

技术参考资料之106

軟件工程手冊

General Electric Company 著

杨绍祺 等译

科海培训中心

一九八七.八

编辑：科海培训中心教材部
发行：科海培训中心资料组
地址：北京2725信箱 科海培训中心
资料组

(北京海淀路332路黄庄站旁)

印刷：河北省蔚县印刷厂

译者的话

二十余年来,计算机系统发生了一系列重大的变化,这些变化包括:1、应用范围日益扩大。二十年前,主要的应用领域是科学计算,现在呢,从管理决策到工业控制,从财政金融到日常办公事务,计算机无所不在;2、研制成本比例迅速改变。二十年前,计算机系统的成本几乎全由计算机硬件的成本决定,计算机软件的成本仅占很少一部分。然而,随着微电子技术的发展,硬件成本垂直下降,甚至发生了五年内计算机硬件的成本下降两个乃至三个数量级的事。计算机硬件成本的下降趋势似乎还将继续下去。与此形成对比的是,计算机软件的成本却持续上升,其主要原因是计算机软件供不应求。一方面,计算机应用范围的日益扩大,需要的软件种类和数量与日俱增。另一方面,计算机软件的研制几乎停留在个体手工劳动的水平上,因而成本昂贵,以至造成今天这种计算机软件成本在整个计算机系统成本中占第一位的状况。

为改变这种情况,推动计算机科学的发展,人们从各个角度研究了计算机软件的研制与生产问题,并取得了丰硕成果,形成了一门新学科——软件工程。

由General Electric Company撰写的SOFTWARE ENGINEERING HANDBOOK是众多丰硕成果中最新(1986年发表)、适用范围最广、内容最全最适用的一个。我们将它翻译成中文,奉献给从事计算机软件研制的朋友们,以促进我国计算机软件事业的发展。

本书在对软件工程做了扼要的介绍之后,着重讨论了下列软件工程步骤:

1. 软件开发计划:项目范围,资源预算和建设工期估计;
2. 软件需求分析:详细叙述功能与性能要求,设计约束条件和验证测试标准;
3. 软件设计:用一套科学的方法,把软件要求转换为:①软件结构;②软件的详细的实现过程;
4. 结构编程:使用一种程序设计语言,把设计结果转换为功能上等价的程序;
5. 软件测试:采用一系列的步骤,试图找出程序模块内隐藏的错误,把程序模块组装成软件包,最后,按软件需求说明书上的要求,验证软件;
6. 软件配置维护:把上述步骤和适当的控制步骤用来管理已交付使用的软件。

此外,在附录C中按不同的课题领域,分类列出了软件工程方面的主要文献,并一一加了评注。因此,本书不仅为我们掌握软件工程提供了一条捷径,而且,也是进一步研究软件工程的指南。

由于许多术语尚未统一,故译文中大部分遵照较通用的译法。原文中有一些错误,我们做了订正。限于水平和时间仓促,译文中肯定有错误与缺点,望读者批评指正。

一九八七年七月于北京

目 录

译者的话

第一章 本书简介	(1)
1.1 问题的提出	(1)
1.1.1 软件开发	(1)
1.1.2 软件检测	(2)
1.2 解决方法	(2)
1.2.1 软件生命期	(2)
1.2.2 工程方法	(2)
1.2.3 软件工程	(2)
1.3 本书中引进的主要概念	(3)
1.3.1 生命期	(3)
1.3.2 软件项目估算	(3)
1.3.3 需求分析	(4)
1.3.4 软件设计	(4)
1.3.5 结构编程	(4)
1.3.6 测试步骤和方法	(4)
1.3.7 小型软件项目的软件工程	(4)
1.3.8 软件维护	(4)
1.3.9 软件配置管理	(5)
1.3.10 文件和检查	(5)
1.4 本书的目的	(5)
1.4.1 经理指南	(5)
1.4.2 技术人员指南	(6)
1.5 本书的内容	(6)
1.5.1 软件工程方法与步骤库	(6)
1.5.2 文件格式库	(7)
1.5.3 文献目录和一般的信息库	(7)
1.6 如何使用本书	(7)
1.6.1 课题与读者关联矩阵	(7)
1.6.2 建议经理阅读的内容	(7)
1.6.3 为技术人员推荐的内容	(7)
1.7 结语	(9)
第二章 软件工程概述	(10)
2.1 定义阶段	(11)

