

高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

软件工程

◎ 张海藩 编著

人民邮电出版社
www.pptph.com.cn

高等学校 21 世纪教材

软 件 工 程

张海藩 编著

人民邮电出版社

图书在版编目(CIP)数据

软件工程/张海藩编著. —北京:人民邮电出版社, 2002.3

高等学校 21 世纪教材

ISBN 7-115-09378-4

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

中国版本图书馆 CIP 数据核字(2001)第 089560 号

内 容 提 要

本书吸取了国内外大量同类书刊的精华,并总结了编者多年来从事软件工程教学和研究的经验和体会,其特点是:讲解深入浅出,着重讲透基本的概念、原理、技术和方法;既注重系统性、科学性和先进性,又特别注重实用性;既有原理性论述,又有丰富、完整的实例与之配合,利于读者理解和掌握。

本书由五篇共 16 章构成,第一篇讲述软件工程与软件过程;第二篇讲述结构化分析、设计与实现;第三篇讲述面向对象的概念、模型、分析、设计与实现;第四篇讲述软件项目的计划、组织和控制,并介绍了一些相关的国际标准;第五篇讲述形式化方法、统一建模语言 UML 和软件重用。

本书内容新颖、实例丰富,可作为高等院校“软件工程”课程的教材或教学参考书,也可供软件工程师、软件项目管理者 and 应用软件的开发人员阅读参考。

高等学校 21 世纪教材

软 件 工 程

◆ 编 著 张海藩
责任编辑 潘春燕

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@pptph.com.cn

网址 <http://www.pptph.com.cn>

读者热线 010-67180876

北京汉魂图文设计有限公司制作

北京朝阳隆昌印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 22.25

字数: 529 千字

2002 年 3 月第 1 版

印数: 1-6 000 册

2002 年 3 月北京第 1 次印刷

ISBN 7-115-09378-4/TP · 2269

定价: 29.00 元

本书如有印装质量问题,请与本社联系 电话:(010)67129223

丛书前言

当今世界,科学技术突飞猛进,知识经济已见端倪,国际竞争日趋激烈。教育在综合国力的形成中处于基础地位,国力的强弱将越来越取决于劳动者的素质,取决于各类人才的质量和数量,这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初,我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体,是进行教学的基本工具,也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此,高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代,计算机教学内容也随之更新,尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施,对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授,精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略,组织编写好这套教材,我们在国内高校做了系统、详细的调查,对教育部制订的教育计划做了认真的研究,还对国内外已出版的教材做了理性的分析,确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中,我们多次组织了由专家及高校一线教师参加的研讨会,对现有比较出色的教材的特点及优点进行了分析,博采众长,力求实现教材权威性与实用性的完美结合。

本套教材有如下特点:

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况,在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展,在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练,理论适中,实例丰富,写作风格上深入浅出,图文并茂,便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点,教材不仅篇幅有很大的压缩,而且均配有电子教案,以满足现代教学新特点的需要,做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授,所编教材具有较高的权威性。

教育的改革将不会停止,教材也将会不断推陈出新。目前本套教材即将推出,将接受广大教学第一线教师的检验。

由于我们的水平和经验有限,这批教材在编审、出版工作中还存在不少缺点和不足,希望使用本套教材的学校师生和广大读者提出批评和建议,以便改进我们的工作,使教材质量不断提高。

编者的话

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,把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来,以便经济地开发出高质量的软件并有效地维护它。经过 30 多年的研究与发展,软件工程正逐步走向成熟,但是,很多个人和公司仍然在随意地开发软件,许多软件专业的学生甚至软件工程师还没有掌握现代化的软件开发方法,以致我们生产的软件仍然存在大量的质量问题。因此认真学习并在实际工作中正确地运用软件工程,是摆在我们面前的一项十分迫切的任务。

