

软件工程实用丛书

软件工程推进方法

罗积玉 李 超 等编著

电子科技大学出版社

图书在版编目 (CIP) 数据

软件工程推进方法 / 罗积玉等编著. —成都: 电子科技大学出版社, 2004. 9

ISBN 7-81094-686-2

.软... . 罗 . 软件工程 . TP311.5

中国版本图书馆 CIP 数据核字 (2004) 第 098376 号

软件工程实用丛书

软件工程推进方法

罗积玉 李超 等编著

出 版：电子科技大学出版社(成都建设北路二段四号 邮编：610054)

责任编辑：许宣伟

发 行：新华书店经销

印 刷：成都金龙印务有限责任公司

开 本：787mm × 1092mm 1/16 印张 8.5 字数 206 千字

版 次：2004 年 9 月第一版

印 次：2004 年 9 月第一次印刷

书 号：ISBN 7-81094-686-2/TP · 380

印 数：1—5000 册

定 价：20.00 元

《软件工程实用丛书》编辑委员会

- 主 编 李 超 （成都信息工程学院）
副主编 罗积玉 （四川四凯计算机软件有限公司）
野濑 義幸 （日本国（株）吴电子计算中心）
黎 明 （西南石油学院）
编 委 张 青 （四川四凯计算机软件有限公司）
朱 毅 （成都信息工程学院）
赵正文 （西南石油学院）
徐 徽 （日本国（株）吴电子计算中心）

前 言

软件开发不再是作坊式的软件开发模式，而必须按照软件工程的思想，走规模化、工业化的发展道路。在这样的背景之下，我们组织编写了《软件工程实用丛书》。

《软件工程的推进方法》一书作为该套丛书的开篇指南，主要阐述有关软件开发的基本概念以及软件工程的概要，主要包括软件与软件工程，软件工程的标准过程，软件开发模型及典型设计方法，软件的质量保证和个体软件过程（PSP）。

在编写风格上，我们从企业开发案例中提取了大量翔实的图表，使其对相关容易于理解和消化，配套习题（思考和练习）便于巩固所学知识。

全书共六章，其中第一章由罗积玉执笔，第三章由付常超执笔，第五章由张青执笔，第二、四、六章由杜斌执笔，最后由罗积玉、李超统稿。

在编写过程中，我们要感谢四川省四凯计算机软件有限公司、成都信息工程学院软件学院、日本国（株）吴电子计算中心（KCC）、西南石油学院软件学院的大力支持和参与，感谢四川软件行业协会的支持，特别是要感谢野濑义幸先生为我们写作提供了大量翔实的第一手材料。

本书以软件开发初学者、与 IT 相关专业的大学本、专科学生和从事 IT 职业的程序员为对象，因此可作为计算机及相关专业、软件工程专业各类层次的“软件工程引（导）论”课程的教材使用，亦可作为软件开发者、组织者和管理者的参考用书。

因水平有限，错误和不足之处在所难免，敬请广大读者批评指正。

编 者

2004 年 6 月

序

21 世纪，信息技术革命风起云涌，深刻地影响着世界经济、科技和社会的发展格局。顺应国际信息技术的飞跃发展，我国软件产业在国家一系列扶持政策的推动下，也逐渐进入国际竞争的轨道，过去作坊式的软件开发已无法适应我国信息产业的发展和国际竞争的需要。因此，规模化、标准化、工业化也是我国软件产业发展的必然趋势。

为了适应信息产业发展对人才的需求，我国高等教育加大了发展 IT 教育的力度，国家近几年相继兴办和扶持了一批软件学院和软件职业技术学院，并鼓励在运行模式、教育理念、课程体系及教材、教学方法等方面进行改革创新，确立了实用软件工程师的培养目标，在这样的指导思想和背景下，我们组织了富有教育经验的高等院校，多年从事软件业务开发和外包软件开发的著名企业，以及日本国专门从事软件开发的企业的学者、专家、各类软件工程师，在反复论证的基础上编写了这套《软件工程实用丛书》。本套丛书编写的指导思想明确，构思富有创意，有很强的实用性，是一部培养实用软件工程师的好教材。

