

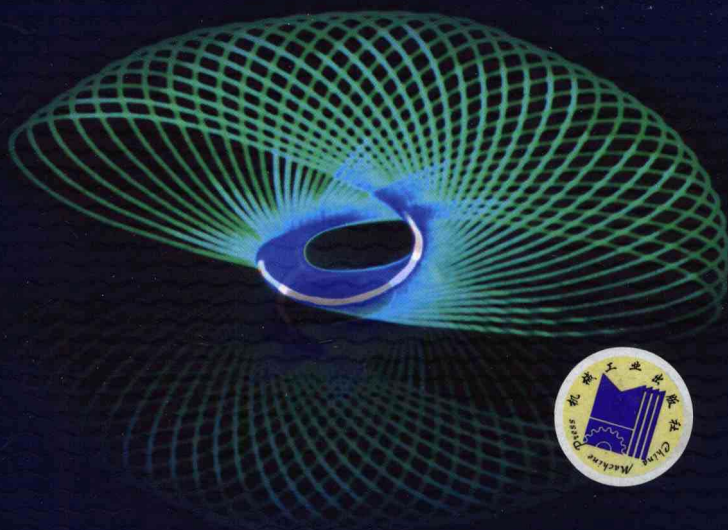
面向21世纪高等院校计算机系列教材



软件工程

——原理、方法和工具

李东生 崔冬华 李爱萍 等编著



机械工业出版社
CHINA MACHINE PRESS

面向 21 世纪高等院校计算机系列教材

软 件 工 程

——原理、方法和工具

李东生 崔冬华 李爱萍 等编著

机 械 工 业 出 版 社

本书全面系统地介绍了软件工程的观念、原理和技术。全书分3篇,第1篇(第1~8章)“面向过程的软件工程”,以传统的软件生存周期为主线,介绍软件工程的原理和方法,讲述了软件生存周期各阶段的任务、过程和方法。第2篇(第9~16章)“面向对象的软件工程”,以UML语言为主线,结合当前流行的软件工程工具 Rational Rose,介绍面向对象软件工程的原理和方法。第3篇(第17章)“软件工程实验”,是在前两篇的理论知识基础上,介绍图书馆图书管理系统和浏览器系统实验。

本书的特色是将软件工程理论与软件工程实际紧密结合,便于教学与应用,既可以作为高等院校计算机专业和软件工程专业本科生的教材,也可作为工程技术人员的参考书。

图书在版编目(CIP)数据

软件工程——原理、方法和工具/李东生等编著. —北京:机械工业出版社, 2009. 1

(面向21世纪高等院校计算机系列教材)

ISBN 978-7-111-25740-0

I. 软… II. 李… 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(2008)第194020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:陈皓

责任印制:李妍

北京富生印刷厂印刷

2009年1月第1版·第1次印刷

184mm×260mm·19.25印张·473千字

0001—4000册

标准书号:ISBN 978-7-111-25740-0

定价:32.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话:(010) 68326294 68993821

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379753 88379739

封面防伪标均为盗版

面向 21 世纪高等院校计算机系列教材
教材编委会

顾问委员：刘开瑛 刘 璟 李东福 施伯乐 谢克昌

主任委员：左孝凌

副主任委员：陈立潮 陈俊杰 余雪丽 李焕珍 梁吉业 曾建潮

委员：马尚才 亢临生 左孝凌 刘晓融 陈立潮 陈俊杰 李东生

李济洪 李焕珍 余雪丽 张荣国 张继福 杨 威 贺利坚

段 富 陶世群 梁吉业 曾建潮 谢康林 韩 燮 缪淮扣

序

在高度信息化的 21 世纪，人们越来越认识到信息教育的重要性。人们都迫切希望信息教育能有较大发展。教育信息化也是摆在我们面前的重要任务。教育部明确要求高等教育实行信息化，要求在未来 5 年内实现信息化教育课程的数量达到 15% ~ 30%。信息社会离不开计算机技术，知识经济需要大量的计算机高级人才。我国正在加强计算机的高等教育，正着眼于为新世纪培养高素质的计算机人才，以适应信息社会高速发展的需要。当前，全国各类高等院校都在各专业基础课程计划中增加计算机的课程内容，而作为与计算机科学密切相关的计算机、通信、信息等专业，更是在酝酿着教学的全面改革，以期规划出一整套面向 21 世纪的、具有中国高校计算机教育特色的课程计划和教材体系。