编者从事软件工程教学工作近 20 年,亲身体会到教材对教师和学生的重要性。此次应邀编写《软件工程》一书,深感责任重大,因此,编写此书前从各方面都做了充分的准备。

本书由五篇共 16 章构成,按专题安排,便于授课教师和读者按需取用。

第一篇软件工程与软件过程,全面、概括地介绍软件工程这门学科及典型的软件过程模型。

第二篇传统方法学,分别讲述了结构化分析、结构化设计和结构化实现的概念、原理、准则、技术和方法。

第三篇面向对象方法学,分别讲述了面向对象的概念与模型、面向对象分析、面向对象设计和面向对象实现。本篇面向对象方法学是本书重点讲述的内容,因此本篇在讲述面向对象分析和面向对象设计的过程中不仅始终结合了 ATM 系统的例子,而且还单独讲述了电梯系统的面向对象分析过程及 C++ 类库管理系统的面向对象分析与设计过程。这些完整的实例对读者深入、具体地理解面向对象方法学很有帮助,也是上机实习的好材料。

第四篇软件项目管理,分别讲述软件项目的计划、组织和控制,并简要地介绍了一些与软件项目管理有关的国际标准,供读者在实际工作中参考、借鉴。

第五篇高级课题,简要地介绍软件工程研究的一些新成果,讲述了软件工程的形式化方法、面向对象的统一建模语言 UML 和软件重用。

在本书的编写过程中,蔡英协助编者查找了一些资料,张劲松和张展新用 VC++ 实现了书中讲述的 C++ 类库管理系统,张雯和张杰完成了本书的一些具体工作,谨在此向他们表示感谢。

编者
2002 年 2 月

目 录

第一篇 软件工程与软件过程

第 1 章 软件工程	1
1.1 软件危机	1
1.1.1 计算机系统的发展历程	1
1.1.2 软件危机的含义	2
1.1.3 产生软件危机的原因	3
1.1.4 消除软件危机的途径	5
1.2 软件工程	6
1.2.1 什么是软件工程	6
1.2.2 软件工程的基本原理	6
1.2.3 软件工程方法学	8
1.3 小结	10
习题	11
第 2 章 软件过程	13
2.1 软件生命周期的基本任务	13
2.2 瀑布模型	16
2.3 快速原型模型	18
2.4 增量模型	19
2.5 螺旋模型	20
2.6 喷泉模型	22
2.7 小结	23
习题	24

第二篇 传统方法学

第 3 章 结构化分析	25
3.1 概述	25
3.2 与用户通信的技术	26
3.2.1 访谈	26
3.2.2 简易的应用规格说明技术	27
3.2.3 软件原型	28
3.3 分析建模与规格说明	29
3.3.1 分析建模	29

3.3.2 软件需求规格说明	29
3.4 实体—关系图	31
3.4.1 数据对象	31
3.4.2 属性	31
3.4.3 关系	32
3.4.4 实体—关系图的符号	32
3.5 数据流图	33
3.5.1 数据流图符号	33
3.5.2 例子	34
3.5.3 命名	37
3.6 状态转换图	38
3.7 数据字典	39
3.8 小结	41
习题	42
第4章 结构化设计	44
4.1 结构化设计与结构化分析的关系	44
4.2 软件设计的概念和原理	45
4.2.1 模块化	45
4.2.2 抽象	47
4.2.3 逐步求精	47
4.2.4 信息隐藏	48
4.3 模块独立	49
4.3.1 耦合	49
4.3.2 内聚	50
4.4 启发规则	51
4.4.1 改进软件结构提高模块独立性	51
4.4.2 模块规模应该适中	51
4.4.3 深度、宽度、扇出和扇入都应适当	52
4.4.4 模块的作用域应该在控制域之内	52
4.4.5 力争降低模块接口的复杂程度	53
4.4.6 设计单入口单出口的模块	53
4.4.7 模块功能应该可以预测	53
4.5 表示软件结构的图形工具	53
4.5.1 层次图和 HIPO 图	53
4.5.2 结构图	55
4.6 面向数据流的设计方法	56
4.6.1 概念	56
4.6.2 变换分析	57
4.6.3 事务分析	63

