

软件工程

方法·工具和实践

(第二版)

冯玉琳 赵保华 编著 中国科学技术大学出版社



TP36/F533

3. 填空

软件工程(第二版)

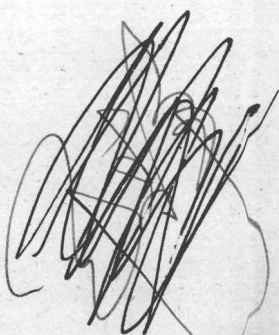
——方法·工具·实践

冯玉琳 赵保华 编著

基础
技术 { 代
数据流图

中国科学技术大学出版社

1992·合肥



(皖)新登字 08 号

软件工程(第二版)
——方法·工具·实践
冯玉琳 赵保华 编著

*

中国科学技术大学出版社出版
(安徽省合肥市金寨路 96 号, 230026)
中国科学技术大学印刷厂印刷
安徽省新华书店发行

*

开本: 850×1168/32 印张: 13.75 字数: 368 千
1988 年 2 月第 1 版 1992 年 9 月第 2 版
1997 年 1 月第 7 次印刷
印数: 48001—53000 册
ISBN 7-312-00652-3/TP·170 定价: 11.80 元

第二版前言

本书第一版于1988年出版之后,被国内许多大专院校和其他单位用作软件工程课的教材或教学参考书,受到普遍欢迎。为适应软件工程技术的发展,考虑到国内大学生和有关技术人员应用水平的不断提高,我们在第二版中对原书内容作了较大修改和补充,以使之更适合于教学,并反映近十年来软件工程领域的一些重要研究成果和进展。

在第1—2章,增加了软件开发方法综述和软件质量保证的有关内容。在第4—5章,增加了面向对象的分析和设计部分,并力求从统一的系统模型分析观点出发,介绍面向数据流和面向对象这两种不同方法。第5章还增加了有关原型开发技术、界面设计技术等的一般介绍。Jackson方法一章,原书只讲了基于数据结构的设计,是该方法的早期研究成果;本书增加了基于进程模型设计部分,反映了该方法的最新进展。第10章强调了软件工程管理的重要性。在第11章,原书中的小编译系统实例篇幅规模过大,不便于教学,本书另取一个小排版系统实例取代之。附录只保留了文档格式和课程实习两部分,增加了一些课程实习项目。考虑到原书中的课程实习示例是“虚构”的,本书从一个实际应用项目中提取部分文档列出,更具示范意义。

本书新版主要由冯玉琳执笔修改和定稿,赵保华参与了部分章节和附录内容的修订。书中更新的小项目示例和课程实习示例分别由中国科学技术大学的黄涛同志和安徽工业设备安装公司徐科进同志编写。

我们在修订过程中,尽可能引入新的观点和方法,并注意兼顾不同层次读者的需要和保持原有的简明实用风格。尽管如此,但难免还

会有疏忽谬误之处，敬请同行指正。

最后，值此再版之机，向一切曾帮助和关心本书定稿和出版的同志致以衷心的感谢。

二 再版前言

冯玉琳 赵保华

本书自出版以来，承蒙广大读者和同行专家的厚爱，多次再版，深受好评。在本书出版过程中，得到了许多专家和同行的帮助，特别是中国科学技术大学图书馆的同志，为本书的出版提供了许多便利。本书在出版过程中，得到了许多专家和同行的帮助，特别是中国科学技术大学图书馆的同志，为本书的出版提供了许多便利。

本书在出版过程中，得到了许多专家和同行的帮助，特别是中国科学技术大学图书馆的同志，为本书的出版提供了许多便利。本书在出版过程中，得到了许多专家和同行的帮助，特别是中国科学技术大学图书馆的同志，为本书的出版提供了许多便利。

本书在出版过程中，得到了许多专家和同行的帮助，特别是中国科学技术大学图书馆的同志，为本书的出版提供了许多便利。本书在出版过程中，得到了许多专家和同行的帮助，特别是中国科学技术大学图书馆的同志，为本书的出版提供了许多便利。