教育部《关于加强高等学校本科教育工作提高教育质量的若干意见》（教字【2001】4 号）文件也强调指出：“要大力提倡编写、引进和使用先进教材。教材的质量直接体现着高等教育和科学研究的发展水平，也直接影响本科教学的质量。高等学校要结合学科、专业的调整，加快教材的更新换代。”

为推动高校教学改革，提高教学质量，我们重点抓了 21 世纪高等教育教学改革项目，组织并支持了“面向 21 世纪计算机系列教材规划”研究课题。该课题组成员均由高校计算机系的专家教授组成。他们有多年的教学经验，也很强的科研能力。该课题的主要目标是密切结合国民经济的需要，优化计算机教材体系结构，力求将国际、国内计算机领域的新概念、新理论、新技术吸收到本系列教材中，编写出具有科学性、先进性、系统性、实用性、实践性很强的教材，经过推广使用，反复修改，不断提高。

“面向 21 世纪计算机系列教材规划”课题以编写非计算机专业的计算机课程、计算机专业的计算机网络课程、计算机软件课程三个系列教材为主要内容，计划在三年内出版 13 ~ 16 种书，服务于本科生、专科生、研究生，以及网络学院和软件学院的学生。本课题把研究系列教材的重点放在影响和带动计算机学科发展的网络与软件，以及直接推动计算机普及和应用的非计算机专业三个方向上，目的是通过集中优势兵力，加强团队协作，能够在教材建设方面按系列有所突破。

相信本套教材的出版必将对教学改革和教材建设起到很大的推动和示范作用。


2002 年 6 月 10 日

前 言

软件工程是指导计算机软件开发的一门工程学科，软件工程采用工程的概念、原理、技术和方法来开发软件。软件工程的方法和技术的发展经历了一个漫长的过程，历经三十余年的研究和发展，软件工程学科正逐步走向成熟。软件工程学科把经过时间考验而证明行之有效的管理技术和方法结合起来，能够高效地开发出高质量的软件，从而避免软件开发的随意性。因此，认真学习并在实际工作中运用软件工程的原理、方法和工具，对软件行业的从业人员，特别是对于相关专业的本科生、研究生是十分重要的。

本书由 3 篇共计 17 章组成，按专题安排，便于授课教师和读者按需选用。

第 1 篇为面向过程的软件工程，以传统的软件生存周期为主线，介绍软件工程的原理和方法。其中，第 1 章介绍了软件工程、软件危机的基本概念；第 2~8 章按照软件生存周期的各个阶段顺序介绍了可行性研究、需求分析、软件总体设计、详细设计、软件编码、软件测试和软件维护，详细介绍了有关概念和软件工程方法，重点介绍了结构化的分析和设计方法。

第 2 篇为面向对象的软件工程，以 UML 为主线，结合当前流行的 Rational Rose 工具，介绍面向对象软件工程的原理和方法。其中，第 9 章为 UML 概述与 UML 开发工具简介；第 10 章为面向对象方法学以及面向对象的基本概念；第 11 章介绍了 UML 通用对象设计模式及应用，以及对象类的高级概念；第 12 章介绍了面向对象的软件开发过程，包括 RUP 概述、RUP 生命周期、建模的概念，并给出一个面向对象软件开发过程案例的简单分析；第 13 章介绍了软件工具 Rose 中各种 UML 图形的画法；第 14 章介绍了软件工具 Rose 的整体功能，并以一个教学管理系统实例介绍了使用 Rose 完成整个系统的设计和实现的全过程；第 15 章介绍了几种典型的面向对象软件体系结构；第 16 章介绍了软件工程领域的一些最新技术。

第 3 篇为软件工程实验，通过两个典型的案例说明了面向过程和面向对象两类软件工程实验的具体实施步骤。

本书由长期从事软件工程教学和科研的教师编写。李东生、崔冬华负责全书架构的构思和统稿。崔冬华编写了本书第 1 篇的第 1~4 章，武昭宇编写了本书第 1 篇的第 5~7 章，李新编写了本书第 2 篇的第 9、10 章和附录，李东生编写了本书第 2 篇的第 11、12 章，李爱萍编写了本书第 1 篇的第 8 章，第 2 篇的第 13~16 章以及第 3 篇。