本套丛书从实际应用出发，考虑到我国软件工程教育与发达国家相比，还有很大的差距，特别在注重实用性，跟踪新技术方面差距更大。同时考虑到我国在发展自己软件产业的同时，也要走软件外包的发展道路，软件工程教育要适合国际上软件开发的规范、基准和要求。

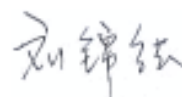
本套丛书研发的组织架构中，有国内多年从事软件开发、软件外包的软件企业，有极富软件教育经验的高等院校，还有日本的软件开发企业。从作者的构成上看，有软件企业的各类软件工程师、项目经理、总经理，有高校的专家、学者，共同构成了强大的编写队伍。参与研发的单位和人员，深入研究、探讨和分析，集思广益，为保证本套丛书的实用性、技术的前沿性和先进性狠下了一番功夫。本套丛书结合了软件企业的应用、人才需求的实际，结合了我国高校软件人才培养的目标，还结合了国际上软件开发的软件工程技术 and 规范，以软件工程为中心构架本套丛书。本套丛书分为《软件工程推进方法》、《软件制造工程》、《软件设计工程》、《软件工程项目管理》、《软件工程师》五个主要部分，形成该套丛书的总体框架。然后，作者根据自己在软件企业多年的开发实

践和经验，特别是从事国际软件开发的实践和经验，收集整理所需要的相关素材，还收集了日本软件开发的工程化技术资料以及和软件开发完整的案例，在此基础上，草拟了五本书的写作大纲。再经过讨论，广泛征集意见，几易其稿，确定了本套丛书的编写内容。最后由软件工程师和从事软件专业教育的教师拟稿、定稿。

本套丛书具有系统性、专业性和实用性的特点，围绕软件开发的标准过程和软件制造、设计、管理的三个环节，系统阐述了在软件开发不同阶段软件工程实施的内容、推动软件开发的方法以及在开发过程中软件工程师应具备的职业素质。五个部分既独立成书又相互衔接，思路清晰，逻辑连贯；本套丛书针对软件工程专业特点，既有专业理论和操作原理的阐释，更注重实际案例的分析，还有软件行业专业规范、基准的论述，形成了本套丛书很强的专业化特色。实用性更是编著者极力追求的原则。本套丛书力图改变传统教材偏重学术理论体系的完整性，而较少考虑内容的实用性的弊病，以实用性为中心，在阐述基本概念和基本理论的同时，注重实际案例分析和操作，并重视图形表现，将图表和文字放在同等重要的位置。

应当指出，对软件工程教材系统性、专业性和实用性的探讨，本套丛书做了一次可喜的尝试。但软件工程课程及教材改革必须跟上国际软件工程发展的步伐，随着软件工程的发展而不断改进和更新。

四川省软件行业协会会长
电子科技大学教授



2004年6月

目 录

第一章 软件与软件工程.....	1
1.1 软件	1
1.1.1 计算机和高级程序语言.....	1
1.1.2 软件的概念.....	2
1.1.3 软件与我们的关系.....	3
1.1.4 什么是好的软件.....	3
1.2 软件分类	4
1.3 软件工程	6
1.3.1 从软件危机到倡导软件工程.....	6
1.3.2 软件工程的定义.....	7
1.4 软件生命周期	9
1.5 软件开发与软件开发方法	10
1.6 软件开发项目管理	12
1.7 我国软件产业的现状	14
1.8 软件人才资源	16
练习题	18
第二章 软件开发模型.....	20
2.1 软件开发模型与软件工程	20
2.2 瀑布式模型	21
2.2.1 瀑布模型概念.....	21
2.2.2 瀑布模型特点.....	23
2.3 演化模型	23
2.3.1 演化模型介绍.....	24
2.3.2 演化开发模型特点.....	24
2.4 螺旋模型	25
2.4.1 螺旋模型介绍.....	25
2.4.2 螺旋模型的特点.....	26
2.5 XP 开发模型	27
2.5.1 XP 开发模型概要.....	27