4.6.4 设计优化	64
4.7 人机界面设计	65
4.7.1 人机界面设计问题	65
4.7.2 人机界面设计过程	66
4.7.3 界面设计指南	67
4.8 过程设计	69
4.9 过程设计的工具	71
4.9.1 程序流程图	71
4.9.2 盒图(N_S图)	72
4.9.3 PAD图	72
4.9.4 判定表	74
4.9.5 判定树	75
4.9.6 过程设计语言(PDL)	75
4.10 面向数据结构的设计方法	76
4.10.1 Jackson图	76
4.10.2 改进的Jackson图	77
4.10.3 Jackson方法	77
4.11 小结	82
习题	83
第5章 结构化实现	86
5.1 编码	87
5.1.1 选择程序设计语言	87
5.1.2 编码风格	88
5.2 软件测试基础	90
5.2.1 测试目标	90
5.2.2 黑盒测试和白盒测试	91
5.2.3 测试准则	91
5.2.4 流图	92
5.3 逻辑覆盖	94
5.4 控制结构测试	96
5.4.1 基本路径测试	96
5.4.2 条件测试	99
5.4.3 数据流测试	101
5.4.4 循环测试	102
5.5 黑盒测试技术	103
5.5.1 等价划分	104
5.5.2 边界值分析	106
5.5.3 错误推测	107
5.6 测试策略	108

5.6.1 测试步骤	108
5.6.2 单元测试	109
5.6.3 集成测试	111
5.6.4 确认测试	114
5.7 调试	115
5.7.1 调试过程	116
5.7.2 调试途径	116
5.8 软件可靠性	118
5.8.1 基本概念	118
5.8.2 估算平均无故障时间的方法	119
5.9 小结	121
习题	121

第三篇 面向对象方法学

第6章 面向对象的概念与模型	127
6.1 面向对象方法学概述	127
6.2 面向对象方法学的主要优点	129
6.3 面向对象的概念	132
6.3.1 对象	132
6.3.2 其他概念	134
6.4 面向对象建模	138
6.5 对象模型	139
6.5.1 表示类—&—对象的图形符号	139
6.5.2 表示结构的图形符号	140
6.5.3 对象模型之例	142
6.6 动态模型	143
6.6.1 概念	144
6.6.2 符号	144
6.7 功能模型	145
6.7.1 表示方法	145
6.7.2 三种模型之间的关系	145
6.8 小结	146
习题	146
第7章 面向对象分析	147
7.1 分析过程	147
7.1.1 概述	147
7.1.2 三个子模型与五个层次	148
7.2 需求陈述	149
7.2.1 书写要点	149

7.2.2 例子	150
7.3 建立对象模型	151
7.3.1 确定类—&—对象	151
7.3.2 确定关联	153
7.3.3 划分主题	156
7.3.4 确定属性	156
7.3.5 识别继承关系	159
7.3.6 反复修改	159
7.4 建立动态模型	161
7.4.1 编写脚本	162
7.4.2 设想用户界面	163
7.4.3 画事件跟踪图	163
7.4.4 画状态图	164
7.4.5 审查动态模型	166
7.5 建立功能模型	167
7.5.1 画出基本系统模型图	167
7.5.2 画出功能级数据流图	167
7.5.3 描述处理框功能	167
7.6 定义服务	168
7.6.1 常规行为	169
7.6.2 从事件导出的操作	169
7.6.3 与数据流图中处理框对应的操作	169
7.6.4 利用继承减少冗余操作	169
7.7 面向对象分析实例	169
7.7.1 需求陈述	169
7.7.2 建立对象模型	170
7.7.3 建立动态模型	171
7.7.4 建立功能模型	173
7.7.5 进一步完善	173
7.8 小结	175
习题	176
第8章 面向对象设计	177
8.1 面向对象设计的准则	177
8.1.1 模块化	178
8.1.2 抽象	178
8.1.3 信息隐藏	178
8.1.4 弱耦合	178
8.1.5 强内聚	179
8.1.6 可重用	179

