

附：软件工程自学考试大纲

软件工程

组编 / 全国高等教育自学考试指导委员会
主编 / 陆丽娜

计算机及应用专业

(独立本科段)

全国高等教育自学考试指定教材

经济科学出版社

全国高等教育自学考试指定教材

计算机及应用专业（独立本科段）

软 件 工 程

（附：软件工程自学考试大纲）

全国高等自学考试指导委员会组编

陆丽娜 主编

陆丽娜

邓良松 编著

刘海岩

经济科学出版社

图书在版编目(CIP)数据

软件工程/陆丽娜主编, —北京: 经济科学出版社, 1999.12

ISBN 7-5058-1970-4

I. 软件… II. 陆… III. 软件工程-高等教育-自学考试-教材 IV. TP311.5

中国版本图书馆 CIP 数据核字 (1999) 第 52627 号

软 件 工 程

(附软件工程自学考试大纲)

全国高等自学考试指导委员会组编

陆丽娜 主编

陆丽娜 邓良松 刘海岩 编著

经济科学出版社出版

社址: 北京海淀区万泉河路 66 号 邮编: 100086

网址: www.esp.com.cn

电子邮件: esp@public2.east.net.cn

北京市鑫鑫印刷厂印刷

787×1092 16开 15.875印张 300000字

2000年3月第一版 2000年3月第一次印刷

印数: 001—10100册

ISBN 7-5058-1970-4/G·415 定价: 21.00 元

(图书出现印装问题, 请与当地教材供应部门调换)

(版权所有 翻印必究)

内 容 简 介

本书主要讲述建造软件系统的方法、技术、流程、工具、规范等等，使学生能够掌握软件工程基本概念，基本原理，实用开发方法，如何用工程化的方法开发软件项目。

本书为全国高等教育自学考试教材，适用于自学者和软件工程开发人员的自学教程和参考书。

组 编 前 言

当您开始阅读本书时，人类已经迈入了二十一世纪。

这是一个变幻难测的世纪，这是一个催人奋进的时代。科学技术飞速发展，知识更替日新月异。希望、困惑、机遇、挑战，随时随地都有可能出现在每一个社会成员的生活之中。抓住机遇，寻求发展，迎接挑战，适应变化的制胜法宝就是学习——依靠自己学习、终生学习。

作为我国高等教育组成部分的自学考试，其职责就是在高等教育这个水平上倡导自学、鼓励自学、帮助自学、推动自学，为每一个自学者铺就成才之路。组织编写供读者学习的教材就是履行这个职责的重要环节。毫无疑问，这种教材应当适合自学，应当有利于学习者掌握、了解新知识、新信息，有利于学习者增强创新意识、培养实践能力、形成自学能力，也有利于学习者学以致用、解决实际工作中所遇到的问题。具有如此特点的书，我们虽然沿用了“教材”这个概念，但它与那种仅供教师讲、学生听，教师不讲、学生不懂，以“教”为中心的教科书相比，已经在内容安排、形式体例、行文风格等方面都大不相同了。希望读者对此有所了解，以便从一开始就树立起依靠自己学习的坚定信念，不断探索适合自己的学习方法，充分利用已有的知识基础和实际工作经验，最大限度地发挥自己的潜能，以达到学习的目标。

欢迎读者提出意见和建议。

祝每一位读者自学成功。

全国高等教育自学考试指导委员会

1999.5

编者的话

软件工程是高等教育自学考试计算机应用专业独立本科段自学考试计划中的专业课程。计算机应用专业的培养目标是在各个领域建立计算机应用系统，软件开发是建立计算机应用系统的重要环节。因此，软件工程是计算机应用专业的一门工程性课程，也是十分重要的一门专业课程。

本书主要讲述建造软件系统的方法、技术、流程、工具、规范等等。本教材的任务是使考生掌握软件工程的基本概念、基本原理、实用的开发方法和技术；了解软件工程各领域的发展动向；如何用工程化的方法开发软件项目，在开发过程中应遵循的流程、准则、标准和规范。软件工程是一门实践性很强的课程，它是各种开发经验的总结与提炼，在学习过程中不但应注重概念、原理、方法、技术的掌握，也应注重方法、技术的实际应用。

