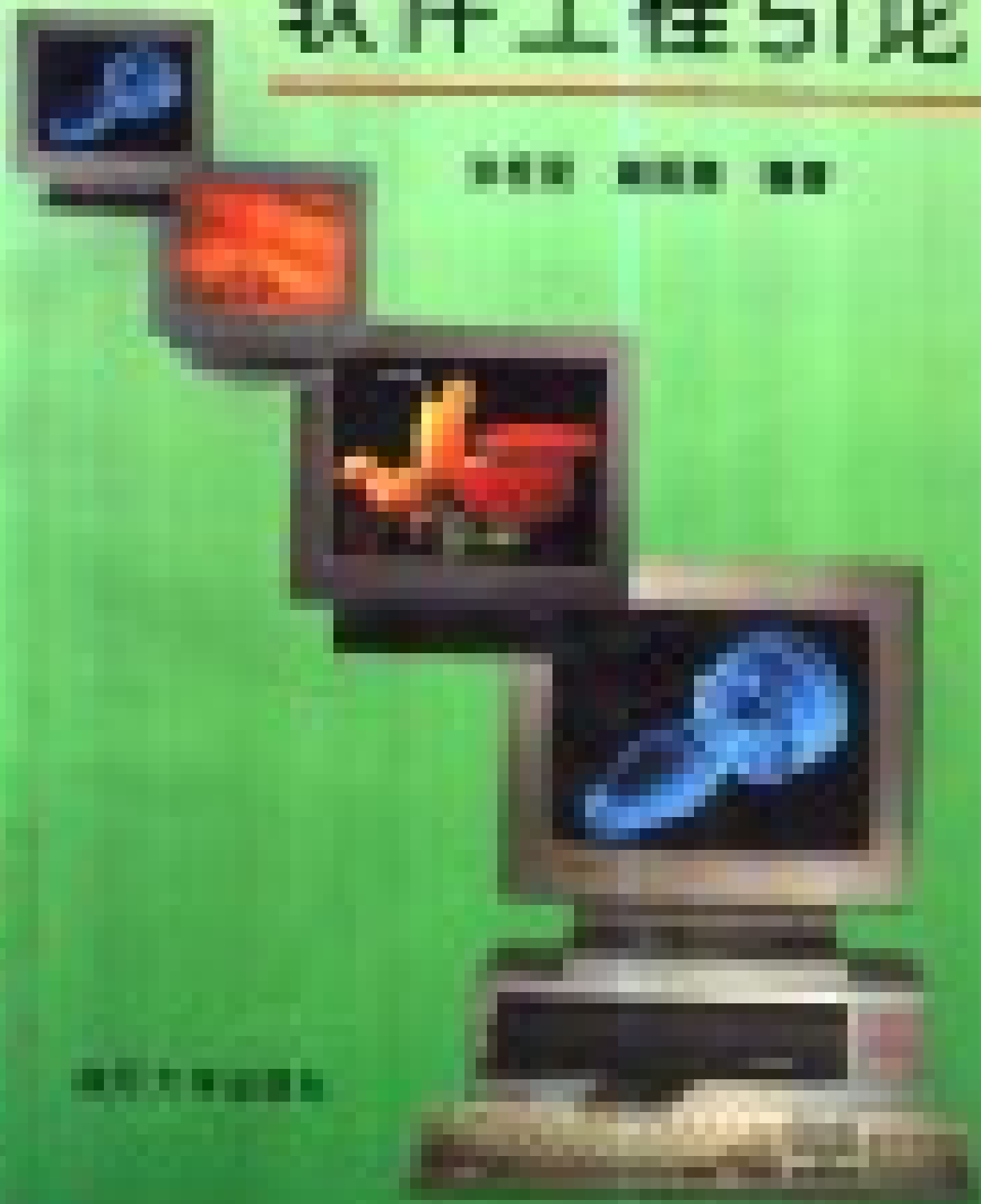
计算机大专教材系列

软件工程引论

孙桂茹　越国瑞　编著

南开大学出版社

软件工程引论



软件工程引论

孙桂茹　赵国瑞　　编著

南开大学出版社

内 容 简 介

本书较系统地阐述了软件开发、维护和管理等方面的软件工程学的概念、原理以及常用的技术方法。全书从软件生存周期各个阶段的任务、过程、方法及其所使用的描述工具进行顺序地讲述；在后一部分也讲述了软件工程的管理技术及软件工程开发环境。全书各章节都列举了实例，通俗易懂。

