12章介绍了软件项目管理。每一章后面都配有相应的习题，用以帮助学生消化和理解本章的知识。

本书的第1、2、3、5、9章由王素芬编写，第6、10章由杨政编写，第4章由王素芬和杨政共同编写，第7章由张楠编写，第8章由栾好利编写，第11、12章由朱克敌编写。

在本书的编写过程中，沈阳工程学院信息工程系主任李卓玲教授提出了许多很好的建议，在此表示衷心的感谢。

由于编者水平有限，书中难免存在不足之处，恳请读者批评指正。

编 者

2009年11月

目 录

第1章 概述	1	2.2.6 V型模型	22
1.1 软件	1	2.2.7 形式化方法模型	22
1.1.1 软件的特点及分类	1	2.2.8 组合模型	23
1.1.2 软件的发展	3	2.3 软件开发方法简述	23
1.1.3 软件危机及其表现	4	2.3.1 结构化的开发方法	23
1.1.4 软件危机产生的主要原因	4	2.3.2 面向数据结构的开发方法	24
1.1.5 解决软件危机的途径	5	2.3.3 面向对象的开发方法	24
1.2 软件工程	5	2.3.4 原型化开发方法	24
1.2.1 软件工程的	5	2.4 案例 微软公司的软件开发过程	24
1.2.2 软件工程的三要素	5	2.4.1 微软开发过程管理的基本原则	24
1.3 软件生命周期	6	2.4.2 微软公司的软件开发过程模型	25
1.4 软件工程的	8	2.5 实战训练	26
1.5 软件工程的开发原则	9	本章小结	26
1.6 软件工程方法学	9	习题2	26
1.7 计算机辅助软件工程	10	第3章 软件项目立项	28
1.8 软件工程与其他相关学科的关系	11	3.1 软件项目立项方法	28
1.9 软件工程职业道德规范	12	3.1.1 委托开发项目	28
1.10 软件工程发展趋势	13	3.1.2 自主开发项目	29
1.11 实战训练	13	3.2 软件可行性分析	29
本章小结	13	3.2.1 可行性分析的内容	29
习题1	14	3.2.2 可行性分析的步骤	29
第2章 软件过程	16	3.3 软件项目规模成本估算	30
2.1 软件过程的概念	16	3.3.1 代码行估算技术	30
2.1.1 软件过程的定义	16	3.3.2 功能点估算法	31
2.1.2 软件过程的基本活动	16	3.3.3 类比估算法	31
2.1.3 软件过程的制品	16	3.3.4 自下而上估算法	31
2.2 典型的软件过程模型	17	3.3.5 专家估算法	32
2.2.1 瀑布模型	17	3.4 成本—效益分析	32
2.2.2 增量模型	18	3.5 制定软件项目开发计划	34
2.2.3 快速原型模型	19	3.5.1 软件项目开发计划书的内容	34
2.2.4 螺旋模型	20	3.5.2 项目计划编制过程	36
2.2.5 喷泉模型	21	3.5.3 项目计划内容确定	38