全书共分13章和一篇实践环节。第1章绪论，从总体上介绍软件工程的基本概念和内容；软件工程过程和生存期的基本概念和内容；软件开发的各种方法和生存周期模型。

第2章至第9章按照软件生存期的各个阶段，包括软件可行性研究与项目开发计划、软件需求分析、软件概要设计、详细设计、软件编码、软件测试和软件维护。详细介绍有关概念与软件工程方法，重点介绍结构化的分析和设计方法；介绍软件开发的增量模型。

第10章介绍目前软件开发方法的主流——面向对象的开发方法。本章全面介绍面向对象的基本思想、基本概念、基本原理，三种模型的基本概念和构造方法，面向对象的分析、设计与实现。

第11章至第13章讨论软件质量保证、软件工程管理技术和软件开发环境。软件工程管理是对软件生存周期一切活动的管理，尤其是对软件项目开发过程的管理，它对保证高质量的软件产品有着重要意义。

实践环节包括类型和实验大纲两部分，有三个课程实验。

本书“附录”附上可行性研究报告、项目开发计划、需求规格说明书、概要设计说明书、详细设计说明书、用户操作手册、测试计划、测试分析报告、开发进度月报的工作流程。

本书由全国高等教育自学考试委员会电子电工及信息类专业委员会组织编写。

本书第1、2、8、9、10章由邓良松教授编写，第3、4、5、6、7章由刘海岩副教授编写，第11、12、13章由陆丽娜教授编写，全书由陆丽娜教授统稿并担任主编。在本书成稿过程中，承蒙西安电子科技大学徐甲同教授、西安交通大学郑守洪教授、西北大学卞雷教授审阅全部书稿并提出许多宝贵意见，使编者受益不浅。上海交通大学陈敏逊教授、西安交通大学薛均义教授为文稿的组织与编写给予极大的支持与关心，仅借此一一表示深深的感谢。

编者

1999年8月26日于西安交通大学

目 录

软件工程

第 1 章 绪论	(1)
1.1 软件工程的产生	(1)
1.1.1 软件的特点	(1)
1.1.2 软件生产的发展	(1)
1.1.3 软件危机	(2)
1.1.4 软件工程	(3)
1.2 软件工程过程和软件生存周期	(5)
1.2.1 软件工程过程	(5)
1.2.2 软件生存周期	(6)
1.3 软件生存周期模型、方法和工具	(8)
1.3.1 软件生存周期模型	(8)
1.3.2 软件开发方法	(10)
1.3.3 软件开发工具	(11)
习题 1	(13)
第 2 章 软件可行性研究与项目开发计划	(14)
2.1 可行性研究	(14)
2.1.1 可行性研究的任务	(14)
2.1.2 可行性研究的具体步骤	(15)
2.1.3 可行性研究报告的主要内容	(16)
2.2 系统流程图	(17)
2.3 成本——效益分析	(19)
2.4 项目开发计划	(20)
习题 2	(20)
第 3 章 软件需求分析	(21)
3.1 需求分析的任务	(21)
3.1.1 需求分析的概念	(21)
3.1.2 需求分析的基本任务	(22)
3.1.3 需求规格说明书主要内容	(22)
3.2 结构化分析方法	(23)
3.2.1 自顶向下逐层分解的分析策略	(23)
3.2.2 描述工具	(24)

3.2.3	SA 分析步骤	(24)
3.3	数据流图 (DFD)	(25)
3.3.1	基本图形符号	(25)
3.3.2	画数据流图的步骤	(26)
3.3.3	实例——销售管理系统	(28)
3.4	数据字典 (DD)	(30)
3.4.1	数据字典的内容及格式	(30)
3.4.2	数据字典的实现	(32)
3.5	加工逻辑的描述	(33)
3.5.1	结构化语言	(33)
3.5.2	判定表	(34)
3.5.3	判定树	(35)
3.6	IDEF 方法	(36)
3.6.1	IDEF ₀ 的图形表示	(36)
3.6.2	建立功能模型的基本方法	(37)
3.6.3	IDEF ₀ 方法的特点	(38)
3.7	结构化分析方法小结	(39)
习题 3		(40)
第 4 章	软件概要设计	(41)
4.1	软件概要设计的基本任务	(41)
4.1.1	基本任务	(41)
4.1.2	软件概要设计说明书的主要内容	(43)
4.2	软件设计的基本原理	(43)
4.2.1	模块化	(43)
4.2.2	抽象	(44)
4.2.3	信息隐蔽	(44)
4.2.4	模块独立性	(45)
4.3	软件结构优化准则	(48)
4.3.1	软件结构图	(48)
4.3.2	软件结构设计优化准则	(49)
4.4	面向数据流的设计方法	(51)
4.4.1	数据流的类型	(51)
4.4.2	设计过程	(52)
4.4.3	变换分析设计	(52)
4.4.4	事务分析设计	(54)
4.4.5	综合型数据流图与分层数据流图映射成软件结构的设计	(55)
4.4.6	实例: 将 3.3.3 实例——销售管理系统的数据流图转化为软件结构	(56)
4.4.7	设计的后处理	(56)
4.5	基于 IDEF ₀ 图的设计方法	(58)
4.6	表示软件结构的另一种图形工具——HIPO 图	(58)

