

高等学校教材·软件工程

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

Software Engineering and Management

软件工程与管理

覃征

Qinzheng

编 著

何 坚 高洪江 李顺东 黄 茹

Hejian Gaohongjiang Lishundong Huangru



清华大学出版社

高等学校教材·软件工程

Software Engineering and Management

软件工程与管理

覃 征

Qinzheng

何 坚 高洪江 李顺东 黄 茹 编著
Hejian Gaohongjiang Lishundong Huangru

清 华 大 学 出 版 社

北 京

内 容 简 介

本书以软件工程的3个层次——传统软件工程、软件工程新方法以及软件工程标准与管理为主线,比较全面地介绍了软件工程的基本概念、基本理论、基本方法及其应用技术。全书共13章,分为3个部分。第1部分包括第1章至第7章,介绍了软件工程的基础知识,即软件过程模型及软件开发方法,重点讨论了软件需求、设计、编码、测试和维护的方法及处理技术;第2部分包括第8章至第10章,分别介绍了软件体系结构技术、面向对象技术、软件复用、净室软件工程和敏捷开发方法;第3部分详细介绍了软件工程标准、规范及软件工程管理的方法和技术。

本书可作为高等院校高年级本科生、研究生学习软件工程的教材和教学参考书,也可作为从事软件工程研究和应用的有关人员的参考书。

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

软件工程与管理/覃征等编著. —北京:清华大学出版社,2005.6

(高等学校教材·软件工程)

ISBN 7-302-10935-4

I. 软… II. 覃… III. 软件工程—管理 IV. TP311.5

中国版本图书馆CIP数据核字(2005)第042479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

组稿编辑: 丁 岭

文稿编辑: 许振伍

封面设计: 王 永

印 刷 者: 北京密云胶印厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 29.25 字数: 726千字

版 次: 2005年6月第1版 2005年6月第1次印刷

书 号: ISBN 7-302-10935-4/TP·7255

印 数: 1~4000

定 价: 38.00元

出版说明

改革开放以来，特别是党的十五大以来，我国教育事业取得了举世瞩目的辉煌成就，高等教育实现了历史性的跨越，已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上，高等教育规模取得如此快速的发展，创造了世界教育发展史上的奇迹。当前，教育工作既面临着千载难逢的良好机遇，同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾，是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月，教育部下发了《关于加强高等学校本科教学工作，提高教学质量的若干意见》，提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月，教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件，指出“高等学校教学质量和教学改革工程”是教育部正在制订的《2003—2007年教育振兴行动计划》的重要组成部分，精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间（2003—2007年）建设1500门国家级精品课程，利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放，以实现优质教学资源共享，提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括：

(1) 高等学校教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

清华大学出版社经过近二十年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材经过二十多年的精雕细刻，形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
E-mail: dingl@tup.tsinghua.edu.cn

前 言

几十年来软件开发过程中的经验教训证明：软件开发是一个系统工程，它需要良好的组织、严格的管理、以及各类人员的协作才得以完成。因此，按工程化的原则和方法组织管理软件开发工作是有效摆脱软件危机的一个主要出路。

软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门学科。20 世纪 80 年代以来，软件工程新技术的研究呈不断上升趋势，国内外研究者在软件工程领域取得了丰硕的研究成果，各种新方法、新技术不断展露。为使得许多初学者能了解软件工程领域的全貌。推动软件工程研究和应用的进一步深入和普及，我们不揣浅陋，集合国内外学者的有关研究成果编成此书。

软件工程的研究历史虽然相对较短，然而近年来成果却十分丰富。因此，我们在材料的取舍上以及本书的章节安排上颇费思量。本书从软件工程领域最基本、最实用的方法、技术入手，对面向对象技术、软件体系结构技术、敏捷软件开发等新技术进行深入介绍；同时系统介绍了软件工程标准、规范，以及软件工程管理的方法与技术。考虑到本书的读者对象和国内教学的实际情况，我们力图在书中较完整地介绍若干典型的方法、技术和教学案例，而不追求面面俱到。因此，本书难免有疏漏之憾。