2.5.2	XP 开发模型的特点.....	29
	练习题	29
第三章	软件开发的 标准过程	30
3.1	软件开发的 标准过程	30
3.2	分析工程	31
3.2.1	需求分析的意义.....	31
3.2.2	结构化分析的思考方法.....	31
3.2.3	面向对象分析的思考方法.....	33
3.2.4	系统分析作业过程.....	34
3.3	设计工程	37
3.3.1	设计工程的思考方法.....	37
3.3.2	从外部设计（概要设计）到内部设计（详细设计）	38
3.3.3	设计工程的输入和输出.....	42
3.4	制造工程	42
3.4.1	制造工程概要.....	42
3.4.2	制造工程作业内容.....	43
3.5	测试工程	46
3.5.1	测试工程的意义.....	46
3.5.2	测试工程与生产性.....	47
3.5.3	组合测试.....	48
3.5.4	综合测试.....	50
3.5.5	应用测试.....	50
3.5.6	测试的实施.....	51
3.6	系统安装	51
3.6.1	实机测试.....	51
3.6.2	移植安装.....	52
	练习题	52
第四章	软件开发方法.....	53
4.1	结构化方法	53
4.1.1	结构化分析.....	55
4.1.2	结构化设计.....	59
4.1.3	结构化编程.....	67
4.2	面向对象的开发	67

4.2.1 面向对象的基本概念.....	68
4.2.2 面向对象的分析 (OOA : Object-Oriented Analysis)	69
4.2.3 面向对象的设计 (OOD : Object-Oriented Design)	74
4.2.4 面向对象的编程.....	79
练习题	79
第五章 软件的质量保证.....	81
5.1 软件质量	81
5.1.1 软件的质量特性.....	81
5.1.2 软件质量的定义.....	83
5.1.3 不易发现的软件缺陷.....	84
5.2 软件质量的思考方法	85
5.2.1 致力于实现符合客户要求的质量.....	85
5.2.2 缺陷的防止和发现.....	86
5.2.3 软件工程标准化.....	87
5.3 设计检查与制造、测试工程的质量控制	88
5.3.1 防止上流工程的缺陷.....	88
5.3.2 设计阶段的质量控制.....	88
5.3.3 编程阶段的质量控制.....	91
5.3.4 测试阶段的质量控制.....	92
5.4 软件质量管理	94
5.4.1 质量保证.....	94
5.4.2 质量策划.....	94
5.4.3 质量控制.....	95
5.4.4 超前的质量管理与管理者的职责.....	95
5.4.5 质量保证与项目总结.....	96
5.5 软件过程改善方法——软件过程能力成熟度模型(CMM)	97
5.5.1 CMM 的结构.....	97
5.5.2 CMM 成熟等级的定义.....	99
5.6 软件质量管理方法——质量管理模型 (ISO9000)	101
5.6.1 八项质量管理原则.....	101
5.6.2 过程管理方法.....	102
5.6.3 实施质量管理模型的标准步骤.....	103
5.6.4 依据 ISO9001 质量管理模型建立的质量管理体系.....	103
练习题	105

第六章 个体软件过程 PSP	106
6.1 个体软件过程 (PSP) 与软件质量	106
6.2 个体计划过程	110
6.2.1 时间管理	110
6.2.2 了解如何使用时间	111
6.2.3 制订时间安排表	112
6.2.4 对照时间安排表跟踪时间使用方式	117
6.2.5 总结时间安排计划	118
6.3 个体缺陷管理过程	120
6.3.1 缺陷记录	120
6.3.2 缺陷查找	122
练习题	124
参考文献	125

第一章 软件与软件工程

自 1946 年美国宾夕法尼亚大学研制出第一台电子计算机 ENIAC (Electronic Numerical Integrator And Computer) 以来, 计算机软件的开发一直是制约计算机广泛应用的瓶颈, 提高计算机软件的质量和生产率一直是人们致力于解决的重大课题。随着软件开发规模的扩大和软件开发费用的不断增加, 软件开发的难度越来越大, 开发技术、生产方式远远不能适应大规模软件开发项目的需要, 由此产生了软件危机。为了克服软件危机, 人们从其他产业的工程化生产中得到启示, 在 1968 年北大西洋公约组织的工作会议上首先提出了“软件工程”的概念, 即是用工程化的思想开发软件, 从此软件生产进入了软件工程化的时代, 从而导致利用软件工程的思想和方法对软件开发项目进行管理。