本书可作为计算机有关专业的教材或教学参考书，也可供计算机用户阅读。

软件工程引论

孙桂茹 赵国瑞 编著

南开大学出版社出版
（天津八里台南开大学校内）
邮编 300071 电话 3508542
新华书店天津发行所发行
天津宝坻第二印刷厂印刷

1995 年 11 月第 1 版　　1995 年 11 月第 1 次印刷
开本：787×1092×1/16　　印张：13
字数：321 千　　印数：1-8000

ISBN 7-310-00880-4
TP·43　定价：14.00 元

“计算机大专教材系列”编委会

出版说明

随着计算机应用的日益深入、普及，目前在我国正在兴起学习计算机专业知识的高潮，各种有关计算机的书籍如雨后春笋般涌现出来，使广大读者大有应接不暇之势。但是，已经出版的这些书籍中，有的偏深偏专，取材偏多偏全，适合有一定基础的计算机专业人员阅读参考；有的则是普及性读物，只适合急于入门的计算机爱好者使用；在为数不多的教材中，大都是为计算机专业本科生使用而编写的，不适合成人教育和大专类学生的需要。鉴于这种形势，我们决定编写一套适合于计算机类各专业大专学生和成人教育使用的教材。这套教材共有十种，虽然它还不能完全覆盖上述办学层次教学计划中的所有课程，但是它包含了培养一个计算机类专科生的主要教学内容。其中入门的教材有《计算机应用基础》和《C 语言程序设计》；属于专业基础的教材有《16 位微型计算机原理与接口》，《汇编语言程序设计》，《数据结构》和《操作系统》；应用性较强的有《单片机及其应用》，《数据库系统教程》，《计算机网络基础》和《软件工程引论》。

这套教材贯彻了理论联系实际、学以致用的原则。在取材方面，不追求包罗万象、面面俱到，而着力保证把最基本、最实用的部分包含进来。在叙述方面，力求做到深入浅出，尽量用实例来说明基本概念和基本方法。我们希望这套教材不仅能适合课堂讲授的需要，也便于广大读者自学。这套教材由南开大学计算机与系统科学系的教师们编写而成，他们之中既有教学经验十分丰富的教授、副教授，也有活跃在计算机应用最前沿的青年教师。这些教师不仅具有教本科生、研究生的教学经验，也具有教大专生和成人教育的教学经验。这就使这套教材的质量有了基本的保证。但是由于我们初次编写这类教材，尚未经过实践的检验，缺点和不足之处在所难免，敬希同行专家和广大使用者批评指正。

前　言

为了摆脱软件危机的困扰，一门研究软件开发和维护的普遍原理与技术的工程学科——软件工程学迅速发展起来，并已成为计算机科学技术的一个重要分支。软件工程的目标在于研究一套科学的工程方法，以及与此相应的方便的工具系统，用来指导软件开发研究工作。

软件工程学研究范围非常广泛，包括技术方法、工具和管理等许多方面，本书是软件工程的入门介绍，它着重从实用角度来讲述软件工程的基本原理和技术方法。

本书正文共十章。第一章介绍软件工程学的产生以及它的基本原理、概念和方法。第二章到第八章介绍了软件生存周期的各阶段的任务、过程、方法和工具。第九章介绍了软件工程的管理技术。第十章介绍了软件工程开发环境和未来前景的展望。附录中详细列出了编写软件开发文档的规范。全书各章末均附有习题，作为读者熟悉本章内容的提要。

《软件工程引论》凝聚了作者多年教学经验和科研成果，很适合于从事软件开发的实际工作者参考，也可以作为大专院校的教材。

本书在编写过程中始终得到了南开大学计算机与系统科学系有关人士的大力支持，特别是朱瑞香教授、陈有祺教授和韩维桓教授的帮助。南开大学出版社的王家骅编审不仅审阅了全部书稿，而且对本书的组稿、编排始终给予了具体的指导，在此对这些同志一并致谢。