3.6 软件项目立项文档	39	5.4 结构化设计	84
3.7 软件项目团队的建立	45	5.4.1 系统结构图中的模块	84
3.7.1 团队定义	45	5.4.2 变换型数据流与变换型系统结构	85
3.7.2 项目组织形式	45	5.4.3 事务流与事务型系统结构	86
3.7.3 团队建设	47	5.4.4 变换映射	86
3.8 实战训练	48	5.4.5 事务映射	88
本章小结	50	5.4.6 “黑盒”技术的使用	88
习题 3	50	5.5 数据库设计及设计原则	88
第 4 章 需求分析	52	5.5.1 数据库设计	88
4.1 需求分析概述	52	5.5.2 数据库设计的原则	90
4.1.1 需求的概念	52	5.6 接口设计	92
4.1.2 需求的层次	53	5.6.1 接口设计的描述	92
4.2 需求分析的目标和任务	54	5.6.2 用户界面设计的主要问题	92
4.3 需求分析的原则与获取需求的方法	55	5.6.3 人机界面设计过程	94
4.3.1 需求分析的原则	55	5.6.4 人机界面设计实现原则及典型案例	96
4.3.2 获取需求的方法	56	5.7 总体设计说明书的编写	97
4.4 需求分析的过程	57	5.8 实战训练	97
4.5 需求分析的方法	57	本章小结	98
4.5.1 结构化分析方法	58	习题 5	99
4.5.2 原型分析方法	63	第 6 章 软件的详细设计	101
4.6 软件需求分析文档	66	6.1 详细设计的任务	101
4.6.1 软件需求规格说明和需求评审	66	6.2 详细设计的原则	101
4.6.2 教务管理系统需求规格说明书	68	6.3 详细设计的方法	102
4.7 实战训练	69	6.4 详细设计可采用的工具	102
本章小结	71	6.5 详细设计说明书	109
习题 4	71	6.6 实战训练	110
第 5 章 软件的总体设计	75	本章小结	113
5.1 总体设计的目标和任务	75	习题 6	113
5.1.1 总体设计的目标	75	第 7 章 软件编码	116
5.1.2 总体设计的任务和步骤	76	7.1 软件编码的目的	116
5.2 设计的概念和原则	77	7.2 程序设计语言	116
5.2.1 模块化	77	7.2.1 程序设计语言的分类	116
5.2.2 抽象	78	7.2.2 程序设计语言特性的比较	117
5.2.3 逐步求精	78	7.2.3 程序设计语言的选择	119
5.2.4 信息隐藏和局部化	78	7.3 结构化程序设计	119
5.2.5 模块独立性	79	7.3.1 结构化程序设计的原则	120
5.3 描绘软件结构的图形工具	81	7.3.2 程序设计——自顶向下，逐步求精	120
5.3.1 HIPO 图	81	7.3.3 数据结构的合理化	122
5.3.2 系统结构图	83		

7.4 编码风格	122	第9章 软件维护	179
7.4.1 程序的内部文档	122	9.1 软件维护的概念	179
7.4.2 数据说明	129	9.2 软件维护的特点	179
7.4.3 语句结构	129	9.3 软件维护的类型	181
7.4.4 输入和输出	130	9.4 软件维护的步骤	183
7.5 程序编码优化技术	131	9.5 软件的可维护性	187
7.5.1 程序优化	131	9.6 软件维护的副作用	188
7.5.2 程序优化方法	132	9.7 软件重用	189
7.5.3 网络优化	139	本章小结	190
7.6 代码评审和版本控制	139	习题 9	190
7.6.1 代码评审	139	第10章 面向对象的方法学	193
7.6.2 版本控制	140	10.1 面向对象的概念	193
7.7 实战训练	142	10.1.1 传统开发方法存在的问题	193
本章小结	144	10.1.2 面向对象的基本概念	194
习题 7	144	10.1.3 面向对象的特征	196
第8章 软件测试与调试	146	10.1.4 面向对象的要素	196
8.1 概述	146	10.1.5 面向对象的开发方法	197
8.1.1 软件测试的定义	147	10.2 面向对象的模型	197
8.1.2 软件测试的目的	147	10.2.1 对象模型	197
8.1.3 软件缺陷带来影响的例子	147	10.2.2 动态模型	201
8.2 软件测试的任务和目标	148	10.2.3 功能模型	203
8.2.1 软件测试的任务	148	10.3 面向对象的分析	203
8.2.2 软件测试的目标	149	10.3.1 面向对象的分析过程	203
8.3 软件测试的基本原则	150	10.3.2 建立对象模型	204
8.4 软件测试的方法	151	10.3.3 建立动态模型	208
8.4.1 测试用例	151	10.3.4 建立功能模型	211
8.4.2 黑盒测试法	153	10.3.5 确定操作	212
8.4.3 白盒测试法	156	10.4 面向对象的设计	212
8.5 软件测试的步骤	163	10.5 面向对象的实现	216
8.5.1 单元测试	163	10.6 面向对象和基于对象的区别	217
8.5.2 集成测试	165	10.7 实战训练	218
8.5.3 确认测试	168	本章小结	219
8.5.4 系统测试	169	习题 10	219
8.6 程序调试	172	第11章 统一建模语言(UML)	221
8.6.1 程序调试技术	172	11.1 概述	221
8.6.2 程序调试策略	173	11.1.1 什么是UML	221
8.7 实战训练	173	11.1.2 UML的发展史	221
本章小结	175	11.1.3 UML的特点	222
习题 8	176	11.1.4 UML的应用领域	222