书中大量引用了国内外众多学者的研究成果与相关资料，没有他们的创造性工作，就不可能有这本书的问世，我们对他们表示崇高的敬意和感谢。本书由覃征教授确定研究内容的选取和组织，覃征、何坚负责统稿定稿。各部分执笔是：第一部分（1~7 章）高洪江、黄茹，第二部分（8~10 章）何坚，第三部分（11~13 章）李顺东。同时，我们也要感谢清华大学出版社的各位领导和编辑，他们为使本书早日与读者见面耗费了大量的心血；感谢清华大学信息学院软件学院计算机科学与技术系许多教师的指教。并提出了很多宝贵的修改意见。

鉴于我们的水平有限，加之时间仓促，书中的错误和不足在所难免，敬请读者批评、指正。

作 者
2004 年

目 录

第 1 部分 软件工程基础

第 1 章 软件工程概述	2
1.1 软件的概念与特点	2
1.2 软件的分类	3
1.3 软件的发展和软件危机	5
1.4 软件过程及其模型	8
1.4.1 软件过程	8
1.4.2 软件过程模型	12
1.4.3 软件工程的目标和原则	25
1.5 软件开发工具	30
1.5.1 软件开发的分析工具	30
1.5.2 CASE 工具	31
1.6 未来之路	34
1.7 小结	38
参考文献	39
第 2 章 软件项目分析	41
2.1 概述	41
2.2 软件项目规模估算	44
2.2.1 代码行技术	44
2.2.2 任务估算技术	45
2.2.3 Delphi 法	46
2.2.4 类比法	47
2.2.5 功能点估计法	48
2.2.6 PERT 估计法	50
2.3 成本估计与效益分析	52
2.3.1 成本估计	52
2.3.2 效益分析	58
2.4 现有系统分析	59
2.5 可行性研究报告	66
2.5.1 可行性研究	66
2.5.2 可行性研究报告	69
2.6 立项评审	74

2.6.1 立项检查.....	74
2.6.2 立项评审报告.....	74
2.7 小结.....	75
参考文献.....	75
第3章 需求分析	77
3.1 概述.....	77
3.1.1 需求的类型.....	78
3.1.2 需求分析目标.....	80
3.1.3 需求分析过程.....	81
3.2 需求获取.....	85
3.2.1 需求获取方法.....	86
3.2.2 分析人员与用户的合作关系.....	87
3.3 需求分析.....	90
3.3.1 软件需求分析.....	90
3.3.2 需求和系统模型之间的关系.....	94
3.4 结构化分析方法.....	95
3.4.1 结构化分析.....	95
3.4.2 使用 PDL 描述需求	106
3.4.3 接口描述.....	107
3.5 软件原型实现.....	107
3.5.1 原型的特征.....	108
3.5.2 原型的选择.....	108
3.5.3 原型实现方法和工具.....	109
3.6 需求描述与评审.....	110
3.6.1 分析建模.....	110
3.6.2 软件需求规约.....	111
3.7 需求验证与评审.....	114
3.7.1 需求有效性验证.....	114
3.7.2 需求评审.....	116
3.8 需求管理.....	117
3.8.1 需求跟踪表.....	118
3.8.2 需求变更管理.....	120
3.9 小结.....	120
参考文献.....	121
第4章 软件设计	123
4.1 软件设计过程.....	123
4.1.1 软件设计的重要性.....	123

4.1.2	总体设计.....	124
4.1.3	详细设计.....	124
4.2	结构化软件设计方法.....	124
4.2.1	功能分解.....	125
4.2.2	抽象.....	125
4.2.3	信息隐藏.....	126
4.2.4	模块化.....	126
4.2.5	软件设计原则.....	127
4.2.6	数据设计.....	128
4.2.7	软件结构设计.....	128
4.2.8	过程设计.....	129
4.3	模块化设计.....	129
4.3.1	功能独立性.....	129
4.3.2	内聚性.....	129
4.3.3	耦合度.....	131
4.4	软件设计工具.....	133
4.4.1	总体设计工具.....	133
4.4.2	详细设计工具.....	135
4.5	Jackson 设计方法.....	140
4.5.1	Jackson 结构图.....	140
4.5.2	Jackson 设计方法.....	141
4.5.3	结构冲突.....	141
4.6	设计规格说明与评审.....	142
4.6.1	设计规格说明.....	142
4.6.2	软件设计评审.....	143
4.7	小结.....	144
	参考文献.....	144
第 5 章	程序编码.....	145
5.1	程序设计语言.....	145
5.1.1	程序设计语言特性.....	145
5.1.2	程序设计语言的分类.....	147
5.1.3	程序设计语言的选择.....	148
5.2	程序设计过程.....	149
5.2.1	结构化程序设计的概念与标准结构.....	150
5.2.2	结构化编程.....	152
5.3	程序设计风格.....	154
5.3.1	选用合适的常量标识符.....	155
5.3.2	选择有实际含义的标识符作为变量名.....	155