由于时间仓促，加之编者水平有限，书中难免有不当之处，诚恳希望广大读者和有关专家指正。

孙桂茹　赵国瑞

1995 年 3 月于天津

目　录

第 1 章　软件工程概述

第 2 章　可行性研究

第3章 需求分析

第4章 总体设计

第 5 章 详细设计

第6章 编码

第7章 软件测试

第8章 软件维护

第9章 软件工程管理

第 10 章 软件工程环境

第1章

软件工程概述

随着计算机应用的逐步扩大，软件需求量迅速增加，规模也日益增大，长达数万行、数十万行乃至百万行以上的软件，已不鲜见。软件规模的膨胀，带来了它的复杂度的增加，而开发一个数万以至数百万行的软件，即使是富有经验的程序员，也难免顾此失彼。其结果是，大型软件的开发费用耗资庞大。

随着微电子学技术的进步，计算机硬件成本逐步下降，而且质量稳定性不断提高。这是与成本上升、质量不可靠的软件形成鲜明对照。可见，软件已经成了限制计算机系统发展的关键因素。

认真研究和解决软件的这一危机，已经成为计算机科学中的一个迫切问题。因此，逐步形成了计算机科学中一门新学科——软件工程学。

1.1 软件的发展和软件危机

计算机软件的进步，是与计算机硬件的发展和计算机的普及深入分不开的。它同其他事物的发展规律一样，也必然要经历产生，发展直至成熟的过程。

1.1.1 软件的发展

在计算机系统发展的早期(60年代中期以前)，通用硬件相当普遍，软件只是根据具体应用而编写程序，强调的也只是编程技巧，因此对软件的开发没有系统化实施。

从计算机系统发展的第二时期(由60年代中期到70年代中期)，计算机在众多的领域得到了应用，计算机系统也从单用户应用发展成为多用户应用，并且出现了实时系统和数据库管理系统。

在计算机系统数量不断增加的同时，计算机软件库开始膨胀，但是计算机用户必须随计算机系统的变化而不断修改自己的程序以适应新环境的要求，这就给软件的维护工作造成了障碍；再加上某些软件个体化的不可维护性，因而早期的软件危机就出现了。

70年代中期到80年代为计算机系统发展的第三代。计算机应用有了更深入更普遍的发展，出现了分布式系统即多台计算机并行工作的情况。硬件的发展速度已经超过了人们对软件的需求速度，因此使得硬件价格下降，软件价格急剧上升。如图1—1所示，这样便造成了软件危机的加剧，致使更多的计算机科学家着手研究软件工程学的科学理论、方法和实现等一系列问题。

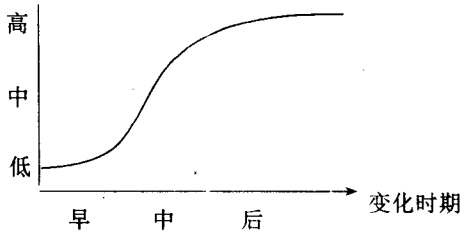


图1—1 硬件、软件成本变化趋势

1.1.2 软件危机

我们把上述软件的发展情况，用表 1—1 详细叙述。从表中可分析三个时期的软件发展特点。

表 1—1 软件发展特点

特点 \ 时期	程序设计	程序系统	软件工程
软件所指	程 序	程序及说明书	产品软件
主要程序设计语言	汇编语言及机器语言	高级语言	高级语言系统
软件工作范围	程序编写	包括了设计和测试	软件生存期
需求者	程序设计者本人	少数用户	市场用户
软件开发组织	个人	开发小组	开发小组、软件工厂
软件规模	小型	中小型	大、中(小)型
决定软件质量的因素	个人程序技术	小组技术水平	管理水平
开发技术和手段	子程序、程序库	结构化程序设计	数据库、开发工具开发环境、工程化开发方法、标准和规范。
维护责任者	程序设计者	开发小组	专职维护人员
硬件特征	价高、存储容量小、工作可靠性差。	降价、速度容量及可靠性有明显提高。	向超高速、大容量以及微型化发展。
软件特征	完全不受重视	软件技术的发展不能满足需要，出现软件危机。	开发技术有了前进，但未获突破性进展，价高软件危机并未完全摆脱。