11.1.5 基于 UML 的设计过程	223	12.5.6 制定项目文档管理	257
11.2 UML 概念模型	223	12.5.7 制定项目培训计划	257
11.2.1 UML 的构成	223	12.5.8 制定项目监控过程	258
11.2.2 UML 的基本构造块	223	12.5.9 制定项目进度表	259
11.2.3 UML 的规则	224	12.5.10 合成项目计划和从属计划	260
11.2.4 UML 的公共机制	225	12.5.11 获得对计划的承诺	260
11.3 UML 的静态建模机制	225	12.5.12 评审	261
11.3.1 用例图	225	12.5.13 跟踪项目计划估计值	262
11.3.2 类图	227	12.5.14 跟踪风险	264
11.3.3 对象图	228	12.5.15 里程碑总结	265
11.3.4 包图	229	12.5.16 变更管理	266
11.4 UML 的动态建模机制	230	12.5.17 问题管理	266
11.4.1 协作图	230	12.5.18 项目总结	267
11.4.2 时序图	231	12.6 模板和表格	268
11.4.3 活动图	231	12.7 实战训练	268
11.4.4 状态图	232	本章小结	268
11.5 UML 面向实现机制	233	习题 12	269
11.5.1 组件图	233	附录 软件开发过程中的文档	270
11.5.2 配置图	234	附录 A 可行性分析报告(GB8567—88)	270
11.6 UML 建模工具	234	A.1 引言	270
11.6.1 Rational Rose	234	A.2 可行性分析的前提	270
11.6.2 Microsoft Office Visio	235	A.3 对现有系统的分析	271
11.7 UML 建模实例	236	A.4 所建议系统	272
11.8 实战训练	239	A.5 可选择的其他系统方案	273
本章小结	243	A.6 投资及效益分析	273
习题 11	243	A.7 社会因素方面的可行性	275
第 12 章 软件项目管理	245	A.8 结论	275
12.1 项目与项目管理	245	附录 B 软件项目开发计划书	
12.2 ISO 9000 国际标准简介	247	(GB8567—88)	275
12.3 CMMI	248	B.1 引言	275
12.3.1 CMMI 的基本概念	248	B.2 项目概述	276
12.3.2 CMMI 的体系结构	248	B.3 实施计划	276
12.4 ISO 9000 与 CMMI 的比较	250	B.4 支持条件	277
12.5 软件项目管理过程	251	B.5 专题计划要点	277
12.5.1 项目组织结构	252	附录 C 软件规格说明书(GB8567—88)	277
12.5.2 项目启动	253	C.1 引言	277
12.5.3 项目过程定义	254	C.2 任务概述	278
12.5.4 工作分解结构	255	C.3 需求规定	278
12.5.5 制定风险计划	255	C.4 运行环境规定	279

附录 D 软件总体设计说明书 (GB8567—88).....	280	G.3 测试设计说明.....	288
D.1 引言.....	280	G.4 评价准则.....	288
D.2 总体设计.....	280	附录 H 测试分析报告(GB8567—88).....	289
D.3 接口设计.....	281	H.1 引言.....	289
D.4 运行设计.....	281	H.2 测试概要.....	289
D.5 系统数据结构设计.....	281	H.3 测试结果及发现.....	289
D.6 系统出错处理设计.....	282	H.4 对软件功能的结论.....	289
附录 E 软件详细设计说明书 (GB8567—88).....	282	H.5 分析摘要.....	290
E.1 引言.....	282	H.6 测试资源消耗.....	290
E.2 程序系统的结构.....	283	附录 I 开发进度月报(GB8567—88).....	290
E.3 程序 1(标识符)设计说明.....	283	I.1 标题.....	290
E.4 程序 2(标识符)设计说明.....	284	I.2 工程进度与状态.....	291
附录 F 操作手册(GB8567—88).....	284	I.3 资源耗用与状态.....	291
F.1 引言.....	284	I.4 经费支出与状态.....	291
F.2 软件特征描述.....	284	I.5 下个月的工作计划.....	292
F.3 安装与初始化.....	285	I.6 建议.....	292
F.4 运行说明.....	285	附录 J 软件项目开发总结报告 (GB8567—88).....	292
F.5 非常规过程.....	286	J.1 引言.....	292
F.6 远程操作.....	286	J.2 实际开发结果.....	293
附录 G 测试计划(GB8567—88).....	286	J.3 开发工作评价.....	293
G.1 引言.....	286	J.4 经验与教训.....	294
G.2 计划.....	287	参考文献.....	295