5.3.3	坚持按一定的缩进规则书写和录入程序.....	155
5.3.4	适当使用注释.....	155
5.4	程序效率.....	157
5.4.1	讨论效率的准则.....	157
5.4.2	算法对效率的影响.....	157
5.4.3	影响存储效率的因素.....	158
5.4.4	影响输入/输出的因素.....	158
5.5	程序编码优化.....	158
5.5.1	程序优化的内容与原则.....	158
5.5.2	程序结构优化的基本方法.....	160
5.5.3	程序代码优化的基本方法.....	161
5.6	小结.....	165
	参考文献.....	165
第 6 章	软件测试.....	167
6.1	测试基础.....	167
6.1.1	软件测试.....	168
6.1.2	验证和确认.....	172
6.1.3	测试流程.....	173
6.1.4	测试的组织.....	176
6.2	测试方法.....	176
6.2.1	测试设计中的系统分析方法.....	176
6.2.2	测试技术的分类.....	177
6.2.3	逻辑覆盖.....	183
6.2.4	基本路径测试.....	184
6.2.5	控制结构测试.....	188
6.3	测试过程.....	190
6.3.1	单元测试.....	191
6.3.2	集成测试.....	193
6.3.3	确认测试.....	196
6.3.4	系统测试.....	197
6.3.5	停止测试的时机.....	198
6.4	调试方法.....	198
6.4.1	调试步骤.....	199
6.4.2	几种主要调试方法.....	199
6.4.3	调试原则.....	200
6.5	自动测试工具.....	201
6.5.1	常用自动测试工具.....	201
6.5.2	工具对测试工作的影响.....	202

6.5.3	测试工具的开发.....	203
6.5.4	软件质量评测及其工具.....	205
6.6	软件可靠性.....	205
6.6.1	基本概念.....	206
6.6.2	软件可靠性与硬件可靠性.....	209
6.6.3	软件可靠性参数和模型.....	211
6.6.4	软件可靠性的衡量方法.....	212
6.6.5	软件的可靠性测试过程.....	215
6.6.6	提高软件可靠性的方法和技术.....	216
6.7	小结.....	219
	参考文献.....	220
第7章	软件维护	223
7.1	软件维护的概念.....	223
7.1.1	软件维护的定义.....	223
7.1.2	软件维护的工作量.....	224
7.1.3	软件维护的代价.....	224
7.1.4	影响维护的因素.....	225
7.2	软件维护活动.....	226
7.2.1	维护机构.....	226
7.2.2	编写报告.....	227
7.2.3	软件维护工作流程.....	227
7.2.4	维护评价.....	228
7.3	程序修改的副作用.....	229
7.3.1	修改代码的副作用.....	229
7.3.2	修改数据的副作用.....	229
7.3.3	文档的副作用.....	230
7.4	软件可维护性.....	230
7.4.1	软件可维护性的定义.....	230
7.4.2	可维护性的度量.....	231
7.4.3	评审	231
7.5	提高可维护性的方法.....	232
7.5.1	明确软件质量管理目标.....	232
7.5.2	使用提高软件质量的技术和工具.....	232
7.5.3	选择可维护的程序设计语言.....	233
7.5.4	改进程序的文档.....	233
7.5.5	进行明确的质量保证审查.....	234
7.6	软件再工程.....	234
7.6.1	再工程的基本概念.....	234