1.1 软件

1.1.1 计算机和高级程序语言

世界上第一台数字电子计算机 ENIAC 诞生, 标志着电子数字计算机时代的开始, 它是现代计算机的雏形。但真正嵌入计算机程序的计算机是 1949 年英国剑桥大学的 M.V.Wilkes 开发的 EDSAC (Electronic Delay Storage Automatic Calculator) 电子延迟存储自动计算机, 它是根据著名数学家冯·诺伊曼 (Johan Ludwig Von.Neumann) 倡导的诺伊曼结构程序的思想而设计、制造的。它的基本思想是把编制的程序蓄积在存储器中, 在存储器中的程序和数据区域没有分开, 混在一起, 从头到尾解释执行, 处理的顺序是按预先编制好的程序顺序执行, 这就是最先使用的低水准的执行程序。

最初的程序都是机器语言, 随着计算机需求量的扩大, 处理效率和处理方法的要求提高, 计算机硬件的性能提高, 程序用的存储器的存储容量扩大, 这样产生了汇编语言。从 1954 年开始产生了高级语言, 首先是用于科学计算的 FORTRAN, 1959 年又发表了用于事务处理的有代表性的高级语言 Cobol, 随后 1960 年发布了 USP, ALGOL 语言, 1965 年发表了 Basic, PL/I, 1970 年发布了 Pascal 语言, 1972 年发布了 C 语言, 1980 年发布了 Smalltalk. Ada 语言, 1983 年发布了 C++ 语言, 1993 年发布了 Java 语言等等, 各种各样的高级语言的发布为计算机应用开辟了广阔的前景。

机器语言以及汇编语言都是要依赖于不同的计算机而编写的不同的机器语言或汇编语言, 这两种依赖于机器的语言很难掌握, 程序作成的效率很低, 这样便限制了计算机应用领域的扩大。而高级语言不依赖于机种, 高级语言编写的程序是通过预先注入机器内存中的翻译程序翻译为可执行的机器语言予以执行, 同时还预先将许多诸如函数计算、四则运算等专用或者通用模块融入内存中与高级语言编写的程序链接, 成为可执行的程序。高级语言易学习, 易掌握, 通用性强, 处理效率高, 便于纠错, 应用领域广泛, 为扩大计

算机应用，推进信息化的发展，特别是推进计算机的发展开辟了广阔的前景。

1.1.2 软件的概念

1. 软件的含义

软件一词源于 20 世纪 60 年代，英文 software 一词是由 soft 和 ware 两字组合而成。软件在不同阶段有不同的认识，在计算机发展初期，硬件的设计和生是占主要地位，而软件仅处于从属地位，那时所谓软件仅仅是程序，那时软件的生产方式是个体手工方式，设计和程序的编制过程在一个人的头脑中完成，程序的质量也完全取决于个人的编程技巧，程序的维护基本上也只能由编制者本人进行，甚至于随着时间的推移，程序编制者本人也无法进行维护。

随着社会对计算机应用需求的扩大，大型开发系统的设计和生生产的工作量大幅度增加，传统的软件生产方式已经不能适应发展的需要，因此把工程学的基本原理和方法引入到软件开发和管理中，形成软件工程，软件工程将软件生产分成几个阶段，每个阶段都有严格的管理、设计说明书之类的文档资料，质量检测的资料，研制软件设计和生的方法、工具，以及共同遵循的规范、基准等，这时的软件含义就是所开发的程序加上相应的文档资料。

因此，软件是计算机系统中与硬件相互依存不可缺少的部分，它包括程序及其相关文档资料。程序是计算机任务的处理对象和处理规则的描述；文档是为了执行业务系统、理解程序所需要的阐述性资料。

2. 软件的特点

软件是计算机系统中与硬件相互依存、不可缺少的部分，软件相对于硬件进行比较具有如下特点：

(1) 软件与硬件的表现形式不一样。软件是一种逻辑实体，它具有抽象性，人们可以把它记录在纸面上，保存在计算机的存储器内部，也可以保存在存储介质上，如磁盘、磁带、光盘等，但无法看到软件本身的形态，必须通过分析、思考，一直到程序在计算机上运行得到结果才能体现它的功能、性能及其他特性。

(2) 生产方式与硬件不同。软件是人们通过智力活动，按照软件工程思想，把知识和技术转化成信息产品，软件的质量控制贯穿在软件开发的过程中。软件一旦完成，所形成的产品可以大量复制，因此软件产品的知识产权保护是需要解决的重大问题。