第 1 章 概 述

本章主要内容

- ◇ 软件的基本概念及特性。
- ◇ 软件的发展。
- ◇ 软件工程诞生的背景及软件工程三要素。
- ◇ 软件生命周期。

1.1 软 件

1.1.1 软件的特点及分类

软件的定义是随着计算机技术的发展而逐步完善的。在 20 世纪 50 年代,人们认为软件就等于程序;60 年代人们认识到软件的开发文档在软件中的作用,提出软件等于程序加文档,但这里的文档仅是指软件开发过程中所涉及的分析、设计、实现、测试、维护等方面的文档,不包括管理文档;到了 70 年代人们又给软件的定义中加入了数据。因此,软件是计算机系统中与硬件相互依存的一部分,它包括:

- 在运行中能提供所希望的功能与性能的程序;
- 使程序能够正确运行的数据及其结构;
- 描述软件研制过程和方法的文档。

1. 软件的特点

从广义来说,软件与硬件一样,也是产品,但两者之间是有差别的,了解并理解这种差别,对理解软件工程是非常重要的。软件的特点包括以下几点:

(1) 软件角色的双重性。软件作为一种产品具有双重性:一方面它是一个产品,利用它来表现计算机硬件的计算潜能,无论它是在主机中还是驻留在设备(如手机)中,软件就是一个信息转换器,可以产生、管理、获取、修改、显示或传送信息;另一方面它又是产品交付使用的载体,它可以控制计算机(如操作系统),可以实现计算机之间的通信,又可以创建其他程序与控制。

(2) 软件是被开发或设计出来的,而不是传统意义上所说的被制造出来的。一般意义上的产品,包括硬件产品,总要经过分析、设计、制造、测试等过程,也就是说要经过一个从无形的设想到一个有形的产品的过程。但软件仅仅是一个逻辑上的产品而不是有形的系

统元件，软件是通过人的智力劳动设计开发出来的，而不是制造出来的。而且软件一旦被开发出来，就可以进行大量的复制，因此其研制成本要远远大于生产成本。这也意味着软件的开发不能像制造其他产品那样进行管理。

(3) 软件不会“磨损”，但会退化。一般情况下，有形的硬件产品在使用过程中总会有磨损。在使用初期，磨损往往比较严重(这实际上是磨合)，而经过了一段不长时间的磨合后，将进入相对的稳定期。随着时间的流逝，硬件的磨损再次趋于严重，这也意味着硬件的寿命快要到了。硬件的磨损与时间之间的关系可以用图 1.1 所示的“浴缸曲线”来表示。

软件并不是一种有形的产品，因此也就不存在所谓的“磨损”问题。理想情况下，软件的故障曲线应该是图 1.2 中所示的理想曲线。在软件的运行初期，未知的错误使得程序有较高的故障率，当修正了这些错误而且未引入新的错误时，软件将进入一种比较理想的平稳运行期。这说明软件是不会“磨损”的。但在实际情况中，软件尽管不会“磨损”，但会退化，如图 1.2 中的实际曲线那样。这是因为软件在其生命周期中会经历多次修改，每次修改都会引入新的错误，而对这些错误又要进行新的修改，使得软件的故障曲线呈现一种锯齿形，导致最后的故障率慢慢升高，即软件产生了退化，而这种退化缘于修改。