8.2 启发规则	179
8.2.1 设计结果应该清晰易懂	180
8.2.2 一般—特殊结构的深度应适当	180
8.2.3 设计简单的类	180
8.2.4 使用简单的协议	181
8.2.5 使用简单的服务	181
8.2.6 把设计变动减至最小	181
8.3 系统分解	181
8.3.1 子系统之间的两种交互方式	182
8.3.2 组织系统的两种方案	182
8.3.3 设计系统的拓扑结构	183
8.4 设计问题域子系统	183
8.4.1 调整需求	184
8.4.2 重用已有的类	184
8.4.3 把问题域类组合在一起	184
8.4.4 增添一般化类以建立协议	184
8.4.5 ATM 系统之例	184
8.5 设计人—机交互子系统	185
8.5.1 设计人—机交互界面的准则	185
8.5.2 设计人—机交互子系统的策略	186
8.6 设计任务管理子系统	187
8.6.1 分析并发性	188
8.6.2 设计任务管理子系统	188
8.7 设计数据管理子系统	189
8.7.1 选择数据存储管理模式	189
8.7.2 设计数据管理子系统	190
8.7.3 例子	192
8.8 设计类中的服务	192
8.8.1 确定类中应有的服务	192
8.8.2 设计实现服务的方法	193
8.9 设计关联	194
8.9.1 关联的遍历	194
8.9.2 实现单向关联	194
8.9.3 实现双向关联	195
8.9.4 链属性的实现	195
8.10 设计优化	195
8.10.1 确定优先级	195
8.10.2 提高效率的几项技术	195
8.10.3 调整继承关系	196

8.11 面向对象分析与设计实例	198
8.11.1 面向对象分析	199
8.11.2 面向对象设计	201
8.12 小结	205
习题	206
第9章 面向对象实现	207
9.1 程序设计语言	207
9.1.1 面向对象语言的优点	207
9.1.2 面向对象语言的技术特点	208
9.1.3 选择面向对象语言	211
9.2 程序设计风格	212
9.2.1 提高可重用性	212
9.2.2 提高可扩充性	214
9.2.3 提高健壮性	214
9.3 测试策略	215
9.3.1 面向对象的单元测试	215
9.3.2 面向对象的集成测试	215
9.3.3 面向对象的确认测试	216
9.4 设计测试用例	216
9.4.1 测试类的方法	216
9.4.2 集成测试方法	217
9.5 小结	219
习题	220

第四篇 软件项目管理

第10章 计划	221
10.1 度量软件规模	221
10.1.1 代码行技术	221
10.1.2 功能点技术	222
10.2 工作量估算	224
10.2.1 静态单变量模型	224
10.2.2 动态多变量模型	224
10.2.3 COCOMO 模型	225
10.3 进度计划	228
10.3.1 基本原则	229
10.3.2 Gantt 图	229
10.3.3 工程网络	230
10.3.4 估算进度	232
10.3.5 关键路径	233

10.3.6 机动时间	233
10.4 小结	235
习题	236
第 11 章 组织	237
11.1 民主制程序员组	237
11.2 主程序员组	238
11.3 现代程序员组	240
11.4 软件项目组	241
11.4.1 三种组织方式	242
11.4.2 四种组织范型	243
11.5 小结	244
习题	244
第 12 章 控制	245
12.1 风险管理	245
12.1.1 软件风险分类	245
12.1.2 风险识别	246
12.1.3 风险预测	250
12.1.4 处理风险的策略	252
12.2 质量保证	253
12.2.1 软件质量	253
12.2.2 软件质量保证措施	254
12.3 配置管理	257
12.3.1 软件配置	257
12.3.2 软件配置管理过程	259
12.4 小结	264
习题	264
第 13 章 国际标准	266
13.1 IEEE 1058.1 软件项目管理计划标准	266
13.1.1 软件项目管理计划的组成	266
13.1.2 IEEE 软件项目管理计划	267
13.2 ISO9000 质量标准	269
13.2.1 基本思想	269
13.2.2 ISO9000-3 标准	270
13.3 ISO/IEC 12207 软件生命周期过程标准	272
13.3.1 概述	272
13.3.2 软件生命周期过程	273
13.4 ISO/IEC TR 15504 软件过程评估标准	276
13.4.1 概述	277
13.4.2 标准的结构	277