第一版前言

从计算机程序设计的领域独立出“软件工程”这个名词,标志着人们试图用“工程化”的思想来指导并解决软件研制中面临的种种困难和混乱。软件工程的目的在于研究一套科学的工程方法,并与此相适应,发展一套方便的工具系统,力求用较少的投资获得高质量的软件。

本书是软件工程的一本简明教程,目的在于使学生对软件工科学的全貌有个概要的了解,从而使学生的注意力从个人小程序的独立活动转移或扩展到大程序设计方面。

与其它一些课程的特点不同,软件工程课是一门“设计”课程,强调实际训练。课堂教学只讲授软件工程中的一些基本概念、方法和原则,更重要的是,学生必须参加一个从计划、分析、设计到编码、测试的软件开发全过程,以便从中得到实际的从事软件工程的训练和经验。课程将提出一些设计题目供学生选用。

全书分为三部分,十三章。

第一部分为 1—2 章,介绍软件和软件工程的基本概念。

第二部分为 3—9 章,按照软件生命期的观点依次介绍各个阶段,包括软件计划、软件需求分析、软件设计、软件编码、软件测试和软件维护。由于软件工程的方法很多,这里只重点介绍结构化的分析和设计方法,包括 DFA, SD, SP, Jackson 方法等。

第三部分为 10—13 章,讨论软件工程的外在支持,包括人的因素方面,例如软件管理;物的因素方面,例如软件工具和环境,强调软件开发工具和环境在软件工程中的重要地位,并对这方面当前发展动态作了一般介绍,特别是对以 UNIX 为背景的支持环境系统作了一些典型分析。

小项目软件有“小”的特点,专辟一章讲述,并给出了一个小编译系统的设计例子。

本书结束语,再次强调了软件工程实践的重要性。有一个课程实习项目提要列在附录 C 中,供读者选择参考用。

本书的主要特点是简明,内容与教学环节紧密配合,重点介绍软件工程的方法和工具,并强调学生的实践训练,适合于大专院校计算机软件专业作为软件工程课的教材或教学参考书。

为顾及一般读者,本书叙述力求通俗易懂,重在思想方法介绍。略去部分章节后可供具有程序设计初步经验的广大工程技术人员自学之用。

在本书的编写过程中,许多同志提出了不少宝贵意见,初稿形成时曾与中国科学院软件研究所仲萃豪、丁茂顺等同志,电子工业部华东计算技术研究所蔡林希同志以及中国科学技术大学计算机系唐策善同志进行过许多有益的讨论。安徽大学李庆同志曾参加本教程教学并很好地指导了课程设计实习。中国计算机函授学院钱洲胜、张宁等同志为本书做了大量实际工作。本书的附录 D:课程实习项目示例是中国科学技术大学硕士研究生桂自强同志编写的。对以上这些同志的热情帮助和辛勤劳动,谨致深切谢意。

本书是根据我校近几年使用的软件工程讲义整理而成的。因时间仓促,其中难免出现有疏忽谬误之处,敬请读者指正。

冯玉琳 赵保华

1986年9月

目 录

第二版前言	(I)
第一版前言	(II)
第一章 绪论	(1)
1.1 什么是软件工程	(1)
1.2 软件工程面临的问题	(2)
1.2.1 软件价格	(2)
1.2.2 软件可靠性	(3)
1.2.3 软件维护	(4)
1.2.4 软件生产率	(4)
1.2.5 软件再应用	(5)
1.3 软件和软件生命期	(7)
1.4 软件开发方法	(9)
第二章 软件评价	(13)
2.1 软件的质量标准	(13)
2.2 软件质量度量模型	(14)
2.3 软件复杂性	(16)
2.4 软件可靠性	(19)
2.5 软件性能评价	(21)
2.6 软件质量保证	(22)
第三章 软件计划	(23)
3.1 可行性研究	(23)
3.2 软件计划内容	(24)
3.3 软件需求描述	(26)
3.4 软件价格估算	(27)