习题 4	(59)
第 5 章 软件详细设计	(61)
5.1 详细设计的基本任务	(61)
5.2 结构化程序设计方法	(61)
5.3 详细设计描述法	(62)
5.3.1 程序流程图	(63)
5.3.2 PAD 图	(63)
5.3.3 过程设计语言	(65)
5.4 Jackson 方法	(68)
5.4.1 概述	(68)
5.4.2 Jackson 结构图	(68)
5.4.3 JSP 设计步骤	(69)
5.4.4 Jackson 方法小结	(73)
习题 5	(74)
第 6 章 软件编码	(75)
6.1 程序设计语言的特性及选择	(75)
6.1.1 程序设计语言特性	(75)
6.1.2 程序设计语言的选择	(77)
6.2 程序设计风格	(79)
习题 6	(81)
第 7 章 软件测试	(82)
7.1 软件测试的目的及原则	(82)
7.1.1 软件测试的目的	(82)
7.1.2 软件测试的原则	(82)
7.2 测试方法	(83)
7.2.1 静态测试与动态测试	(83)
7.2.2 黑盒测试法与白盒测试法	(84)
7.3 测试用例的设计	(85)
7.3.1 白盒技术	(85)
7.3.2 黑盒技术	(90)
7.4 测试过程	(94)
7.4.1 软件测试过程中的信息	(94)
7.4.2 软件测试的步骤及与各开发阶段的关系	(94)
7.4.3 单元测试	(95)
7.4.4 集成测试	(96)
7.4.5 确认测试	(99)
7.5 调试	(100)
7.5.1 调试的目的	(100)
7.5.2 调试技术	(100)
习题 7	(102)

第 8 章 软件维护	(104)
8.1 软件维护的内容	(104)
8.2 维护的特点	(105)
8.2.1 非结构化维护和结构化维护	(105)
8.2.2 维护的困难性	(105)
8.2.3 软件维护的费用	(106)
8.3 维护任务的实施	(107)
8.3.1 维护的组织	(107)
8.3.2 维护的流程	(108)
8.3.3 维护技术	(109)
8.3.4 维护的副作用	(110)
8.4 软件可维护性	(111)
8.4.1 可维护性定义	(111)
8.4.2 可维护性的度量	(111)
8.4.3 提高可维护性的方法	(112)
习题 8	(115)
第 9 章 软件开发的增量模型	(116)
9.1 概述	(116)
9.1.1 瀑布模型的局限性	(116)
9.1.2 增量模型的基本思想	(117)
9.1.3 增量模型的分类	(117)
9.2 渐增模型	(118)
9.2.1 增量构造模型	(118)
9.2.2 演化提交模型	(119)
9.3 快速原型模型	(119)
9.3.1 基本思想	(119)
9.3.2 快速原型模型表示	(120)
9.3.3 原型开发过程	(121)
9.4 快速原型的开发技术和开发环境	(123)
9.5 增量模型的评价	(124)
习题 9	(125)
第 10 章 面向对象的方法	(126)
10.1 面向对象概述	(126)
10.1.1 传统开发方法存在的问题	(126)
10.1.2 面向对象的概念	(127)
10.1.3 面向对象的开发方法	(131)
10.2 面向对象的模型	(132)
10.2.1 对象模型	(132)
10.2.2 动态模型	(136)
10.2.3 功能模型	(138)