2.1.1	系统需求分析	(11)
2.1.2	初步的软件计划	(12)
2.1.3	软件需求	(12)
2.2	开发阶段	(13)
2.2.1	初步设计	(13)
2.2.2	详细设计	(16)
2.2.3	编写程序和程序模块测试	(19)
2.2.4	组装测试	(19)
2.2.5	验证测试	(20)
2.2.6	系统测试	(21)
2.3	维护阶段	(21)
2.3.1	软件监察	(21)
2.3.2	维护	(21)
2.3.3	软件配置管理	(22)
2.4	软件开发组织	(23)
2.5	结语	(24)
第三章	软件开发计划	(25)
3.1	系统定义	(25)
3.2	初步的软件开发计划	(26)
3.2.1	软件功能范围	(26)
3.2.2	资源需求	(27)
3.3	成本估算	(29)
3.3.1	软件的种类	(29)
3.3.2	软件生产数据	(29)
3.4	成本估算技术	(30)
3.4.1	估算源程序的行数	(31)
3.4.2	任务成本技术	(32)
3.4.3	自动估算成本	(35)
3.4.4	结果比较	(35)
3.5	做买软件的决定	(35)
3.5.1	购买软件的例子	(36)
3.5.2	软件的选择	(37)
3.6	其他成本	(37)
3.6.1	计算机成本费	(37)
3.6.2	差旅费	(39)
3.6.3	消耗品和其他硬件费	(39)
3.7	影响软件成本的因素	(39)
3.8	计划安排	(41)
3.9	检查软件计划	(42)

3.10 结论.....	(42)
第四章 软件需求分析	(43)
4.1 系统需求.....	(44)
4.2 需求子任务.....	(45)
4.2.1 研究软件开发计划.....	(46)
4.2.2 确定做分析时的联络范围.....	(46)
4.2.3 构造一个数据流模式.....	(46)
4.2.4 确定功能细节与界面.....	(47)
4.2.5 确定设计约束条件.....	(47)
4.2.6 规定验证测试标准.....	(47)
4.2.7 编写初步的用户手册.....	(48)
4.2.8 检查软件需求说明书.....	(48)
4.2.9 检查软件开发计划.....	(48)
4.3 分析员.....	(49)
4.4 数据流程图.....	(49)
4.5 软件需求说明书.....	(52)
4.6 需求分析的自动工具.....	(52)
4.7 结语.....	(54)
第五章 软件设计	(55)
5.1 设计过程.....	(55)
5.2 初步设计.....	(56)
5.3 软件设计概念.....	(56)
5.3.1 结构与过程.....	(57)
5.3.2 隶属与“管辖”关系.....	(57)
5.3.3 输入与输出特点.....	(58)
5.3.4 外联度.....	(58)
5.3.5 内联度.....	(59)
5.3.6 程序模块数.....	(59)
5.3.7 结构特点.....	(60)
5.3.8 影响域与控制域.....	(60)
5.3.9 结语.....	(61)
5.4 结构设计.....	(61)
5.4.1 系统数据流.....	(61)
5.4.2 确定信息流边界.....	(63)
5.4.3 转换中心分析.....	(63)
5.4.4 事务中心分析.....	(66)
5.4.5 结构设计结论.....	(68)
5.5 其他的初步设计技术.....	(68)
5.6 初步设计文件.....	(68)

5.7 初步设计检查	(69)
5.8 详细设计	(71)
5.8.1 图形工具	(71)
5.8.2 详细设计语言	(74)
5.8.3 表格工具	(77)
5.8.4 详细设计工具比较	(80)
5.9 详细设计标志文件	(80)
5.10 详细设计检查	(81)
5.10.1 非正式的详细设计检查	(81)
5.10.2 较正式的详细设计检查	(81)
5.10.3 正式的设计检查	(82)
5.11 设计管理	(82)
5.11.1 安排设计任务	(83)
5.11.2 会见用户/客户	(83)
5.12 结语	(83)
第六章 结构编程	(84)
6.1 程序结构	(84)
6.1.1 结构的重要性	(84)
6.1.2 结构编程	(85)
6.1.3 程序格式	(86)
6.2 程序文件	(86)
6.3 防止错误	(90)
6.4 编程风格	(91)
6.5 可移植的软件	(93)
6.6 实现工具	(94)
6.6.1 准备源程序	(94)
6.6.2 处理语言	(95)
6.6.3 辅助评价程序	(95)
6.6.4 一般的开发环境	(95)
6.7 标志文件与检查	(96)
6.8 管理措施	(96)
6.9 结语	(96)
第七章 软件测试	(97)
7.1 测试目的与定义	(97)
7.1.1 软件的可靠性与失灵	(97)
7.1.2 测试与调试	(97)
7.2 测试顺序	(98)
7.3 单元测试	(99)
7.4 组装测试	(99)