(3) 维护不同，与硬件相比，在软件的运行和使用期间，没有硬件那样的机械磨损、老化问题。但软件存在退化问题，在软件生存周期中，应用需求的变更、管理模式的变化、数据存储量的加大、硬件环境的变化等多方面原因都将导致软件效率降低，软件失效，使得软件发生退化。

(4) 软件的开发和运行对计算机系统有着不同程度的依赖性，对计算机系统的配置，对支撑软件运行的操作系统都有不同程度的依赖性，因此在软件的安装、移植方面必须考虑依存性问题。

(5) 软件产品包括有通用的和定制的，而且大多数为定制产品。由于软件是人类社会事务的反映，具有一定的复杂性，包括社会事务反映的复杂性和开发技术、程序逻辑

结构的复杂性，因此软件的成本相当高，软件的开发费用远远大于硬件的费用，特别是大型软件开发需要数千人·年。由于大型软件开发具有复杂性，耗费的人力、物力资源庞大，因此软件的可靠性、安全性将成为软件开发的重要技术指标。在现实生活中，也有耗费大量人力、物力而未取得成功的软件，这是我们必须高度重视的问题。

1.1.3 软件与我们的关系

近年来计算机已经进入社会生活的各个方面，成为不可缺少的东西，而计算机软件作为商品同样在家庭、企业、社会、政府、公共场所等广泛流通，一般人最为熟悉的的就是游戏软件，它已渗透到几乎每个有家用电脑的家庭。在我们身边常见的软件有：

(1) 在公共场所中常见的图书馆的图书检索系统、旅游指南、道路交通信息查询系统、宾馆预约系统、航空订票系统等；

(2) 业务上使用的应用系统，如销售管理系统、财务管理系统、生产管理系统、银行的柜员机、ERP 系统等；

(3) 家庭、个人常用的软件有游戏软件、文字处理软件、图形编辑软件等；

(4) 嵌入电子产品中的软件有手机、数字显示的钟表、电视机、录相机、VCD、汽车装载的控制系统等中嵌入的软件等；

(5) 使计算机系统运行的基本软件(操作系统)，如 OS、Windows、计算机开发语言，Oracle 数据库系统等；

随着计算机的普及和应用的扩大，电子产品的智能化发展，以及社会、科学的进步，软件已经渗透到社会生活的各个领域，深入到家庭，成为人们不可缺少的东西。

1.1.4 什么是好的软件

前面我们已经概述了软件的基本概念，下一步我们说明什么是好的软件。好的软件从不同的视点来看有不同的评价基准，从软件开发者来看，好的软件就是硬件资源的充分使用，软件作为资源而容易得到保存；顾客和开发者共通的评价标准是：

(1) 质量好(客户的需求充分满足，没有错误)

所谓质量好，就是能正确、充分地反映客户的需求，潜在的错误很少。质量特性是评价软件好坏最重要的因素。

(2) 开发费用低

在做软件开发时，所发生的开发费用不要超过计划费用，而且客户所要求的功能也能得到满足。当然开发费用与开发周期有密切关系，缩短开发周期也是降低开发费用最重要的一个因素。

(3) 开发周期短

规模比较大的软件，其开发周期比较长，容易产生危险，导致失败，同时较长的开发周期也使开发费用难以控制，因此严格按估计周期进行开发，这也是评价好的软件的基准。近年来，随着硬件技术的提高，软件性能的改善，缩短开发周期是容易实现的。

(4) 使用方便、操作容易

软件面向的使用对象不是计算机专家，而是广大从事具体业务工作的人员，因此使用方便、操作容易是非常重要的，这也是评价软件好坏的重要标准。

(5) 安全性好

由于硬件的进步、网络的发达，一些具有很高价值的信息将会联结到多个终端访问，大量的数据也需要存储。这样数据的丢失、访问、修改，以及由于故障造成损失等多方面因素，都需要可靠的软件进行数据管理和访问，因此软件的安全性也是评价好的软件的重要基准。

1.2 软件分类

软件分类有各种各样的方法，由于人们与软件的关系各不相同，关于软件分类的侧重点也不相同，所以很难给出一个科学的、统一的、严格的分类标准，但是对软件的类型进行必要的划分，根据不同类型的工程对象采用不同的开发和维护方法是很有价值的。