10.3 面向对象的分析	(140)
10.3.1 面向对象分析过程	(140)
10.3.2 建立对象模型	(141)
10.3.3 建立动态模型	(146)
10.3.4 建立功能建模	(148)
10.3.5 确定操作	(149)
10.4 面向对象的设计	(150)
10.4.1 面向对象设计的准则	(151)
10.4.2 面向对象设计的启发规则	(152)
10.4.3 系统设计	(154)
10.4.4 对象设计	(157)
10.5 面向对象的实现	(159)
10.5.1 程序设计语言	(159)
10.5.2 类的实现	(160)
10.5.3 应用系统的实现	(161)
10.5.4 面向对象测试	(161)
习题 10	(162)
第 11 章 软件质量与质量保证	(164)
11.1 概述	(164)
11.1.1 软件质量的定义	(164)
11.1.2 软件质量的度量评价	(164)
11.1.3 软件质量保证	(165)
11.2 质量度量模型	(167)
11.2.1 McCall 质量度量模型	(167)
11.2.2 ISO 的软件质量评价模型	(168)
11.3 软件复杂性	(169)
11.3.1 软件复杂性的基本概念	(169)
11.3.2 软件复杂性的度量方法	(169)
11.4 软件可靠性	(171)
11.4.1 软件可靠性定义	(171)
11.4.2 软件可靠性指标	(172)
11.4.3 软件可靠性模型	(172)
11.5 软件评审	(173)
11.5.1 设计质量的评审内容	(173)
11.5.2 程序质量的评审内容	(174)
11.6 软件容错技术	(175)
11.6.1 容错软件定义	(176)
11.6.2 容错的一般方法	(176)
11.6.3 容错软件的设计过程	(177)
习题 11	(178)

第 12 章 软件工程管理	(179)
12.1 软件工程管理概述	(179)
12.1.1 软件产品的特点	(179)
12.1.2 软件工程管理的重要性	(179)
12.1.3 软件工程管理的内容	(180)
12.2 软件项目计划	(181)
12.2.1 软件项目计划概念	(181)
12.2.2 软件项目计划内容	(182)
12.2.3 制定软件工程规范	(183)
12.2.4 软件开发成本估算	(183)
12.2.5 风险分析	(186)
12.2.6 软件项目进度安排	(186)
12.2.7 软件质量保证	(188)
12.3 软件配置管理	(189)
12.3.1 基线	(189)
12.3.2 软件配置项	(189)
12.3.3 版本控制	(190)
12.3.4 变更控制	(190)
12.4 软件工程标准化与软件文档	(191)
12.4.1 什么是软件工程标准化	(191)
12.4.2 软件工程标准化的意义	(192)
12.4.3 软件工程标准的层次	(192)
12.4.4 文档的作用与分类	(193)
习题 12	(194)
第 13 章 软件开发环境	(195)
13.1 软件开发环境	(195)
13.1.1 软件开发环境概论	(195)
13.1.2 软件开发环境的分类	(198)
13.2 软件工具的基本概念	(199)
13.2.1 什么是软件工具	(200)
13.2.2 当前软件工具发展的特点	(201)
13.2.3 软件工具的分类	(201)
13.3 计算机辅助软件工程 (CASE)	(202)
13.3.1 CASE 定义	(202)
13.3.2 CASE 分类	(202)
13.3.3 CASE 的集成	(204)
13.3.4 CASE 生存期	(205)
13.3.5 CASE 工作台	(206)
习题 13	(209)
实践环节	(211)

附录	(212)
参考文献	(218)

软件工程自学考试大纲

出版前言	(221)
一、课程性质与设置目的	(223)
二、课程内容与考核目标	(224)
第1章 绪论	(224)
第2章 软件可行性研究与项目开发计划	(225)
第3章 软件需求分析	(225)
第4章 软件概要设计	(227)
第5章 软件详细设计	(228)
第6章 软件编码	(228)
第7章 软件测试	(229)
第8章 软件维护	(230)
第9章 软件开发的增量模型	(230)
第10章 面向对象的方法	(231)
第11章 软件质量与质量保证	(233)
第12章 软件工程管理	(234)
第13章 软件开发环境	(235)
三、有关说明与实施要求	(237)
附录 题型举例	(239)
后记	(240)

第 1 章 绪 论

1.1 软件工程的产生

1.1.1 软件的特点

“软件”一词是 20 世纪 60 年代才出现的，它的定义是：计算机程序及其说明程序的各种文档。在该定义中，“程序”是计算任务的处理对象和处理规则的描述；“文档”是有关计算机程序功能、设计、编制、使用的文字或图形资料。软件与硬件一起构成完整的计算机系统，它们是相互依存、缺一不可的。软件是一种特殊的产品，它具有一些独特的特性。