7.6.2 再工程的相关软件技术.....	236
7.6.3 再工程应用实例.....	237
7.7 小结.....	237
参考文献.....	238

第2部分 新技术、新方法

第8章 软件体系结构	243
8.1 体系结构的概念及描述.....	243
8.1.1 体系结构的概念.....	243
8.1.2 体系结构的描述.....	244
8.1.3 体系结构的生命周期模型.....	246
8.2 体系结构风格.....	248
8.2.1 管道过滤模式.....	248
8.2.2 面向对象模式.....	250
8.2.3 事件驱动模式.....	251
8.2.4 分层模式.....	256
8.2.5 知识库模式.....	258
8.2.6 解释器模式.....	260
8.2.7 过程控制环模式.....	260
8.3 基于体系结构的软件开发.....	266
8.3.1 体系结构驱动的软件分析设计.....	266
8.3.2 体系结构评估.....	270
8.4 多层软件体系结构实例.....	273
8.4.1 J2EE 概述	273
8.4.2 JBoss 应用服务器	275
8.5 小结.....	279
参考文献.....	279
第9章 面向对象方法	280
9.1 OO 基本原理.....	280
9.1.1 OO 基本概念.....	280
9.1.2 OO 的特征.....	282
9.1.3 OO 的要素.....	283
9.2 OO 开发方法.....	284
9.2.1 Booch 方法.....	284
9.2.2 Coad 方法.....	288
9.2.3 OMT 方法.....	292
9.3 UML 技术.....	304

9.3.1 UML 概述.....	304
9.3.2 UML 表示法.....	305
9.3.3 UML 建模实例.....	312
9.4 小结.....	317
参考文献.....	317
第 10 章 软件开发新方法	318
10.1 软件复用.....	318
10.1.1 软件复用概念.....	318
10.1.2 领域工程.....	321
10.1.3 基于构件的软件开发.....	323
10.1.4 软件生产线.....	326
10.2 净室软件工程.....	327
10.2.1 净室基础.....	328
10.2.2 净室小组的工作.....	329
10.2.3 净室技术.....	330
10.2.4 净室过程.....	333
10.2.5 净室与其他软件工程实践的关系.....	334
10.3 敏捷软件工程.....	336
10.3.1 敏捷方法学.....	337
10.3.2 敏捷建模.....	338
10.3.3 敏捷方法.....	341
10.4 小结.....	344
参考文献.....	345

第 3 部分 软件工程管理标准、规范

第 11 章 软件工程标准.....	348
11.1 软件工程标准概述.....	348
11.1.1 软件工程标准化的意义.....	348
11.1.2 软件工程标准的制订与推行.....	351
11.1.3 软件工程标准的层次.....	352
11.1.4 中国的软件标准化工作.....	353
11.2 软件能力成熟度模型简介.....	354
11.2.1 CMM 概述.....	354
11.2.2 CMM 级别.....	356
11.2.3 CMM 与管理.....	359
11.2.4 软件过程评估.....	360
11.2.5 软件过程评估方法.....	360

11.2.6 从 CMM 到 CMMI	361
11.3 软件工程国际标准	365
11.3.1 ISO9000 的管理思想	365
11.3.2 ISO9000 族标准构成	366
11.3.3 ISO9000-3 实施	369
11.3.4 ISO/IEC15504 简介	370
11.3.5 其他软件工程标准	373
11.4 小结	373
参考文献	374
第 12 章 软件工程规范	375
12.1 PSP 方法	375
12.1.1 PSP 概述	375
12.1.2 PSP 的演化	376
12.1.3 PSP 过程实施	377
12.2 TSP 方法	379
12.2.1 TSP 概述	379
12.2.2 TSP 的结构	381
12.2.3 实现 TSP 的条件	382
12.2.4 TSP 实施方法	382
12.3 RUP 方法	384
12.3.1 RUP 概述	384
12.3.2 RUP 应用	385
12.3.3 RUP 的核心工作流	386
12.3.4 RUP 管理实施	388
12.4 小结	388
参考文献	389
第 13 章 软件工程管理	390
13.1 软件管理的目标与基本原则	390
13.1.1 管理目标	390
13.1.2 基本原则	390
13.1.3 专题原则	394
13.2 风险管理	396
13.2.1 风险的分类	396
13.2.2 风险的识别	397
13.2.3 风险评估	397
13.2.4 风险的驾驭和监控	398
13.3 组织管理	399

13.3.1	民主制程序员组	399
13.3.2	主程序员组	401
13.3.3	现代程序员组	402
13.3.4	软件项目组	404
13.3.5	IT 组织管理	406
13.4	质量管理	408
13.4.1	软件质量概述	409
13.4.2	软件质量因素	411
13.4.3	质量认证	411
13.4.4	测试管理	423
13.5	配置管理	427
13.5.1	配置管理的意义	427
13.5.2	配置管理的实施过程	428
13.5.3	配置控制	431
13.5.4	配置管理计划	435
13.5.5	配置管理报表	439
13.6	软件项目管理	442
13.6.1	软件项目产品的特点	442
13.6.2	软件项目失控的原因	443
13.6.3	软件项目管理的内容	447
13.7	小结	448
	参考文献	448
附录 A	主要英文缩写	450