第四章 软件需求分析	(34)
4.1 需求分析的目标和任务	(34)
4.2 系统模型分析	(36)
4.3 数据流分析技术	(40)
4.3.1 数据流模型	(40)
4.3.2 DFA 描述	(42)
4.3.3 数据流分析实例	(57)
4.4 面向对象分析技术	(66)
4.4.1 概念模型	(67)
4.4.2 实例:机票预订系统分析.....	(70)
4.5 软件规格说明确认	(78)
4.5.1 原型化方法	(80)
4.5.2 原型开发技术	(82)
4.6 形式化软件规格说明	(83)
4.6.1 代数规格说明	(84)
4.6.2 Z 框架规格说明	(86)
4.7 软件分析工具	(88)
第五章 软件设计	(91)
5.1 软件设计概论	(91)
5.2 模块化设计准则	(93)
5.2.1 结构和过程	(93)
5.2.2 模块化	(94)
5.2.3 模块独立性	(96)
5.2.4 设计准则	(102)
5.3 结构化设计技术	(108)
5.3.1 数据流图的类型	(108)
5.3.2 设计步骤	(109)
5.3.3 SD 实例.....	(117)
5.4 面向对象的设计技术	(131)

5.4.1	面向对象的模块分解	(131)
5.4.2	实例:机票预订系统设计.....	(133)
5.5	界面设计技术	(140)
5.6	详细设计表示法	(143)
5.7	软件设计工具	(152)
第六章	Jackson 方法	(154)
6.1	基于数据结构的设计	(154)
6.1.1	数据结构表示法	(155)
6.1.2	Jackson 结构设计方法	(156)
6.1.3	一个实例	(165)
6.2	基于进程模型的设计	(170)
6.2.1	进程模型	(172)
6.2.2	JSD 步骤	(173)
6.2.3	形式化开发技术	(178)
第七章	软件编码.....	(184)
7.1	结构化程序设计	(184)
7.2	编码风格	(187)
7.3	程序设计语言	(191)
7.3.1	语言类别	(191)
7.3.2	语言选择	(193)
7.4	软件编码工具	(194)
第八章	软件测试.....	(196)
8.1	软件测试的原则	(197)
8.2	软件测试方法	(198)
8.3	测试用例的设计	(200)
8.4	测试过程和步骤	(208)
8.4.1	概述	(208)
8.4.2	单元测试	(209)
8.4.3	整体测试	(212)

8.4.4	有效性测试	(216)
8.4.5	系统测试	(217)
8.5	纠错技术	(217)
8.6	测试工具	(219)
第九章	软件维护	(222)
9.1	软件维护的内容	(222)
9.2	维护任务	(225)
9.3	维护的副作用	(229)
9.4	维护工具	(231)
第十章	软件工程管理	(233)
10.1	软件产品的特点	(233)
10.2	软件工程管理的内容	(235)
第十一章	小项目软件的开发	(240)
11.1	小项目软件的开发过程	(240)
11.2	实例:小排版软件系统开发	(242)
11.2.1	需求分析	(242)
11.2.2	系统设计和实现	(250)
第十二章	软件开发环境	(280)
12.1	程序设计环境概论	(280)
12.1.1	环境定义	(280)
12.1.2	环境特点	(281)
12.2	程序设计环境分类	(282)
12.2.1	基于语言的程设环境	(283)
12.2.2	基于操作系统的程设环境	(284)
12.2.3	基于方法论的程设环境	(286)
12.2.4	程设环境发展的新方向	(287)
12.3	UNIX 程序设计环境	(289)
12.3.1	可适应性	(289)
12.3.2	源代码变换	(290)

12.3.3	shell 编程	(291)
12.3.4	分离编译	(294)
12.3.5	LEX 和 YACC	(296)
12.4	语法制导的程序设计环境	(300)
12.4.1	概述	(300)
12.4.2	程序编辑	(302)
12.4.3	程序执行和查错	(308)
12.4.4	实现	(309)
12.4.5	环境自动生成	(311)
12.5	用户软件工程方法和环境	(315)
12.5.1	自外向内和界面原型	(315)
12.5.2	IIS 设计	(316)
结束语		(331)
附录 A 文档格式		(332)
A.1	软件计划任务书	(332)
A.2	软件需求规格说明书	(334)
A.3	软件设计说明书	(335)
A.4	软件测试任务书	(337)
A.5	软件维护文档	(338)
A.6	用户使用手册	(339)
附录 B 课程实习		(341)
B.1	课程实习项目提要	(341)
B.2	课程实习项目示例	(343)
参考文献		(423)