13.5 能力成熟度模型	278
13.5.1 能力成熟度模型的结构	279
13.5.2 能力成熟度等级	280
13.5.3 关键过程域	281
13.5.4 应用 CMM	282
13.6 小结	282
习题	283
第五篇 高级课题	
第 14 章 形式化方法	285
14.1 概述	285
14.1.1 非形式化方法的缺点	285
14.1.2 软件开发过程中的数学	286
14.1.3 应用形式化方法的准则	286
14.2 有穷状态机	287
14.2.1 基本概念	287
14.2.2 电梯问题	288
14.2.3 评论	291
14.3 Petri 网	291
14.3.1 基本概念	291
14.3.2 应用实例	293
14.4 Z 语言	294
14.4.1 简介	294
14.4.2 评论	296
14.5 小结	297
习题	297
第 15 章 统一建模语言 UML	299
15.1 概述	299
15.1.1 UML 的产生和发展	299
15.1.2 UML 的结构	300
15.1.3 UML 的图	301
15.1.4 UML 的应用领域	302
15.2 静态建模机制	303
15.2.1 用例图	303
15.2.2 类图和对象图	306
15.3 动态建模机制	314
15.3.1 消息	314
15.3.2 状态图	314
15.3.3 顺序图	316

15.3.4	协作图	317
15.3.5	活动图	318
15.4	描述物理架构的机制	319
15.4.1	逻辑架构和物理架构	319
15.4.2	构件图	319
15.4.3	配置图	320
15.5	使用和扩展 UML	321
15.5.1	使用 UML 的准则	321
15.5.2	扩展 UML 的机制	322
15.6	小结	323
	习题	323
第 16 章	软件重用	324
16.1	可重用的软件成分	324
16.2	软件重用过程	325
16.2.1	构件组装模型	325
16.2.2	类构件	326
16.2.3	重用过程模型	327
16.3	领域工程	328
16.3.1	分析过程	328
16.3.2	领域特征	329
16.3.3	结构建模和结构点	330
16.4	开发可重用的构件	330
16.4.1	为了重用的分析与设计	330
16.4.2	基于构件的开发	331
16.5	分类和检索构件	332
16.5.1	描述可重用的构件	333
16.5.2	重用环境	335
16.6	软件重用的效益	335
16.7	小结	336
	习题	337
	参考文献	338

第一篇 软件工程与软件过程

第 1 章 软件工程

人类社会已经跨入了 21 世纪,计算机系统的发展也经历了四个不同的阶段,但是,我们仍然没有彻底摆脱“软件危机”的困扰,软件已经成为限制计算机系统发展的关键因素。

为了更有效地开发与维护软件,软件工作者在 20 世纪 60 年代后期开始认真研究消除软件危机的方法,从而逐渐形成了计算机科学技术领域中一门新兴的工程学科——软件工程。

1.1 软件危机

1.1.1 计算机系统的发展历程

所谓计算机系统就是指适当地组织在一起的一系列系统元素的集合,这些系统元素互相配合、相互协作,通过对信息的处理而完成预先定义的目标。计算机系统中通常包含的系统元素有:计算机软件、计算机硬件、人员、数据库、文档和过程。其中,软件是程序、数据结构和相关文档的集合,用于实现所需要的逻辑方法、过程或控制;硬件是提供计算能力的电子设备和提供外部世界功能的电子机械设备(例如传感器、马达、水泵等);人员是硬件和软件的用户和操作人员;数据库是通过软件访问的大型的、有组织的信息集合;文档是描述系统使用的手册、表格、图形及其他描述性信息;过程是一系列步骤,它们定义了每个系统元素的特定使用方法或系统驻留的过程性语境。