(1) 软件是一种逻辑产品，它与物质产品有很大的区别。软件产品是看不见摸不着的，因而具有无形性，它是脑力劳动的结晶，它以程序和文档的形式出现，保存在计算机存储器的磁盘和光盘介质上，通过计算机的执行才能体现它的功能和作用。

(2) 软件产品的生产主要是研制，软件产品的成本主要体现在软件的开发和研制上，软件开发研制完成后，通过复制就产生了大量软件产品。

(3) 软件产品不会用坏，不存在磨损、消耗问题。

(4) 软件产品的生产主要是脑力劳动，还未完全摆脱手工开发方式，大部分产品是“定做”的。

(5) 软件费用不断增加，软件成本相当昂贵。软件的研制工作需要投入大量的、复杂的、高强度的脑力劳动，它的成本非常高。

1.1.2 软件生产的发展

自从第一台计算机诞生以后，就开始了软件的生产，到现在为止，经过了三个阶段。即程序设计时代，程序系统时代，软件工程时代。

1. 程序设计时代 (1946 年~1956 年)

这个阶段的生产方式是个体手工劳动，使用的工具是机器语言、汇编语言。开发方法是追求编程技巧，追求程序运行效率。因而使得程序难读、难懂、难修改。硬件特征是价格贵、存储容量小、运行可靠性差。软件特征是只有程序、程序设计概念，不重视程序设计方

法。

2. 程序系统时代 (1956 年~1968 年)

这阶段的生产方式是作坊式的小集团合作生产, 生产工具是高级语言, 开发方法仍旧靠个人技巧, 但开始提出结构化方法。硬件特征是速度、容量、工作可靠性有明显提高, 价格降低, 销售有爆炸性增长。软件特征是程序员数量猛增, 大量其他行业人员进入这个行业, 因为缺乏训练, 因而开发人员素质差, 这时已意识到软件开发的重要性, 但开发技术没有新的突破, 大量软件开发的需求已提出, 但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, 产生了尖锐的矛盾, 导致软件危机的产生。

3. 软件工程时代 (1968 年至现在)

这阶段的生产方式是工程化的生产, 使用数据库、开发工具、开发环境、网络、分布式、面向对象技术来开发软件。硬件特征是向超高速、大容量、微型化以及网络化方向发展。软件特征是开发技术有很大进步, 但是未能获得突破性进展, 软件价格不断上升, 没有完全摆脱软件危机。

1.1.3 软件危机

1. 软件危机的产生

软件发展第二阶段的末期, 由于计算机硬件技术的进步, 计算机运行速度、容量、可靠性有显著的提高, 生产成本显著下降, 这为计算机的广泛应用创造了条件。一些复杂的、大型的软件开发项目提出来了, 但是, 软件开发技术的进步一直未能满足发展的要求。在软件开发中遇到的问题找不到解决的办法, 使问题积累起来, 形成了尖锐的矛盾, 因而导致了软件危机。

2. 软件危机的表现

(1) 经费预算经常突破, 完成时间一再拖延。由于缺乏软件开发的经验和软件开发数据的积累, 使得开发工作的计划很难制定。主观盲目制定计划, 执行起来与实际情况有很大差距, 使得开发经费一再突破。由于对工作量估计不足, 对开发难度估计不足, 进度计划无法按时完成, 开发时间一再拖延。

(2) 开发的软件不能满足用户要求。开发初期对用户的要求了解不够明确, 未能得到明确表达。开发工作开始后, 软件人员和用户又未能及时交换意见, 使得一些问题不能及时解决, 导致开发的软件不能满足用户的要求, 因而开发失败。

(3) 开发的软件可维护性差。开发过程没有统一的、公认规范, 软件开发人员按各自的风格工作, 各行其是。开发过程无完整、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。

(4) 开发的软件可靠性差。由于在开发过程中, 没有确保软件质量的体系和措施, 在软件测试时, 又没有严格的、充分的、完全的测试, 提交给用户的软件质量差, 在运行中暴露