本书的内容涵盖了传统的软件工程以及面向对象软件工程的基本原理和方法，详细介绍了当前流行的 UML 及软件工程工具 Rational Rose，做到软件工程的基本原理、方法、工具和实践密切结合，便于组织教学和读者掌握相关知识。

本书编写过程中参考了国内外有关软件工程的专著、教材和论文，详见书后所附的主要参考文献。在此，向所有作者表示感谢。

段利国、丁秀兰、朱兰强、候晓芳、肖燕锋、蔺兢等参与了书稿文字的录入、修改、校对和插图的绘制等工作，在此表示感谢。

由于软件工程近年来的快速发展以及作者水平所限，书中难免存在错误和不足之处，敬请读者提出宝贵意见和建议。

编 者

目 录

序
前言

第1篇 面向过程的软件工程

第1章 软件危机、软件工程	1
1.1 软件工程的发展史	1
1.1.1 程序设计时代	1
1.1.2 程序系统时代	1
1.1.3 软件工程时代	2
1.2 软件危机主要表现形式	2
1.3 产生软件危机的原因及解决途径	3
1.3.1 产生软件危机的原因	3
1.3.2 解决软件危机的途径	3
1.4 软件和软件工程	4
1.4.1 软件	4
1.4.2 软件工程	4
1.5 软件质量	5
1.6 软件的生存周期及开发模型	5
1.6.1 软件生存周期	5
1.6.2 软件开发模型	7
1.7 习题	12
第2章 可行性研究	13
2.1 可行性研究的目的与任务	13
2.2 可行性研究的步骤	13
2.3 系统流程图	14
2.3.1 系统流程图的符号	15
2.3.2 系统流程图示例	16
2.4 成本-效益分析	16
2.4.1 货币的时间价值	16
2.4.2 投资回收期	17
2.4.3 纯收入	17
2.5 可行性研究报告的主要内容	17
2.6 习题	18
第3章 软件需求分析	19
3.1 需求分析的任务和步骤	19



3.1.1 需求分析的任务	19
3.1.2 需求分析的步骤	20
3.2 需求获取的常用方法	21
3.2.1 常规的需求获取方法	21
3.2.2 快速建立软件原型来获取需求	22
3.3 需求分析的方法	23
3.3.1 功能分解方法	23
3.3.2 结构化分析方法	23
3.3.3 信息建模方法	23
3.3.4 面向对象的分析	24
3.4 结构化分析方法	24
3.4.1 自顶向下逐层分解的分析策略	24
3.4.2 结构化分析描述工具	25
3.4.3 数据流图	25
3.4.4 数据字典	28
3.4.5 加工逻辑的描述	30
3.5 需求分析图形工具	32
3.5.1 层次方框图	32
3.5.2 Warnier 图	33
3.5.3 IPO 图	33
3.6 SA 方法的应用	34
3.7 习题	39
第4章 软件总体设计	40
4.1 软件总体设计的目标和任务	40
4.2 软件设计的概念和原理	40
4.2.1 模块和模块化	40
4.2.2 抽象	41
4.2.3 信息隐蔽和局部化	42
4.2.4 模块独立性及其度量	42
4.3 软件结构设计准则	44
4.4 软件结构设计的图形工具	46
4.4.1 软件结构图	46
4.4.2 层次图	47
4.4.3 HIPO 图	47
4.5 结构化设计方法	48
4.5.1 数据流图的类型	48
4.5.2 结构化设计方法的步骤	49
4.5.3 变换型分析设计	49
4.5.4 事务型分析设计	51