迄今为止,计算机系统已经经历了四个不同的发展阶段。下面简要地介绍各个发展阶段的特点,以便读者了解产生软件危机的背景。

60 年代中期以前,是计算机系统发展的早期时代。在这个时期通用硬件已经相当普遍,软件却是为每个具体应用而专门编写的,大多数人认为软件开发是无需预先计划的事情。这时的软件实际上就是规模较小的程序,程序的编写者和使用者往往是同一个(或同一组)人。由于规模小,程序编写起来相当容易,也没有什么系统化的方法,对软件开发工作更没有进行任何管理。这种个体化的软件环境,使得软件设计往往只是在人们头脑中隐含进行的一个模

糊过程,除了程序清单之外,根本没有其他文档资料保存下来。

从 60 年代中期到 70 年代中期,是计算机系统发展的第二代。在这 10 年中计算机技术有了很大进步。多道程序、多用户系统引入了人机交互的新概念,开创了计算机应用的新境界,使硬件和软件的配合上了一个新的层次。实时系统能够从多个信息源收集、分析和转换数据,从而使得进程控制能以毫秒而不是分钟来进行。在线存储技术的进步导致了第一代数据库管理系统的出现。

计算机系统发展的第二代的一个重要特征是出现了“软件作坊”,广泛使用产品软件。但是,“软件作坊”基本上仍然沿用早期形成的个体化软件开发方法。随着计算机应用的日益普及,软件数量急剧膨胀。在程序运行时发现的错误必须设法改正;用户有了新的需求时必须相应地修改程序;硬件或操作系统更新时,通常需要修改程序以适应新的环境。上述种种软件维护工作,以令人吃惊的比例耗费资源。更严重的是,许多程序的个体化特性使得它们最终成为不可维护的。“软件危机”就这样开始出现了。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,在这次会议上正式提出并使用了“软件工程”这个名词,一门新兴的工程学科就此诞生了。

计算机系统发展的第三代从 20 世纪 70 年代中期开始,并且跨越了整整 10 年。在这 10 年中计算机技术又有了很大进步。分布式系统极大地增加了计算机系统的复杂性,局域网、广域网、宽带数字通信以及对“即时”数据访问需求的增加,都对软件开发者提出了更高的要求。但是,在这个时期软件仍然主要在工业界和学术界应用,个人应用还很少。

这个时期的主要特点是出现了微处理器,而且微处理器获得了广泛应用。以微处理器为核心的“智能”产品随处可见,当然,最重要的智能产品是个人计算机。在不到 10 年的时间里,个人计算机已经成为大众化的商品。

在计算机系统发展的第四代已经不再看重单台计算机和程序,人们感受到的是硬件和软件的综合效果。由复杂操作系统控制的强大的桌面机及局域网和广域网,与先进的应用软件相配合,已经成为当前的主流。计算机体系结构已迅速地从集中的主机环境转变成分布的客户机/服务器(或浏览器/服务器)环境。世界范围的信息网为人们进行广泛交流和资源的充分共享提供了条件。

软件产业在世界经济中已经占有举足轻重的地位。随着时代的进步,新的技术也不断地涌现出来。面向对象技术(本书第三篇将系统地讲述)已经在许多领域迅速地取代了传统的软件开发方法。

软件开发的“第四代技术”改变了软件界开发计算机程序的方式。专家系统和人工智能软件终于从实验室中走出来进入了实际应用,解决了大量实际问题。应用模糊逻辑的人工神经网络软件,展现了模式识别与拟人信息处理的美好前景。虚拟现实技术与多媒体系统,使得与用户的通信可以采用和以前完全不同的方法。遗传算法使我们有可能开发出驻留在大型并行生物计算机上的软件。

1.1.2 软件危机的含义

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。这些问题不仅仅是不能正常运行的软件才具有的,实际上,几乎所有软件都不同程度地存在这些问题。

概括地说,软件危机包含下述两方面的问题:如何开发软件,以满足对软件日益增长的需求