出大量的问题。这种不可靠的软件，轻者会影响系统正常工作，重者会发生事故，造成生命财产的重大损失。

3. 软件危机的原因

造成上述软件危机的原因是由于软件产品本身的特点以及开发软件的方式、方法、技术和人员引起的。

(1) 软件的规模越来越大，结构越来越复杂。随着计算机应用的日益广泛，需要开发的软件规模日益庞大，软件结构也日益复杂。1968年美国航空公司订票系统达到30万条指令；IBM 360 OS第16版达到100万条指令，花了5000个人/年；1973年美国阿波罗计划达到1000万条指令。这些庞大软件的功能非常复杂，体现在处理功能的多样性和运行环境的多样性。有人曾估计，软件设计与硬件设计相比，其逻辑量要多达10~100倍。对于这种大规模软件的复杂性，其调用关系、接口信息复杂，数据结构也复杂，这种复杂程度超过了人能接受的程度。

(2) 软件开发管理困难而复杂。由于软件规模大，结构复杂，又具有无形性，因此导致管理困难，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。软件生产是一种智力劳动，它是资金密集、人力密集的产业，大型软件投入人力多，周期长，费用上升很快。

(4) 软件开发技术落后。在60年代，人们注重一些计算机理论问题的研究，如编译原理、操作系统原理、数据库原理、人工智能原理、形式语言理论等，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，它们之间的差距越来越大。

(5) 生产方式落后。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，靠言传身教方式工作。

(6) 开发工具落后，生产率提高缓慢。软件开发工具过于原始，没有出现高效率的开发工具，因而软件生产率低下。在1960年~1980年这20年间，计算机硬件的生产由于采用计算机辅助设计、自动生产线等先进工具，使硬件生产率提高100万倍，而这20年间软件生产率只提高2倍，相差十分悬殊。

1.1.4 软件工程

由于上述原因导致了软件危机，为了克服软件危机，人们从其他产业的工程化生产得到启示，于是在1968年北大西洋公约组织的工作会议上首先提出“软件工程”的概念，提出要用工程化的思想来开发软件，从此，软件生产进入软件工程时代。

1. 软件工程定义

软件工程有多种定义，其中一种是：用科学知识和技术原理来定义、开发、维护软件的一门学科。

这个定义说明了软件工程是计算机科学中的一个分支，其主要思想是在软件生产中用工程化的方法代替传统手工方法。工程化的方法借用了传统的工程设计原理的基本思想，采用若

干科学的、现代化的方法技术来开发软件，这种工程化的思想贯穿到需求分析、设计、实现直到维护的整个过程。

2. 软件工程性质

软件工程是一门综合性的交叉学科，它涉及计算机科学、工程科学、管理科学、数学等领域。

计算机科学中的研究成果均可用于软件工程，但计算机科学着重于原理和理论，而软件工程着重于如何建造一个软件系统。

软件工程要用工程科学中的观点来进行费用估算、制定进度、制定计划和方案。

软件工程要用管理科学中的方法和原理进行软件生产的管理。

软件工程要用数学的方法建立软件开发中的各种模型和各种算法，如可靠性模型，说明用户需求的形式化模型等。

3. 软件工程目标

软件工程是一门工程性学科，目的是成功地建造一个大型软件系统，所谓成功是要达到以下几个目标：付出较低的开发成本；达到要求的软件功能；取得较好的软件性能；开发的软件易于移植；需要较低的维护费用；能按时完成开发任务，及时交付使用；开发的软件可靠性高。

4. 软件内容

软件工程研究的主要内容是软件开发技术和软件开发管理两个方面。在软件开发技术中，主要研究软件开发方法、软件开发过程、软件开发工具和环境。在软件开发管理中，主要是研究软件管理学、软件经济学、软件心理学等。

5. 软件工程面临的问题

摆在软件工程面前有许多需要解决的棘手问题，如软件费用、软件可靠性、软件可维护性、软件生产率和软件重用等。

(1) 软件费用

由于软件生产基本上仍处于手工状态，软件是知识高度密集的技术的综合产物，人力资源远远不能适应软件这种迅速增长的社会要求，所以软件费用上升的势头必然还将继续下去。

(2) 软件可靠性

软件可靠性是指软件系统能否在既定的环境条件下运行并实现所期望的结果。在软件开发中，通常要花费 40% 的代价进行测试和排错，就这样还不能保证以后不再发生错误，为了提高软件可靠性，就要付出足够的代价。

(3) 软件维护

统计数据表明：软件的维护费用占整个软件系统费用的 $\frac{2}{3}$ ，而软件开发费用只占整个软件系统费用的 $\frac{1}{3}$ 。之所以有如此大的花费，因为已经运行的软件还需排除隐含的错误，新增加的功能要加入进去，维护工作又是非常困难的，效率又是非常低下的。因此，如何提