7.4.1	组装测试计划和实施方案	(100)
7.4.2	组装过程	(101)
7.4.3	测试报告	(102)
7.5	验证测试	(102)
7.5.1	测试计划	(102)
7.5.2	测试过程	(103)
7.5.3	系统测试	(103)
7.6	测试组	(103)
7.7	测试原理	(104)
7.7.1	一般的指南	(104)
7.7.2	测试情况设计技术	(104)
7.7.3	检查错误表	(105)
7.7.4	调试程序	(106)
7.8	自动测试工具	(107)
7.8.1	程序流分析程序	(107)
7.8.2	测试驱动程序	(107)
7.8.3	测试台	(108)
7.8.4	典型系统	(108)
7.8.5	性能分析工具	(108)
7.9	结语	(109)
第八章	配置管理	(111)
8.1	软件配置	(111)
8.2	标志文件与分界线	(111)
8.3	配置标识	(112)
8.4	修改控制	(114)
8.4.1	控制过程	(114)
8.4.2	修改控制委员会	(115)
8.5	配置审查	(116)
8.6	说明配置情况	(116)
8.7	结语	(116)
第九章	软件维护	(117)
9.1	维护工作分类	(117)
9.2	可维护性	(118)
9.3	维护过程	(118)
9.3.1	组织	(118)
9.3.2	计划	(119)
9.3.3	实施	(119)
9.3.4	错误报告	(119)
9.3.5	修改评估	(120)

9.3.6	纠正错误	(120)
9.3.7	维护过程结语	(121)
9.4	维护课题	(122)
9.4.1	接收经过验证测试的软件	(123)
9.4.2	修订已交付使用的软件	(123)
9.4.3	配置管理	(123)
9.4.4	技术人员调配	(124)
9.4.5	维护记录与保存	(126)
9.5	维护检查	(126)
9.6	结语	(127)
第十章	小型项目的软件工程	(128)
10.1	小型项目的种类	(128)
10.2	小型项目的定义	(129)
10.3	小型项目的开发	(129)
10.4	小型项目的维护	(129)
10.5	小型项目的软件配置	(129)
10.5.1	定义文件	(129)
10.5.2	开发文件	(130)
10.5.3	维护文件	(130)
10.5.4	配置管理	(130)
10.6	结语	(130)
第十一章	管理问题	(131)
11.1	管理联络方法	(131)
11.1.1	计划、分析和管理方法	(132)
11.1.2	定义阶段的管理问题	(132)
11.2	软件项目失败	(133)
11.2.1	失败的原因	(133)
11.2.2	失败的根本原因	(134)
11.3	标准	(135)
11.4	软件工程教育	(135)
11.5	如何建立软件工程	(136)
11.5.1	实现方法	(137)
11.5.2	选择试点项目	(137)
11.6	结语	(138)
第十二章	帮助理解软件工程方法的例子	(139)
序言		(139)
系统说明		(139)
软件计划		(140)
需求说明书		(142)

附录1、初步的用户手册.....	(148)
附录2、希望的测试结果.....	(153)
附录3、词汇.....	(153)
设计文件	(154)
附录1、FORTFORM的详细设计	(163)
FORTFORM的组装测试说明书.....	(165)
验证测试说明书.....	(170)
FORTFORM的用户手册	(173)
附录A文件格式	(183)
附录A·1 系统说明书.....	(183)
附录A·2 软件计划.....	(188)
附录A·3 软件需求说明书.....	(193)
附录A·4 设计文件	(198)
附录A·5 测试说明书	(203)
附录A·6 程序模块开发文件夹	(206)
附录A·7 维护文件	(210)
附录A·8 安装手册与用户指南	(216)
附录B.软件方法与工具	(221)
附录B·1 面向数据结构的设计	(221)
附录B·2 软件工具	(229)
附录B·3 详细设计语言	(231)
附录C 软件工程参考文献.....	(239)
附录D 软件工程的定量方法.....	(244)