4.6 习题	52
第5章 软件详细设计	54
5.1 详细设计的目的与任务	54
5.2 结构化程序设计	54
5.3 详细设计工具	55
5.3.1 程序流程图	55
5.3.2 N-S图	55
5.3.3 PAD图	56
5.3.4 过程设计语言	57
5.4 习题	60
第6章 软件编码	62
6.1 程序设计语言的分类	62
6.1.1 基础语言	62
6.1.2 结构化语言	63
6.1.3 面向对象的语言	64
6.2 程序设计语言的选择	65
6.3 程序设计风格	66
6.3.1 程序内部文档	66
6.3.2 数据说明	67
6.3.3 语句构造	67
6.3.4 输入/输出	67
6.3.5 效率	68
6.4 习题	69
第7章 软件测试	70
7.1 软件测试的目标	70
7.2 软件测试的原则	70
7.3 软件测试方法	71
7.3.1 静态测试与动态测试	71
7.3.2 黑盒测试法与白盒测试法	72
7.4 软件测试用例的设计	73
7.4.1 白盒技术	73
7.4.2 黑盒技术	78
7.5 软件测试过程	82
7.5.1 单元测试	82
7.5.2 集成测试	84
7.5.3 确认测试	86
7.5.4 系统测试	87
7.6 调试	87
7.6.1 调试的目的	87

7.6.2 调试技术	87
7.7 习题	89
第8章 软件维护	90
8.1 软件维护的分类	90
8.2 软件维护的特点	91
8.2.1 结构化维护与非结构化维护	92
8.2.2 维护的代价	93
8.2.3 软件维护中存在的问题	93
8.3 软件可维护性	94
8.3.1 软件可维护性的定义	94
8.3.2 软件可维护性的度量	96
8.3.3 提高软件可维护性的方法	96
8.4 维护的副作用	98
8.5 软件再工程	99
8.5.1 软件再工程与逆向工程的概念	99
8.5.2 实施软件再工程的原因	100
8.5.3 软件再工程技术	101
8.6 习题	102

第2篇 UML 与面向对象的软件工程

第9章 UML 简介	103
9.1 UML 概述	103
9.1.1 UML 的组成	104
9.1.2 UML 的特点和用途	107
9.1.3 UML 的模型视图简介	108
9.2 UML 软件开发工具简介	110
9.2.1 Rational Rose	110
9.2.2 Visio 简介	114
9.3 习题	114
第10章 面向对象的概念	115
10.1 面向对象的方法学	115
10.1.1 面向对象建模	115
10.1.2 面向对象的方法与传统软件方法的比较	118
10.2 对象与类及其 UML 表示	119
10.2.1 对象	119
10.2.2 类与实例	120
10.2.3 对象属性与操作	120
10.2.4 对象类的关联	123
10.3 聚集、组合、继承和多态	127

10.3.1 聚集与组合	127
10.3.2 抽象与继承	129
10.3.3 多态	131
10.4 习题	132
第 11 章 对象设计模式	133
11.1 对象设计模式概念	133
11.1.1 历史背景	133
11.1.2 对象设计模式	134
11.1.3 设计模式的分类	135
11.2 几种典型的对象设计模式及应用	136
11.2.1 行为型模式中的职责链模式及应用	136
11.2.2 结构型模式中的外观模式及应用	138
11.2.3 创建型模式中的抽象工厂模式及应用	142
11.3 对象类的高级概念	146
11.3.1 抽象类、参数对象类、型与实现对象类	146
11.3.2 接口	147
11.3.3 版型	148
11.4 组件、包和结点	149
11.4.1 组件	149
11.4.2 包和包图	153
11.4.3 结点	159
11.5 习题	160
第 12 章 面向对象的软件开发过程	162
12.1 RUP 概述	162
12.1.1 RUP 的历史	162
12.1.2 RUP 的特色	163
12.2 RUP 软件开发生命周期和建模	164
12.2.1 RUP 软件开发生命周期	164
12.2.2 RUP 的动态结构	165
12.2.3 RUP 的静态结构	165
12.2.4 RUP 的建模	166
12.3 面向对象软件开发过程的案例分析	167
12.3.1 系统需求	167
12.3.2 系统的静态结构模型	168
12.3.3 顺序图和协作图	168
12.3.4 状态图和活动图	170
12.3.5 组件图和部署图	171
12.4 习题	172
第 13 章 软件开发工具 Rose	173