下面我们从不同的角度对软件进行分类。

1. 按规模大小划分

按规模划分的着眼点是软件使用人数、使用者的地理范围、开发者小组的规模、程序规模大小、系统接续的机器数量等。往往一个着眼因素的变化也将影响其他因素的变化。当然规模的大小也没有严格划分的界线，是相对而言。举例如下：

(1) 大规模软件：如宇宙开发、大型的航空管理系统、交通管理系统等；

(2) 中规模软件：如生产管理系统、财务管理系统、OA 办公自动化系统、ERP 系统等；

(3) 小规模软件：工资计算等。

2. 按层次划分

从与计算机硬件相近的顺序来看有基本软件（操作系统或运行支撑软件），进行开发、维护的支持软件（软件工具），中间软件（中间件）以及应用软件。其层次构造如图 1.1 所示。

软件的层次构造随着硬件的进化、软件规模的扩大、应用领域的拓展、开发效率的提高、通用性的追求而发生变化。

从图 1.1 中看，在硬件和基本软件之间的存储器（ROM）是用于存储常驻内存的软件，在第 2、第 3 阶段，它是把基本软件和应用软件并在一起注入内存存储器中，在其他情况，可以认为它是硬件的一部分。

(1) 基本软件有系统软件和支撑软件。系统软件是指能与计算机硬件配合，使计算机系统各个部件、相关软件和数据协调、高效地工作的软件。系统软件通常有操作系统、语言处理程序、服务性程序、数据库管理系统、计算机网络软件五类。

操作系统是管理计算机各种资源，自动调度用户作业程序，处理各种中断的软件。由操作系统管理的资源通常有硬件、软件和数据信息。

语言处理程序是将利用高级语言如 Basic、Cobol、C++ 等编写的源程序翻译成目标程序，不同语言的源程序对应不同的语言处理程序。

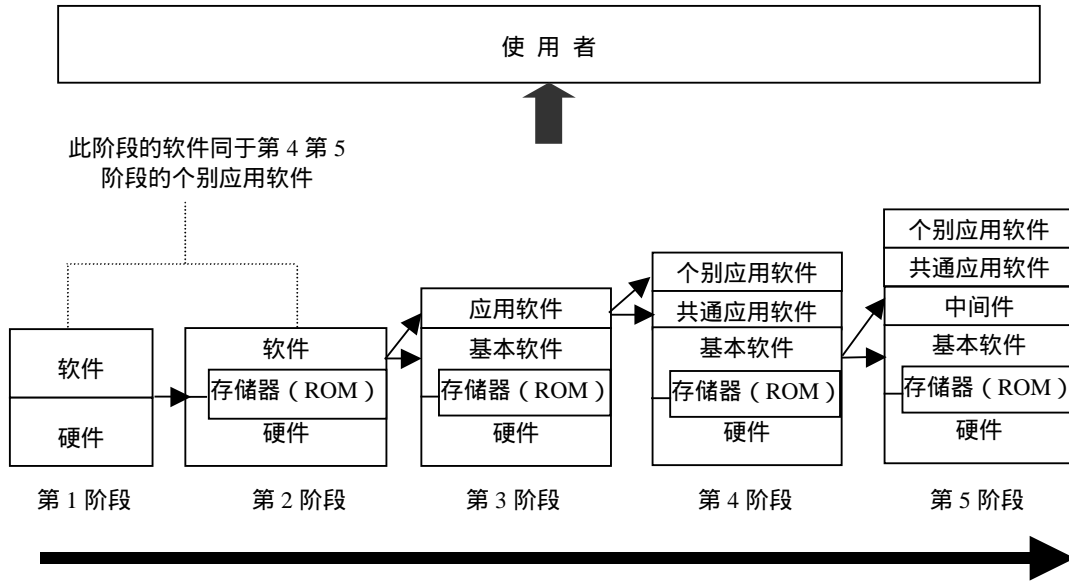


图 1.1 软件层次构造的变化

服务性程序亦称实用程序，是一种辅助计算机工作的程序，用于程序的装入、连接、编辑、调试的程序，用于诊断故障、纠错、监督等服务功能的程序。

数据库管理系统又称为数据库管理软件，用来管理系统中的文件，实现检索、数据共享等多种功能。

计算机网络软件是为计算机网络配置的系统软件，它负责对网络资源进行管理、实现相互之间的通信。计算机网络软件包括网络操作系统和数据处理程序。前者用于协调网络中各机器的操作系统实现网络资源管理，后者用于网络通信，实现网络操作。