第一章 本书简介

软件工程是开发和维护软件的规范化方法，是从一般曾叫做“计算机程序设计”的活动发展而来的。本书由GEC撰写，于1976年内部首次发表，它向软件经理和技术人员提供软件工程各步的入木三分的解释。

本书向您介绍软件项目计划、需求说明书、设计、编程、测试和软件维护的管理和技术方法。详细介绍对软件项目的成功起重要作用的文件格式和检查方法。还讨论追踪、报告和控制作为软件开发的一部分而产生的各种文件之方法。

本章介绍软件工程，并推荐使用本书的方法。随后各章详细讨论软件工程的各个阶段和步骤。

1.1 问题的提出

十年来，计算机软件已成为计算机系统的主体。决定这类系统成败的关键是计算机软件。软件通过挖掘硬件的“潜力”而提供新的功能，进而导致伴随计算机产品的“人工智能”。

不幸的是软件的开发和检测有不少问题。在1960年代，软件开发成本，在整个计算机系统的成本中，仅占很少一部分。硬件十分昂贵，管理工作和技术工作主要用于控制硬件的成本。

随着微电子技术的发展，硬件成本垂直下降，五年内，硬件成本降低两个甚至三个数量级是极为普通的事。然而，软件成本却持续上升。到1980年代，在许多计算机系统的开发过程中，软件成本在整个系统成本中，占绝大部分。

1.1.1 软件开发

软件开发问题的根源在于开发过程中缺乏质量控制。因为，在1960年代，软件成本相对来说很低，管理部门也不重视。因而，管理技术和开发技术均很落后。在许多组织中，这种现象尚存。因此，有下列问题：

1. 软件成本和开发进度的估计常常是不准确的。超支和贻误工期损伤了开发员的信心，导致质量下降，总是满足不了客户的要求。

2. 即使已证明可以采用计划、需求说明、设计、编程和测试方法，但仍没有系统地开发软件。

3. 没有撰写适当的软件文件。计算机软件应该有一套文件，这些文件应在开发过程中（而不是之后）撰写，并作为帮助经理控制和评价开发工作取得的进展之标志。

4. 软件质量可疑。没有始终如一地使用已证明的保证软件质量的技术——检查、测试和验证，结果产生软件质量问题。

5. 软件常常无法维护。纠正许多程序中的潜在错误是困难的，在新的计算机上使用原来的程序或提供客户要求的新功能，实际上是不可能的。可以重用的软件仍然是难于得到的。

使用本书介绍的软件工程技术，可以解决上述问题。

1.1.2 软件检测

软件凭借它自己的强有力的功能，已经成为计算机市场上的主要产品。许多不熟悉软件检测技术的经理，因购买了不满足所述要求的软件产品而受挫。伴随软件检测的问题，与购买任何其他产品时的问题大不相同。然而，为确保正确地提出下述范围的问题，要小心谨慎：

1. 要获得的软件产品必须满足所述的要求。买主必须充分理解卖主的产品提供的功能和详细说明自己的要求。做到这两点的方法是有的。

2. 若为适应买主的运行环境，必须修改软件，则修改成本必须合理，且要估计实际的成本。通常，改变客户的环境要便宜些！

3. 卖主必须为软件产品提供足够的支持。

和开发问题一样，检测问题也是可以避免的。伴随软件工程的培训方法也能用于软件检测，

1.2 解决方法

用软件生命期的观点可以得到解决上述问题的方法。与硬件一样，计算机软件也应该用一系列的精心控制和系统执行的阶段来开发。

1.2.1 软件生命期

整个软件生命期可分为三个阶段：定义、开发和维护。在定义阶段，订软件开发计划，提出预算和估计工程建设期；分析和详细说明需求。在开发阶段，使用已为实践证明的设计、编程和测试方法，把软件需求变成可运行的程序。最后，在维护阶段，纠正现场碰到的问题；对不同的运行环境，进行自适应维护；提高交付使用的软件之功能。

一系列的工程步出现在上述各软件开发阶段，每一步以公布可以检查的标志而结束。

1.2.2 工程方法

开发任何产品的工程方法：

- * 在定义、开发和维护阶段的各步要求使用已为实践证明可行的方法；
- * 要求进行一系列的检查，以确保软件产品开发进程中的质量；
- * 确定在各步要产生的具体文件；
- * 鼓励使用 and 开发能加速开发进程的工具与方法；
- * 提供可以从最初的产品设想追踪到最后的产品生产之途径。