第一章 绪 论

1.1 什么是软件工程

软件工程是研究“大”规模程序设计的方法、工具和管理的工程科学。

首先，软件工程的要点是“大”程序设计。假设一个人在一个月内能够完成 1000 行代码的设计，按照如此的工作效率，一个 10,000 行代码的程序是否一个人花十个月或者十个人花一个月就能完成呢？回答是否定的。当程序的规模从 1000 行增加到 10,000 行时，源代码行增加了十倍，但是，程序复杂性程度的增加却远不止十倍。

当许多人共同设计一个由许多模块组成的大型系统时，在许多年的时间内会出现许多不同的版本，面对着如此的许多人、许多模块、许多年和许多版本……，如此之“多”的活动，如何进行设计、开发和维护，这里不仅涉及到技术的方面，如设计方法、版本控制等，而且关系到行政管理的方面，如价格评估，人员组织等。

60 年代末期国际上出现了“软件危机”，其主要表现是：软件质量差，可靠性难以保证；软件成本增长难于控制，极少有在预定的成本预算内完成的；软件开发进度难于控制，周期拖得很长；软件的维护很困难，维护人员和费用不断增加。例如：IBM 公司开发的 OS/360 系统，耗资几千万美元，花费了五千多人年，拖延了几年才交付使用。交付使用后仍不断发现新的错误。有的软件耗费了大量的人力财力，结果半途而废。

考虑到研制一个软件系统同研制一台机器或建造一座楼房有许多共同之处，因此可以参照机械工程、建筑工程中的一些技术来进行

软件的研制,人们试图用“工程化”的思想作指导来解决软件研制中面临的困难和混乱,从根本上解决软件危机。1968年北大西洋公约组织的学术会议第一次使用了“软件工程”这个名词。

正象所有的工程科学一样,软件工程自身遵循着一套科学的设计原理和方法,以这些方法为前导,人们发展一系列的软件工具系统来帮助工程软件的开发。但是,软件工程又与其他各种工程科学很不相同。软件是抽象的、逻辑性的产品,不是实物性的。研制和维护软件本质上是一个“思考”的过程,很难对它进行控制。到目前为止,软件的设计基本上还无统一的设计标准,无准确的数量分析,无足够的可靠性保证以及无有效的维护手段,这就决定了软件的研制和开发较其它工程项目要困难得多。

综言之,软件工程的目的在于研究一套科学的工程方法,并与此相适应,发展一套方便的工具系统,力求用较少的投资获得高质量的软件。

1.2 软件工程面临的问题

摆在软件工程科学面前有许多方面的棘手问题,以下就软件价格、软件可靠性、软件维护、软件生产率和软件再应用等方面作一些简单分析。

1.2.1 软件价格

由于新的电子器件的出现,计算机硬件的功能和质量在不断提高,而成本却大幅度下降。计算机硬、软件投资所占比例中,软件投资在迅速上升。美国空军计算机投资中软件所占比例,1955年低于20%,1970年剧增到60%,1985年为85%,软件投资比重上升的曲线如图1.1所示。

随着计算机应用愈来愈广泛,新的应用领域要求开发新的软件。另一方面,当新一代计算机取代原有的旧计算机时,软件系统也要随

之更新。软件社会需求增长的速度有增无减。虽然精确的数字很难统计,但据现有公布资料^[5],1985年世界软件总产值已超过1400亿美元,并正在以每年12%的速度增长;按照这种增长速度计算,到1995年,世界年度软件总产值将达4350亿美元。

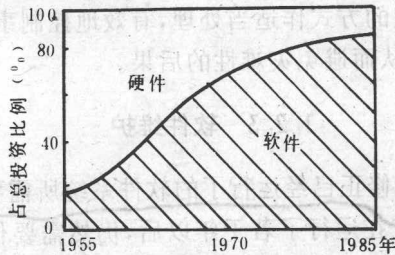


图 1.1 软硬件投资比例

问题在于软件生产基本上仍然处于手工状态,软件是知识高度密集性技术的综合产物,人力资源远远不能适应这种迅速增长的软件社会需求,所以,软件价格上升的势头必然还将继续下去。