支撑软件是协助用户开发软件的工具软件。

(2) 中间件是基本软件和应用软件之间的软件，它是把各种各样的应用软件的共通功能部分抽出来作为中间件，中间件使用得好，可以大大地减轻应用软件开发的负担。中间件例子有 DBMS、通信控制系统、EUC 工具、GVI 控制等。中间件是信息系统建设的基础设施，复杂程度高，技术环节多，投资规模大，目前国际 IT 巨头如 IBM、ORACLE、BEA、Microsoft 等都制定了相应战略，制定有关标准，投入巨资进行产品研发工作。

(3) 应用软件分为共通应用软件和个别应用软件，共通应用软件一般都作为软件商品销售，如结构分析的 CAD 软件，统计分析的 SAS 软件，科学计算程序库，人工智能等。而个别应用软件通常为定制软件，按照使用单位管理的需要而进行设计、编制的软件，如交通管制系统、生产管理系统、销售管理系统、情报检索系统等。应用软件也可按应用领域分类，如：

按行业领域分类：制造业、流通业、金融业、保险业、政府办公系统应用软件等；

按业务领域分类：人事、财务、销售、生产管理、物流管理应用软件等；

按功能分类：事务处理、科学计算、控制软件等；

按用途分类：业务用、家庭用、个人用软件等。

1.3 软件工程

随着计算机系统的发展和计算机应用的普及深入，对计算机软件的需求越来越大，计算机软件的开发规模也越来越大。然而计算机软件的开发成本却不断上升，质量保证也不尽人意，生产率低下，这些多方面的原因，已使软件的发展成为限制计算机系统发展和计算机应用扩大的关键因素。

1.3.1 从软件危机到倡导软件工程

1968 年 NATO (北大西洋条约机构) 的科学委员会主持召开的学术会议上第一次提出了“软件危机”，并且提出了解决“软件危机”的“软件工程”概念，但是这次会上没有提出“软件工程”的具体内容。一直到 1975 年软件工程国际会议 (ICSE: International Conference on Software Engineering) 前后才充实了软件工程的内容。

软件危机的产生以及软件危机意识的形成主要是在 20 世纪 70 年代以美国和前苏联为代表的军事、航空宇宙领域计算机系统开始进行大规模软件开发，以 IBM 公司为代表的计算机 OS/360 系统开发规模非常庞大，项目规模达到 5000 人年，结果并不尽人意。人们已经意识到软件系统规模的扩大需要投入大量的人力和物力，但系统的质量却难以保证，软件开发所需要的高成本和产品的低质量间的矛盾越来越突出，软件开发出现了危机，由此提出了“软件危机”的概念，如图 1.2 所示。