13.1	Rose 的用例图与类图	173
13.1.1	用例图	174
13.1.2	类图	176
13.2	Rose 的交互图与状态机图	177
13.2.1	交互图	177
13.2.2	状态机图	180
13.3	包、组件图与部署图	182
13.3.1	包	182
13.3.2	组件图	183
13.3.3	部署图	185
13.4	Rose 的代码生成和逆向工程	187
13.4.1	代码生成	187
13.4.2	逆向工程	188
13.5	Rose 应用举例	188
13.6	习题	189
第 14 章	Rose 及其应用	190
14.1	Rose 的主要功能	190
14.2	Rose 的使用	190
14.2.1	Rose 主菜单窗口	190
14.2.2	模型与工作方式的组织	192
14.3	教学管理系统的分析与设计	193
14.3.1	系统需求分析	193
14.3.2	系统问题领域分析	194
14.3.3	静态结构模型的建立	198
14.3.4	动态行为模型的建立	201
14.3.5	物理模型的建立	205
14.4	UML 类图与 VB 代码的转换	206
14.4.1	VB 代码生成属性	206
14.4.2	Rose 模型—VB 代码的生成	207
14.4.3	VB 代码的逆向工程	207
14.5	UML 类图与 Java 代码的转换	208
14.5.1	UML 类图—Java 代码的生成	208
14.5.2	Java 代码的逆向工程	209
14.6	习题	209
第 15 章	面向对象的软件体系结构	211
15.1	软件体系结构概述	211
15.2	流程处理与客户机/服务器体系结构	212
15.2.1	流程处理	212
15.2.2	客户机/服务器体系结构	213

15.3	三层和多层体系结构	214
15.3.1	三层体系结构	215
15.3.2	浏览器/服务器体系结构风格	216
15.3.3	多层体系结构	217
15.3.4	团聚与串行	218
15.4	公共对象请求代理体系结构	219
15.5	基于层次消息总线的体系结构风格	221
15.6	异构结构风格	222
15.7	习题	224
第 16 章	软件工程新技术	225
16.1	软件复用技术	225
16.1.1	软件复用概念及分类	225
16.1.2	软件复用的关键技术和复用粒度	226
16.2	中间件技术	226
16.2.1	中间件概念及特点	226
16.2.2	中间件的分类	228
16.3	组件技术	229
16.3.1	组件与组件化	230
16.3.2	组件模型及描述语言	231
16.3.3	组件的检索与组装	233
16.4	计算机辅助软件工程技术	237
16.4.1	CASE 的基本概念	237
16.4.2	CASE 工具与集成 CASE 环境	237
16.5	软件产品线技术	240
16.5.1	软件产品线基本概念	240
16.5.2	软件产品线方法	243
16.5.3	北大青鸟工程	243
16.6	软件过程与标准化	245
16.6.1	软件过程及其改进	245
16.6.2	ISO 9000 标准	246
16.6.3	软件能力成熟度模型	247
16.7	习题	250

第 3 篇 软件工程实验

第 17 章	软件工程实验大纲	251
17.1	面向过程的软件工程实验	252
17.1.1	实验 1 图书馆图书管理系统的可行性分析	253
17.1.2	实验 2 图书馆图书管理系统的项目开发计划	253
17.1.3	实验 3 图书馆图书管理系统的需求分析	254

17.1.4	实验4 图书馆图书管理系统的总体设计	254
17.1.5	实验5 图书馆图书管理系统的详细设计及编码实现	255
17.1.6	实验6 图书馆图书管理系统的系统测试	256
17.1.7	实验7 图书馆图书管理系统用户手册的撰写	257
17.1.8	实验8 图书馆图书管理系统项目开发总结报告的撰写	257
17.2	面向对象的软件工程实验	257
17.2.1	实验1 浏览器系统的可行性分析	258
17.2.2	实验2 浏览器系统的项目开发计划	258
17.2.3	实验3 浏览器系统的需求分析	259
17.2.4	实验4 浏览器系统的体系结构设计	265
17.2.5	实验5 采用面向对象方法进行浏览器系统的详细设计	267
17.2.6	实验6 利用 Rational Rose 正向工程功能完成编码的实验	269
17.2.7	实验7 利用 Rational Rose 逆向工程完成修改设计的实验	270
17.2.8	实验8 浏览器系统的测试	270
17.2.9	实验9 浏览器系统用户手册的撰写	272
17.2.10	实验10 浏览器系统项目开发总结报告的撰写	272
附录		273
附录 A	UML 图总结	273
附录 B	UML 中定义的常用版型、约束和标记	278
附录 C	GOF 给出的软件设计模式	280
参考文献		291