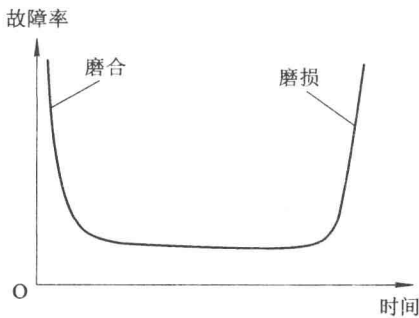


图 1.1 硬件故障率曲线

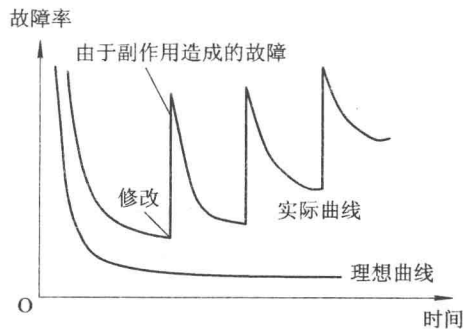


图 1.2 软件故障率曲线

(4) 绝大多数软件都是定制的且是手工编写的。在硬件制造业，构件的复用是非常自然的。理想情况下，软件构件应该能够被复用于不同的程序。但由于软件本身的特殊性，软件构件复用才刚刚起步。尽管今天的面向对象技术、构件技术已经使软件的复用逐渐成为流行，但这种复用还不能做到像硬件产品那样拿来即用，还需要进行必要的定制(构件之间的组合、接口的设计、功能的修改与扩充等)，而且软件开发中构件的使用比例也是有限的。整个软件产品的设计基本上还依赖于人们的智力与手工劳动。

(5) 开发过程的复杂与昂贵的费用。现代软件的体系结构越来越复杂，规模越来越庞大，所涉及的学科也越来越多，导致了软件的开发过程也异常复杂。靠一个人单枪匹马开发一套软件的时代已经一去不复返了，而是需要一个分工明确、层次合理、组织严密的团队才能完成。这样，软件的开发成本也会越来越昂贵。

2. 软件的分类

软件的应用非常广泛，几乎渗透到各行各业。因此要给出一个科学的、统一的、严格的计算机软件分类标准是不现实也是不可能的，但可以从不同的角度对软件进行适当的分类。常用的软件分类方法及典型应用如表 1.1 所示。

表 1.1 软件 的 分类

分类序号	分类方法	对应类别	典型应用与特征
1	按功能分类	(1) 系统软件	与计算机硬件接口并为其其他程序服务的软件, 如操作系统、驱动程序等
		(2) 支撑软件	用于开发软件的工具性软件, 如开发平台、数据库管理系统、各种应用工具软件等
		(3) 应用软件	为解决某一领域的应用而开发的软件, 如商业软件、嵌入式软件、个人计算机软件、Web 软件、人工智能等
2	按版权分类	(1) 商业软件	版权受法律保护、经授权方可使用且必须购买的软件
		(2) 共享软件	与商业软件类似, 但可以“先尝后买”, 其获取途径主要是通过因特网
		(3) 自由(免费)软件	无需支付许可证费用便可得到和使用的软件, 发行渠道类似于共享软件
		(4) 公有领域软件	没有版权, 任何人均可以使用而且可以获得源代码的软件
3	按工作方式分类	(1) 实时软件	用于及时处理实时发生的事件的软件, 如控制、订票系统等
		(2) 分时软件	多个联机用户同时使用计算机的软件
		(3) 交互式软件	能够实现人机通信的软件
		(4) 批处理软件	将多个作业或多批数据一次运行、顺序处理的软件
4	按销售方式分类	(1) 定制软件	受某个特定客户委托, 在合同的约束下开发的软件
		(2) 产品软件	由软件开发机构开发, 可以为众多用户服务并直接提供给市场的软件

1.1.2 软件的发展

自 20 世纪 40 年代出现了世界上第一台计算机以后, 就有了程序的概念, 可以认为它是软件的前身。经过了几十年的发展, 人们对软件有了更深刻的认识。在这几十年中, 计算机软件经历了程序设计、程序系统和软件工程三个发展时期。