任何学科都有个发展过程，起码是几年。在一些典型的领域，诸如机械和电子工程，已开发了一般能接受的工程科学。软件工程是通向软件工程科学的必由之路。

1.2.3 软件工程

软件工程是应用于软件的定义、开发和维护的一套方法、工具、文件、实践、标准和

步骤。软件工程与硬件工程十分类似，它具有1.2.2节讨论过的一般工程的特性。

本书其余各章叙述下列各步：

- 1.计划：定义项目范围、资源预算和建设工期；
- 2.需求分析：详细定义功能和性能要求，设计约束条件和验证标准；
- 3.设计：井井有条的设计过程，它把功能需求转换为：i) 软件结构的表述；ii) 详细的处理过程表述；
- 4.编程：利用程序设计语言，把设计表述转换为机器能理解的形式；
- 5.测试：一系列的步骤：i) 试图揭露程序中的潜在的错误；ii) 把程序模块装配成软件包；iii) 按照功能要求，验证软件。
- 6.维护：把步骤1~5（加上一些控制方法）应用于现有软件的维护。

软件工程是一套完整的方法，只有不折不扣地应用这套方法，才能解决伴随软件开发的问题。

1.3 本书中引进的主要概念

本书介绍了许多软件工程的管理和实践的重要概念。本节概述这些概念。

1.3.1 生命期

本书始终讨论软件“生命期”概念。软件生命期确定软件工程事件表。当把软件定义为计算机系统的一部分时，软件生命期就开始了。一旦系统工程师把一定的功能/性能与软件结合起来时，软件生命期的定义阶段就开始了。

如前所述，定义阶段集中在软件项目计划和需求分析/说明。利用已有的项目说明书，软件计划员和分析员定义项目范围，并估计投资和建设工期。然后，再把项目范围扩展为详细的需求说明书，并进行检查。软件需求说明书构成软件生命期的开发阶段的所有工作之基础。

需求必须转换为最终能由计算机执行的形式。设计阶段从这种需求转换开始，应用设计方法来产生软件结构和软件的处理过程表述。这种结构和处理过程表述起动编程步，编程步产生源程序。最后，应用测试来保证软件质量和满足功能要求。

维护阶段要做定义和开发阶段做过的各项工作，只不过此时是把软件工程方法用于已有的程序。与维护活动密切配合的一种正式的控制方法，即软件配置管理。修改软件时，配置管理保证软件的完整性。

1.3.3 软件项目估算

本书在多处讨论软件项目计划与控制。软件项目估算，在项目开始时向经理提供预算和建设工期等指导性意见。事实上，一个项目上马与否可以以这种预算/建设工期为基础，而预算/建设工期可用历史资料、经验模型或自动的成本/工期工具来估算。

本书将介绍几种估算技术。每一种都需要彻底了解项目的范围；参考过去搜集的资料；要完成的工作之逻辑划分；并用不同的方法进行对比检查，以保证做出的估计之正确性。

1.3.3 需求分析

软件需求分析为以后的软件工程师工作提供一个蓝图。软件工程的各步必须说明对需求的“可追踪性”。

在系统的定义阶段，当计算机系统的所有成分（例如，硬件、软件、信息和人）已经确切划分，它们之间的界面也有定义时，需求分析就开始了。用手工或自动工具来模拟信息结构和信息流；撰写软件需求说明书。

1.3.4 软件设计

设计是软件工程师的技术核心。设计师一开始把软件需求映射为软件结构，其成分是定义得很好的程序模块。任选一种设计方法来确定软件结构，撰写文件和进行检查。随着设计过程的发展，设计师的工作范围越来越窄。一旦确定了软件结构，就可将过程设计技术工具分别用于各个程序模块，所得的结果就是详细设计，这是结构编程的基础。

1.3.5 结构编程

软件设计必须转换为机器可执行的形式。结构编程应用一套程序设计原则来产生源程序，它具有良好的设计风格和透明性等。因为设计过程也是结构化的，所以，设计和源程序之间有一一对应的关系。