第 1 篇 面向过程的软件工程

第 1 章 软件危机、软件工程

“软件工程”一词是由北大西洋公约组织(North Atlantic Treaty Organization, NATO)的计算机科学家在联邦德国召开的国际会议上首次提出来的。产生软件工程这门学科的时代背景是“软件危机”。软件工程的发展和应用不仅缓和了软件危机,而且促使一门新兴的工程学科诞生了。

1.1 软件工程的发展史

软件工程是随着计算机系统的发展而逐步形成的计算机科学领域中一门新兴的学科。软件工程的发展经过了三个时代。

1.1.1 程序设计时代

1946 年到 1956 年为程序设计时代,在这个时代,人们用很大力气研究和发展计算机硬件,经历了从电子管计算机到晶体管计算机的变革,然而对计算机软件的研究和发展却不够重视。当时,由于硬件的价格昂贵,运行速度低,内存容量小,所以,当时的程序员非常强调“程序设计技巧”,把缩短每一个微秒的 CPU 时间和节省每一个二进制存储单元,作为程序设计的重要目标。但设计的程序难读、难懂、难修改。

1.1.2 程序系统时代

1956 年到 1968 年为程序系统时代,或者称为“程序 + 说明”时代。在这个时代,硬件经历了从晶体管计算机到集成电路计算机的变革,CPU 速度和内存容量都有了很大的提高,从而为计算机在众多领域中的应用提供了潜在的可能性。这个时代的另一个重要特征是出现了“软件作坊”。这是因为随着计算机应用的普及和深化,需要的软件往往规模相当庞大,以致单个用户无法开发,此外,许多不同的部门和企业往往需要相同或者类似的软件,各自开发就会浪费很大的人力。在这种形势下,“软件作坊”就应运而生了。不过这个时代的开发方法基本上沿用了程序设计时代的开发方法,但开始提出了结构化的方法。随着计算机应用的日益普及,软件需求量急剧增长,用户的需求和使用环境发生变化时,软件可修改性很差,往往需要重新编制程序,其研制时间很长,不能及时满足用户要求,质量得不到保证,开发人员的素质和落后的开发技术不适应规模大、结构复杂的软件开发,因此,产生了尖锐的矛盾,所谓的“软件危机”便由此产生了。

IBM 公司的 360 OS 和美国空军后勤系统,在开发过程中都花费了几千人的工作量,最后以失败告终。其中 360 OS 由 4000 个模块组成,共约 100 万条指令,花费了 5000 人的工作

量,经费达数千万美元,拖延几年才交付使用,交付使用后每年发现近 100 个错误,结果以失败告终。360 OS 开发负责人 Brooks 生动描述了研制过程中的困难和混乱:“……像巨兽陷入泥潭作垂死挣扎,挣扎得越猛,泥浆就沾得越多,最后没有一个野兽能逃脱淹没在泥潭中的命运……。程序设计就像这样的泥潭,一批批程序员在泥潭中挣扎……。没有料到问题会这样棘手……”。比 360 OS 更糟的软件系统并不少,即花费大量的人力、物力、财力结果半途而废,或者说完成之日就是遗弃之时。这就是人们常说的“软件危机”。

1.1.3 软件工程时代

1968 年至今为软件工程时代,或者称为“程序+文档”时代。这个时代,硬件发展的特点是集成电路计算机发展到超大规模集成电路计算机,高性能低成本的微处理机大量出现,硬件速度的发展已经超过人们提供支持软件的能力。然而,硬件只提供了潜在的计算能力,对于复杂的大型软件开发项目,需要十分复杂的计算机软件才能实现。也就是说,如果没有软件来驾驭和开发的这种能力,人类并不能有效地使用计算机。在这个时代,软件维护费用、软件价格不断上升。没有完全摆脱软件危机。

1.2 软件危机主要表现形式

1. 软件危机的概念

所谓“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。软件危机主要包含两方面的问题:一是如何开发软件以满足对软件日益增长的需求;二是如何维护数量不断增长的已有软件。

2. 软件危机的主要表现形式

软件危机的主要表现形式如下:

1) 对软件开发成本和研制进度的估计常常很不精确。经费预算经常突破,完成时间一拖再拖。这种现象降低了软件开发组织的信誉。而有时为了赶进度和节约成本,所采取的一些权宜之计又往往影响了软件产品的质量,从而不可避免地会引起用户的不满。

2) “已完成”的软件不能满足用户要求。软件开发人员常常在对用户需求只有模糊的了解,甚至对所要解决的问题还没有确切认识的情况下,就匆忙着手编写程序了。软件开发人员和用户又未能及时交换意见,使得一些问题不能得到及时解决,导致开发的软件不能满足用户要求,使得开发失败。

3) 软件产品质量差,可靠性得不到保证。软件质量保证技术(审查、复审和测试)还没有坚持不懈地应用到软件开发的全过程中,提交给用户的软件质量差,在运行中暴露了大量的问题。

4) 软件产品可维护性差。软件开发人员在开发过程中按各自的风格工作,没有统一、公认的规范和完整规范的文档,发现问题后进行杂乱无章的修改。程序结构不好,运行时发现错误也很难修改,导致维护性差。

5) 软件成本在计算机系统总成本中所占的比例逐年上升。软件的发展跟不上硬件的发展。由于微电子技术的进步和生产自动化程度不断地提高,硬件成本逐年下降,然而软

件开发需要大量人力，软件成本也随着通货膨胀以及软件规模和数量的不断扩大而持续上升。

6) 软件开发生产率提高的速度远远跟不上计算机应用速度普及深入的趋势。软件的发展跟不上用户的要求。软件产品“供不应求”的现象使人类不能充分利用现代计算机硬件提供的巨大潜力。

以上列举的仅仅是软件危机的典型表现，与软件开发和维护有关的问题远远不止这些。

1.3 产生软件危机的原因及解决途径

1.3.1 产生软件危机的原因

造成上述软件危机的原因与软件自身的特点有关，也与软件开发人员在开发和维护时所采用的生产方式、方法、技术有关。可概括为以下几方面：

1) 软件是计算机系统逻辑部件，软件产品往往规模庞大，结构复杂。给软件的开发和维护带来客观的困难。

2) 软件开发的管理困难。由于软件规模大、结构复杂，又具有无形性，导致管理困难，进度控制困难，质量控制困难，可靠性无法保证。

3) 软件开发费用不断增加，维护费用急剧上升，直接威胁计算机应用的扩大。

4) 软件开发技术落后。在 20 世纪 60 年代，人们注重如编译原理、操作系统原理、数据库原理等一些计算机理论问题的研究，而不注重软件开发技术的研究，用户要求的软件复杂性与软件技术解决复杂性的能力不相适应。

5) 生产方式落后。有人统计，硬件的性能价格比在过去 30 年中增长了 10^6 倍。一种新器件的出现，其性能较旧器件提高，价格反而有所下降，而软件则相形见绌。一方面，软件规模与复杂性增长了几个数量级，但生产方式仍然采用个体手工方式开发，根据个人习惯爱好工作，无章可循，无规范可依靠，带有很强的“个性化”特征的程序，因缺乏文档而根本不能维护，加剧了供需之间的矛盾。

6) 开发工具落后，生产效率提高缓慢。软件开发工具原始，没有出现高效率的开发工具，因而软件生产效率低下。另外，软件开发人员忽视需求分析的重要性，轻视软件维护也是造成软件危机的原因。

1.3.2 解决软件危机的途径

目前，计算机的应用日益广泛，世界上发达国家的许多企业将全部投资的 10% 以上用于计算机。但到目前为止，计算机的体系结构在硬件上仍然是冯·诺依曼计算机。实际中复杂、庞大的问题只能由专门人员编制软件来解决。假设计算机能实现智能化，能进行推理和运算，正确解决用户所提出的问题，那么软件危机就会有根本性的缓解。然而，新一代计算机体系结构的研制可能还需要一段时间。在目前的计算机硬件条件下，要想解决软件危机必须解决以下问题：

1) 首先应该对计算机软件有一个正确的认识，彻底清除“软件就是程序”的错误观念。

2) 要使用好的开发技术和方法，并且要不断研究探索更好、更有效的方法和技术。尽快