表 1.2 列出了三个发展时期主要特征的对比, 由此可以看出几十年来软件最根本的变化。

表 1.2 计算机软件发展的三个时期及其特点

特 点	程序设计 (20 世纪 50~60 年代)	程序系统 (20 世纪 60~70 年代)	软件工程 (20 世纪 70 年代以后)
软件所指对象	程序	程序及说明书	程序、文档、数据
主要程序设计语言	机器及汇编语言	高级语言	软件语言(包括需求定义语言、软件功能语言、软件设计语言、程序设计语言等)
软件工作范围	程序编写	包括设计和测试	软件生存期
需求者	程序设计者本人	少数用户	市场用户
开发软件的组织	个人	开发小组	开发小组及大中型软件开发机构

续表

特 点	程序设计 (20 世纪 50~60 年代)	程序系统 (20 世纪 60~70 年代)	软件工程 (20 世纪 70 年代以后)
软件规模	小型	中小型	大中小型
决定质量的因素	个人编程技术	小组技术水平	管理水平
开发技术和手段	子程序 程序库	结构化程序设计	数据库、开发工具、开发环境、工程化开发方法、标准和规范、网络及分布式开发、面向对象技术
维护责任者	程序设计者	开发小组	专职维护人员
硬件特征	价格高 存储容量小 工作可靠性差	价格降低, 速度、容量及工作可靠性有明显提高	向超高速、大容量、微型化及网络化方向发展
软件特征	完全不受重视	软件技术的发展不能满足需要, 出现软件危机	开发技术有进步, 但未获突破性进展, 价高, 未完全摆脱软件危机

1.1.3 软件危机及其表现

所谓软件危机, 即指在软件开发与维护过程中遇到的一系列严重问题。

软件危机主要表现在以下方面:

- (1) 软件开发进度难以预测;
- (2) 软件开发成本难以控制;
- (3) 用户对软件产品的功能要求难以满足;
- (4) 软件产品的质量无法保证, 系统中的错误难以消除;
- (5) 软件产品难以维护;
- (6) 软件缺少适当的文档资料;
- (7) 软件开发的速度难以满足社会需求的增长。

1.1.4 软件危机产生的主要原因

随着社会对计算机应用需求的增长, 软件系统的规模越来越庞大, 开发难度和成本越来越高, 软件需求量剧增, 软件质量没有可靠的保证, 软件开发的生产率低。这些因素构成了软件生产的恶性循环。软件生产的复杂性和高成本, 使大型软件的生产出现了很大的困难, 由此出现了“软件危机”, 其具体表现如下:

(1) 开发人员和用户之间的矛盾。用户在开发初期, 由于各种原因往往不能准确地提出需求描述; 开发人员在还没有准确、完整地了解用户的实际需求后就急于编程。

(2) 大型软件项目需要组织一定的人力共同完成, 多数管理人员缺乏开发大型软件系统的经验; 多数软件开发人员缺乏协同方面的经验; 软件项目开发人员不能有效地、独立自主地处理大型软件的全部关系和各个分支, 因此容易产生疏漏和错误。

(3) 缺乏有力的方法学和工具方面的支持, 过分依靠程序设计人员的技巧和创造性。重编程, 轻需求分析; 重开发, 轻维护; 重程序, 轻文档。这样做的后果就是在软件系统中“埋藏”了许多故障隐患, 直接危害着系统的可靠性和稳定性。

1.1.5 解决软件危机的途径

分析了造成软件危机的原因后, 人们开始探索用工程的方法进行软件生产的可能性, 即用软件工程的概念、原理、技术和方法进行软件的开发、管理、维护和更新。于是, 计算机科学的一个新领域——“软件工程”诞生了。

1.2 软件工程

1.2.1 软件工程的定义

通俗地说, 软件工程即借用传统工程设计的基本思想, 采用工程化的概念、原理、技术和方法来开发与维护软件, 突出软件生产的科学方法, 把经过时间考验而证明正确的管理技术与当前能够得到的最好的技术和方法结合起来, 降低开发成本, 缩短研制周期, 提高软件的可靠性和生产效率。软件工程是指导计算机软件开发和维护的工程学科。