源程序是编译程序的输入，编译程序把它译为目标程序。

1.3.6 测试步骤和方法

本书用了许多节来讨论各种各样的保证软件质量的问题。软件测试是保证软件质量的技术之一。

测试是从程序模块开始的。软件工程师使用测试情况设计方法和/或自动测试工具，来揭露隐蔽的软件错误。测试完所有的程序模块后，必须把它们组装在一起，构成完整的软件系统。组装测试是装配和测试计算机软件的一套策略和方法。最后，组装成的软件必须通过验证测试，以保证满足所有的软件要求。

1.3.7 小型软件项目的软件工程

软件工程适用于任何规模的软件开发项目。然而，开发小项目时（例如，工作量少于六个人-月，非关键的软件，一般的应用软件），可以缩短某些软件工程步骤；可以缩短各种软件文件，并把它们合在一起，检查可以不那么正式，但必须彻底。本书给小型软件项目提供了一套管理和技术方法。

1.3.8 软件维护

计算机软件开发成之后，常常用10年甚至20年。在此期间，必须纠正发现的错误；必须使它适应于新的环境；必须提高系统的功能。总之，这些活动统称为软件维护——占软件总投资百分之六十以上的一个阶段。

在理想的情况下，用于软件维护阶段的方法与步骤和开发阶段使用的是相同的。不幸

的是，有的程序根本没有用软件工程方法来开发，撰写的文件质量很低，用不便于修改的方式进行设计，偶尔也做些修改（有时没有修改记录）。只有用软件工程方法才能减轻这种沉重的维护负担。

1.3.9 软件配置管理

在整个软件生命期中，要产生各种各样的文件，报告，表格和数据。这一切构成软件配置——以多种形式表述软件的信息组。软件配置管理（SCM）是标识、控制、审查和解释软件配置的一组活动。

本书单独用一章来讨论软件配置问题，但伴随SCM的活动穿插于整个软件生命期。文件必须适当标识，以便存档和检查；修改必须审查，以保证符合标准；当前的项目状况必须向管理部门报告。

1.3.10 文件和检查

软件生命期中的各步结束时都要产生文件，这些文件是软件配置的一部分。撰写文件是各软件工程步的自然结果。文件检查作项目开发进展标记。

1.4 本书的目的

为满足这样两种要求我们撰写了本书：1）相当于一本指导性的教科书，但以工业生产中常用的术语来叙述软件工程；2）起建立软件工程方法论之源泉的作用。本书的主要目的是：

1. 为软件管理提供一套内部联系紧密的指导原则、方法和工具、以控制软件工程各步；
2. 向软件技术人员提供一套适用于需求分析、设计、编程、测试和维护的方法与工具；
3. 确定文件格式，检查技术和配置控制方法，以帮助保证软件质量；
4. 介绍一套适用于广阔的应用问题和环境的软件工程方法。

本书涉及的领域复盖了管理理论和计算机科学。它起着引进所谓的“80年代最重要的工程方法”的关键作用。

1.4.1 经理指南

对于直接或间接负责软件开发或检测的经理来说，本书却是一本指南：

1. 本书为项目计划与投资估算提供了技术指南，它介绍了软件项目的预算和计划技术，确定了在项目开发初期必须提出来研究的重要问题；
2. 本书推荐了一些组织技术人员的方法，详细讨论了工程队的组织问题；
3. 本书定义了项目标志，介绍了用来评审这些标志的方法，通过回答这些问题：“我希望要什么？应该如何叙述它？应于何时公布它？”，本书向经理提供了一些贯穿于软件项目的参考点；
4. 本书单独介绍了一些特殊的控制方法，随着软件配置的发展，可用这些方法来管理它。控制能使经理知道软件项目在任何时候的状态；

5.本书介绍了许多保证质量的技术：检查方法，测试方法，配制控制与管理方法，

6.本书为撰写正式的软件工程标准与方法手册提供了一个基础。

虽然，经理应该了解本书介绍的所有课题，但许多节是技术性的，不必仔细研究。本章1.6节提供了使用本书的指南。

1.4.2 技术人员指南

应用软件工程方法和工具是软件技术人员的责任。本书通过介绍下列课题，起着应用软件工程方法的指南作用：

1.本书介绍软件需求说明书，设计，编程和测试的技术，工具和叙述方法。来源于许多渠道的信息，已综合于本书中，所以，软件技术人员能回答下列问题：“什么方法可以用？那一种好些？那一种次之？”