第 1 部分

软件工程基础

随着计算机硬件的迅速更新换代、成本下降及性能提高，计算机的应用更加普及，当代科学和技术各个领域、工业和社会各个部门对计算机系统的依赖性也日益增强。

计算机在迅速解放社会生产力、社会高度自动化和信息化的同时，计算机软件生产却依然非常落后。经典的软件开发方法起源于计算机硬件性能比较低下、计算机应用尚不广泛的年代，因而无法摆脱人工方式的局限性，开发出的软件无法重用，重复性的劳动无法避免，既大量浪费了宝贵的人力、物力、财力资源，也严重阻滞了软件产业的发展，制约了软件生产率的提高。这促使软件专家在改进和完善基于经典开发范式涉及的理论、技术和方法的同时，着重研究新思想、新方法、新理论、新技术，以求从根本上改变软件生产面临的窘境。

自 20 世纪中期尤其是“软件危机”之后，随着研究的不断深入和广泛，作为计算机学科中一个既年轻又充满活力的研究领域，软件工程学已成为信息社会高技术竞争的关键领域之一。软件工程学强调软件产品的生产特性，对软件设计方法论及工程化开发技术重点地展开了研究，并将计算机科学理论和工程方法相结合，系统地研究软件生命周期的一切活动，最终为经济有效地开发软件系统提供科学的方法和工具。

软件产业是一个朝阳的、新兴的知识型产业。一个国家软件业的发展程度，也体现了国家的综合国力，决定着国家未来的国际竞争地位。

本篇系统地介绍经典的面向过程软件工程的基本概念、原理、技术、方法和工具。这部分内容是软件工程学的基础内容，至今依然适用于许多中小型项目的开发。

第 1 章 软件工程概述

近年来随着微电子学技术的飞速发展，计算机硬件性能/价格比平均每 10 年至少提高两个数量级，而且其质量也不断提高。与此同时，计算机软件系统的成本却在逐年上升，规模越来越庞大、结构越来越复杂，由于软件错误而导致灾难性后果的报道屡见不鲜，软件开发的生产率也越来越满足不了计算机应用日益普及的需求。事实上软件早已成为制约计算机发展的关键因素。

在计算机系统发展早期，软件开发基本上沿用“软件作坊”式的个体化方法，这种方法在软件开发和维护过程中遇到了一系列严重问题：程序质量低下，错误频出，进度延误，费用剧增……。这些问题导致了“软件危机”。1968 年，北大西洋公约组织的计算机科学家在联邦德国召开国际会议讨论软件危机问题，正式提出并使用了“软件工程”这个名词，从此诞生了一门新兴的工程学科。

人们针对“软件危机”现象提出的多种解决方法归纳起来有两类：一是采用工程方法来组织、管理软件的开发过程；二是深入探讨程序和程序开发过程的规律，建立严密的理论，以期能够用来指导软件开发实践。前者导致“软件工程”的出现和发展，而后者则推动了形式化方法的深入研究。

1.1 软件的概念与特点

1983 年 IEEE 将软件定义为：计算机程序、方法、规则和相关文档资料以及在计算机上运行时所必需的数据。目前对软件比较公认的解释是：程序、支持程序运行的数据以及与程序有关的文档资料的完整集合。其中，程序是按事先设计的功能和性能要求执行的指令序列，数据是使程序能正常操作信息的数据结构，文档是与程序开发、维护和使用有关的图文材料。

软件具有以下一些特点：

- 软件是一种逻辑实体，它具有抽象性。
- 由于软件是被开发或设计的（不是传统意义上被制造的），所以软件成本集中在开发上，对软件的质量控制必须从软件的开发着手。
- 与机械、电子设备不同，软件在运行和使用过程中没有磨损、老化问题。前者在运行和使用中的故障率大都遵循如图 1-1 (a) 所示的 U 形曲线（即浴缸曲线）；而软件不存在磨损和老化问题，但是它会退化，要多次经过修改或维护，其故障率表现为如图 1-1 (b) 所示的锯齿形。因此，软件维护要比硬件维护复杂得多。
- 软件一旦研制成功，其生产过程就变成复制过程，不像其他工程产品那样有明显的生产制造的特征，但是会出现软件产品版权保护问题和打击盗版问题。
- 软件的开发和运行常受到计算机硬件和环境的限制，这导致软件升级和移植的问