经过 40 多年的发展, 软件的工程化生产已成为软件产业。软件已成为产品, 它涉及价值、市场、版权和法律保护等方面的问题。

软件工程是一门交叉学科, 需要用管理学的原理和方法来进行软件生产管理, 用工程学的观点来进行费用估算, 制定进度和实施方案, 用数学方法来建立软件可靠性模型并分析各种算法。

1.2.2 软件工程的三要素

软件工程以关注软件质量为目标, 由方法、工具和过程三个要素构成, 如图 1.3 所示。

软件工程方法为软件开发提供了“如何做”的技术, 涉及软件工程的多个方面, 如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法过程的设计、编码、测试以及维护等。

软件工程工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前, 已经推出了许多软件工程工具, 这些软件工程工具集成起来, 建立起了称之为计算机辅助软件工程 (Computer Aided Software Engineering, CASE) 的软件开发支撑系统。CASE 将各种软件工程工具、开发机器和一个存放开发过程信息的工程数据库组合起来, 形成一个软件工程环境。

软件工程过程将软件工程的方法和工具综合起来, 以达到合理、及时地进行计算机软

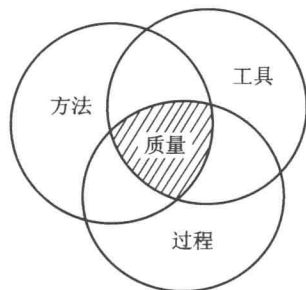


图 1.3 软件工程三要素

件开发的目的是。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理及软件开发各个阶段完成的里程碑。

1.3 软件生命周期

软件产品从形成概念开始，经过开发、使用和维护，直到最后退役的全过程称为软件的生命周期。软件的生命周期可以根据软件所处的状态、特征以及软件开发活动的目的、任务划分为若干个时期，而每一个时期又进一步划分为若干个阶段。

我国国家标准《计算机软件开发规范》(GB8566—88)把软件生命周期划分为可行性研究与计划、需求分析、概要设计、详细设计、实现、组装测试、确认测试、使用与维护八个阶段。通常，人们把可行性研究与计划、需求分析两个阶段称为软件定义时期，把概要设计、详细设计、实现、组装测试和确认测试等五个阶段称为软件开发时期，而把使用与维护阶段称为软件运行与维护时期。软件生命周期各个时期及阶段的关系如图 1.4 所示。

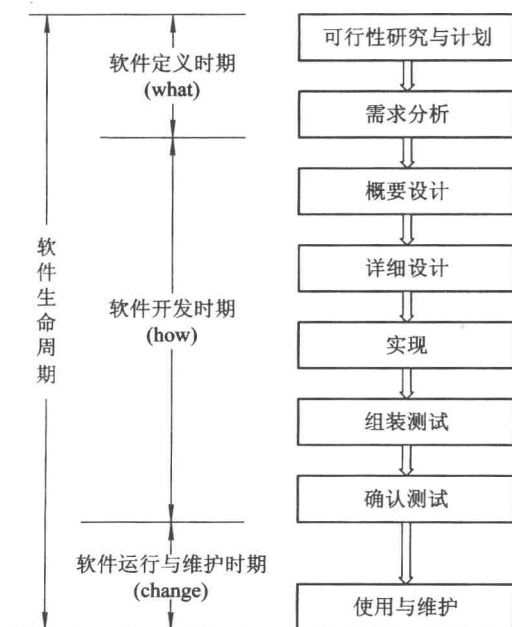


图 1.4 软件生命周期各个时期及阶段的关系

软件生命周期各阶段的主要工作步骤、任务和阶段性成果如下。

1. 软件定义时期

在软件生命周期中，软件定义时期又可分为可行性研究与计划和需求分析两个阶段。

(1) 可行性研究与计划阶段。

① 问题定义：通过调研，提出要解决的问题、工程目标和规模，形成用户的初步需求报告并得到用户的确认。

② 可行性论证：根据用户确认的初步用户需求报告和现实环境条件，从技术、经济和社会等方面研究并论证软件系统的可行性，对方案进行选择并形成可行性分析报告。