2.本书介绍了一些检查技巧，解释并说明了各种软件检查方法。

3.本书介绍了一套工具，包括图形表述（例如数据流图，结构图）方法和自动工具（例如PSL/PSA，PDL）。

4.本书介绍了各种文件格式及其内容，还详细介绍了软件配置的各种成分。

虽然，软件技术人员的注意力集中在工具和技术上，但它们几乎对本书的全部内容感兴趣（参看本书的1.6节）。

1.5 本书的内容

本书介绍软件生命期的各个阶段。三至七章按照软件工程各步在定义和开发阶段出现的顺序，叙述这些步骤。第九章深入介绍软件维护阶段。其余各章研究与整个软件生命期有关的课题。

第二章概述软件工程，它把这一学科介绍给不熟悉软件工程方法的人们。第八章介绍配置管理——一套适用于整个软件生命期的标准和方法。第十章讨论适用于小型软件项目的软件工程方法。第十一章介绍与软件开发有关的课题。

最后，第十二章详细介绍一个实际项目的软件配置。

书末的附录详细介绍了文件格式（附录A），讨论了补充的方法和工具（附录B），对软件工程方面的参考文献，加了注解（附录C和附录D）。

1.5.1 软件工程方法与步骤库

本书的三至十章是开发软件的方法与步骤库。第三章介绍计划管理、估算方法和项目追踪与控制的步骤。第四章介绍软件需求分析、说明方法与工具。第五章介绍初步设计、详细设计的方法、步骤和工具。文件和检查步骤也在三至五章讨论。

第六章讨论伴随程序风格、透明度、写程序文件、可移植性和编程工具等方面的课题。第七章介绍各个测试步骤、讨论测试策略、技术和工具。

第八章概述软件配置管理，介绍对控制软件配置必须的步骤与组织问题。第九章讨论软件维护方法与文件。

第十章介绍适用于小型软件项目的方法与步骤。本章是以三至九章介绍的材料为基础的。

1.5.2 文件格式库

书末的附录A详细介绍了软件配置内的各种文件格式，附录A.1至A.9概述各个文件的内容，并逐节介绍它们的内容。

第十二章介绍了一个软件项目的所有文件：系统说明书，开发计划，需求说明书，设计文件，源程序清单和测试文件。

1.5.3 文献目录库和一般的信息库

与软件管理和开发有关的课题在本书中介绍过许多次。第二章和第十一章主要介绍这方面的内容。

附录C有流行的文献之目录，按课题组织，且加有注释，可以作为进一步阅读的指南。

1.6 如何使用本书

我们设计本书，使其起指导性的教科书和参考书的作用。一旦理解了软件生命期和回顾了软件工程的各个步骤，本书的各章都可以单独使用。

每个读者都应读第一章和第二章。这两章介绍了撰写本书的目的和本书的内容，并概述了软件工程。每个读者可以以此为起点，挑选自己感兴趣或需要的部分来阅读和使用。下面几小节建议使用本书的方法。

1.9.1 课题与读者关联矩阵

虽然通读本书似乎总是需要的，但本书介绍的某些课题所有读者都会感兴趣，有的课题，经理更感兴趣，而有的课题，技术人员更感兴趣。课题——读者关联矩阵如图1.1所示。此矩阵列出了本书讨论过的34个主要课题，并说明了它们的读者范围。

标有“A”字的课题是所有读者感兴趣的课题。这些课题组成了应该熟悉的材料之最小子集。除这些课题外，经理还应掌握标有“M”的课题，而技术人员应精通标有“T”字的课题，技术经理也可以选一些标有“T”字的课题来研究。

1.6.2 建议经理阅读的内容

下面向不同级别和类型的经理推荐了各自应熟悉的章节。应读的章节用章节号表示，附录用相应的字母表示。括号内的数字或字母表示可以任选的部分。

高级经理：1, 2, (11), C

中级（项目）经理：1, 2, 3, 8, 9, 10, 11, (4-7, A), D

技术经理：1, 2, 3, 4, 8, (5-7), 9, 10, (11), A, (B), 12, F

1.6.3 为技术人员推荐的内容

下面向不同级别和类型的技术人员推荐一些应熟悉的章节。符号含义同1.6.2节。

系统工程师与分析员：1, 2, 4, 8 (3), (5-7, 9-12, A, B), D.

软件设计师：1, 2, 4, 5, B, 8, 9 (3), 6, 7, (10-12, A), D

软件工程师：1, 2, 4-9, (3, 10, 11), 12, A, B, D.