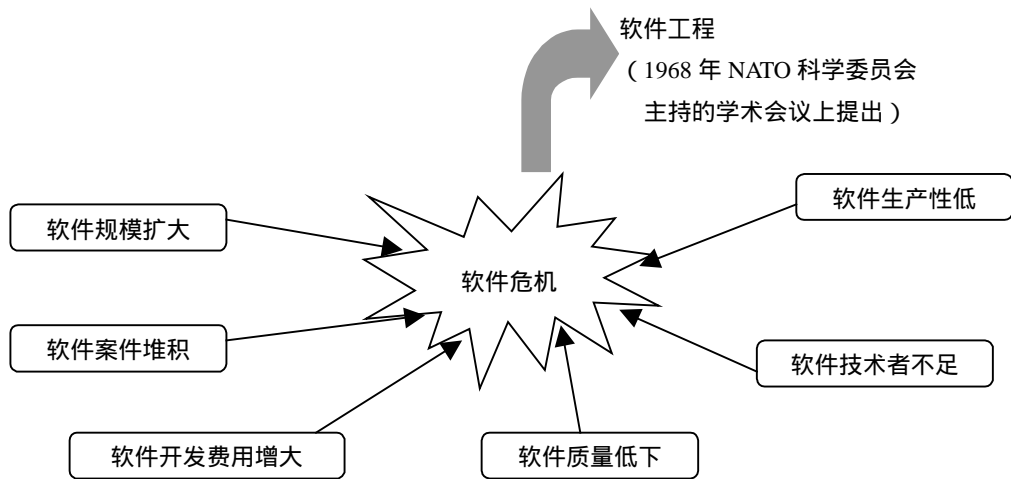


图 1.2 从软件危机到倡导软件工程

由于软件需要解决的课题增加，规模逐渐扩大，需要从事软件开发的技术人员增多，使得工程管理复杂化，生产性低，质量低下；同时软件案件堆积，为处理这些案件，软件开发费用增大，这种恶性循环，致使软件危机加剧。软件危机积聚的历史如表 1.1 所示。

表 1.1 软件危机及与软件有关的课题

时 期	与软件有关的课题	
20 世纪 70 年代	规模问题	提出了大规模软件开发的解决方法和开发技术
80 年代	量的问题	由于开发领域迅速扩大，开发项目数量增加，信息处理技术人员不足，软件问题案件堆积，软件开发人员的需求扩大成为主要问题
90 年代前半期	质的问题	<ul style="list-style-type: none"> · 重点解决软件故障问题，使得软件质量提高 · 解决终端用户接口问题，形成友好使用界面
90 年代后半期	接口问题	主要解决数据和程序等方面的互换性、接续性

1.3.2 软件工程的观念

软件工程学是一门指导软件开发和软件维护的工程学科，是为了经济地获得能够在实际机器上有效运行的可靠软件而建立和使用的一系列工程化原则。软件工程的原则一是排除人的个性，而归结为统一的学问体系，无论你学什么专业，有什么个性，都要按相同的理论和方法进行软件开发，其目的在于追求开发效率，提高生产性，进而实现生产自动化。另一个原则是开发的步骤如同机械开发步骤一样，不能随意改变开发的步骤。

图 1.3 列出了机械开发步骤与软件开发步骤的比较。

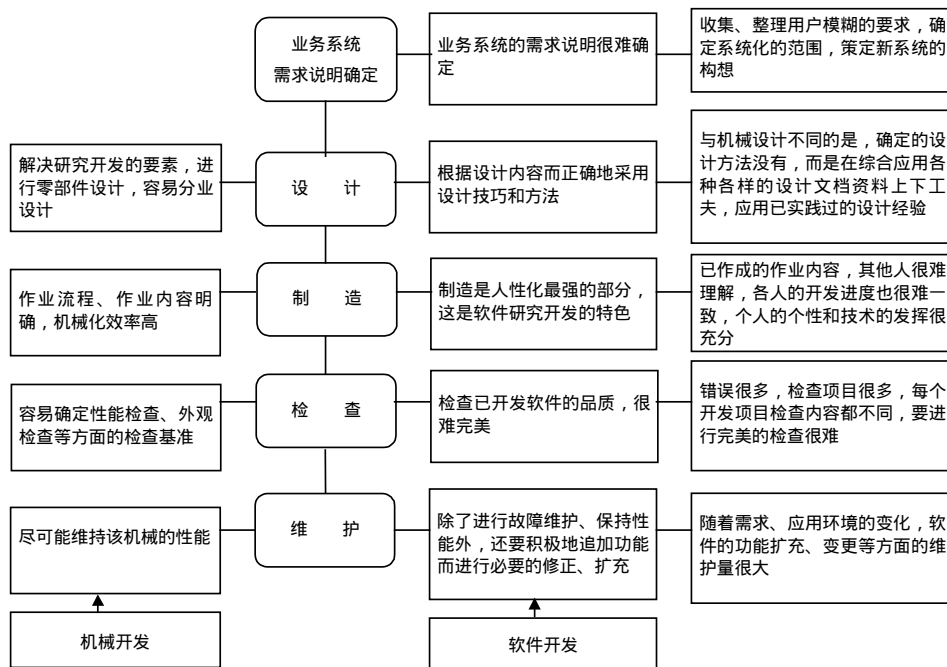


图 1.3 机械开发步骤和软件开发步骤的比较