1.2.2 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。通常大约有40%左右的软件开发代价要花在软件设计之后的测试和排错上,即使如此,也不能保证经测试的软件就没有错误。现代的程序设计技术,如结构程序设计,能有助于减少程序设计中的错误;不幸的是,我们很难保证一个大的软件系统确实是无任何错误的。

大的软件系统的测试是一件很复杂费时的事情,程序的任何改变都可能会涉及到许多人,这就使得软件系统花在测试阶段的代价非常之大。为了提高软件的可靠性,就要付出足够的代价。

除了软件的正确性以外,在更广泛的意义上,软件的可靠性还应该包括安全性和健壮性。

对于合理的一组输入,系统会给出正确的结果;而对于用户的有

意或者无意的不合理输入,系统应能拒绝这种输入,并指出输入的不合理性,提醒用户注意。这是软件系统的安全性。对于不合理的输入束手无策,甚至导致系统失败,这样不安全的系统是不能使用的。

健壮性是指软件系统对环境变化的适应性。当软件系统所处的环境发生变化时,例如存贮溢出,硬件故障等意外事件发生时,系统都能按照某种预定的方式作适当处理,有效地控制事故的蔓延,不致丢失重要的信息,从而避免灾难性的后果。

1.2.3 软件维护

软件维护是指修正已经运行了的软件系统所需要做的工作。几乎所有的软件系统在运行了若干年以后,仍然需要作这样或者那样的修改和补充。一个软件开发机构,要把很大的精力花在维护已有的软件上。随着更多软件的生产,用之于软件维护的力量还会愈益增加,这样势必会束缚开发组织无法去生产新的软件。

为什么维护需要花费如此大的精力呢?这是因为已经运行的程序还总是在变化,故障要排除,改进要加进去,而且优化也要做。不仅当前的版本要改变,而且仍在使用的往年的版本也要修改,可能即将投入使用的新的版本也在修改,除了解决原有的问题需要精力外,改变本身又会带来新的问题,也要花精力去解决。

因此,如何减少软件维护的总工作量,也就成为软件工程的主要目标之一。

1.2.4 软件生产率

计算机的广泛使用使得软件的需求量迅速上升。世界各国普遍感到软件人力资源的缺乏,这种趋势仍在继续增长下去。但是,目前的软件生产仍然基本上处于手工状态,生产率很是低下,而生产出来的软件的质量又很不理想,不适应市场对软件的大量需求。如何以较少的投资,获得“高产优质”的软件,这是软件工程的主要目标之一。其实现途径有二,即良好的软件工程设计方法和方便的软件设计工

具环境。

由于软件研制过程毕竟是相当复杂的,涉及的因素很多,所以软件工程设计方法总是具有不同程度的灵活性与试探性。一般说来,软件方法规定了

(1)明确的工作步骤

(2)具体的文档格式

(3)确定的评价标准

近几十年来,软件工程已研究发展了各种各样的软件工程设计方法,这些方法的实用范围是各不相同的,有的方法适用于数据处理系统,而有的方法适用于实时控制系统;各种软件方法的风格也迥然不同。有的方法仅仅是一组指导性的原则,而有的方法则有较具体的处理规则,有的方法建立在严密的数学基础之上,而有的方法则是实际经验的总结。本课程只能选择一些比较实用的、有代表性的方法进行讨论。

方法和工具两者之间有着密切的联系,相辅相成。在软件开发维护的不同阶段,都应该有这样或者那样的作为软件工具的程序系统,帮助人们自动地完成许多数据分析和处理工作。这样的工具环境愈完善,软件的设计和维护也就愈方便。这是当前提高软件生产率的一个很重要的方面。人们希望研究出一套系统的软件方法,一组配套成龙的软件工具,从而为软件人员提供一个能复盖整个软件生命期的良好的工作环境,使软件生产率大幅度提高。

1.2.5 软件再应用

软件再应用也称软件复用,对于提高软件生产率、降低软件成本具有特殊重要的意义。当人们一次又一次地去设计一些互相类同的程序时,很多的劳动都花费在一些基本重复的工作上,软件再应用技术就是旨在减少这种重复。

有各种级别的软件再应用,软件规格说明、软件模块、软件代码乃至软件文档,都可以是软件再应用的软件单位。