从表中可总结如下几点：

(1)人们对软件的认识不断地在改变；

(2)人们对软件的需求促进了软件的发展；

(3)软件的生产逐步走向规范化、科学化和工程化的轨道；

(4)软件的危机从产生到发展，目前正在逐渐摆脱或消灭危机。

下面我们来讨论什么是软件危机和它产生的原因。

1. 什么是软件危机

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，这些问题表现在以下几个方面：

(1)对软件开发成本和进度的估计常常很不准确。实际成本比估计成本有可能高出一个数量级，实际进度比预期进度拖延几个月甚至几年的现象并不罕见，这样就降低了软件开发组织的信誉，从而引起了用户的不满。

(2)用户不满意已开发出的软件系统的情况时有发生。软件开发人员常常在对用户的要求了解得不很清楚，或者所要解决的问题没有确切认识的情况下，就急于编程序，结果使开发出的软件系统不符合用户的需要。

(3)软件产品的质量往往靠不住。由于软件可靠性和质量保证的确切的定量概念刚刚出现不久，软件质量保证技术(审查、复审和测试)还没有完全贯彻于软件开发的全过程中，这样就使得软件产品质量缺乏可靠保证。

(4)软件常常是不可维护的。在很多情况下，程序中的错误是非常难以改正的，或者开发出的系统不完全适应新的硬件环境，或者不能在原系统的基础上增加新的功能等，对于这些软件的维护性，正是开发人员努力追求的目标。

(5)软件通常没有适当的文档资料。计算机软件不仅仅是程序，它还包括一整套文档资料，这套文档资料应能反映系统的开发过程，并且与程序代码一起表示软件系统的产品，软件开发管理人员可以使用这些文档资料作为“里程碑”，来管理和评价软件开发工程的进展状况；对于软件开发人员，文档资料是他们进行信息交流的工具；对于软件维护人员，文档资料是他们进行软件维护的依据。文档资料不合格会给开发人员和维护人员带来许多困难和问题。

(6)软件成本在计算机系统总成本中所占的比例逐年上升。由于微电子学技术的进步和生产自动化程度的不断提高，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。美国 1985 年软件成本大约已占计算机系统总成本的 90%。

(7)软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的趋势。

以上仅列举了软件危机的明显表现，当然，与软件开发和维护有关的问题远远不止这些。

2. 产生软件危机的原因

造成软件危机的原因是多方面的，有属于开发人员方面的，也有软件本身的特点方面的。

软件是程序以及开发、使用和维护程序需要的所有文档。因此，软件产品必须由一个完整的配置组成，所以应该肃清那种主观盲目制定计划、草率编程以及不考虑维护工作必要性的糊涂观念。对于计算机系统来说，软件是逻辑部件，软件开发过程没有统一的、公认的方法论和规范指导，因此也造成软件维护的困难。

一个软件从定义、开发、使用和维护，直到最终被废弃，要经历一个漫长的时期，通常把软件经历的这个漫长的时期称为生存周期。软件开发最初的工作是问题定义，也就是确定要求解决的问题是什么；然后进行可行性研究，决定该问题是否存在一个可行的解决办法；接下来应该进行需求分析，也就是深入具体地了解用户的要求，在所要开发的系统上与用户取得完全一致的看法。到了软件的开发时期，首先需要对软件进行设计，然后才能进入编写程序阶段，程序编写完之后，还必须经过大量的测试工作才能交付使用，所以，编写程序只是软件开发过程中的一个阶段，而且此阶段的工作量仅是软件开发全部工作量的 10～20%。

从前面叙述软件发展过程的图 1- 1 中可以看出，软件成本在计算机系统总成本中所占的

比例越来越大，而软件成本的55～70%用于软件维护。美国贝尔实验室曾统计出改正一个问题在软件的生存周期中需要付出的代价，如图1—2所示。

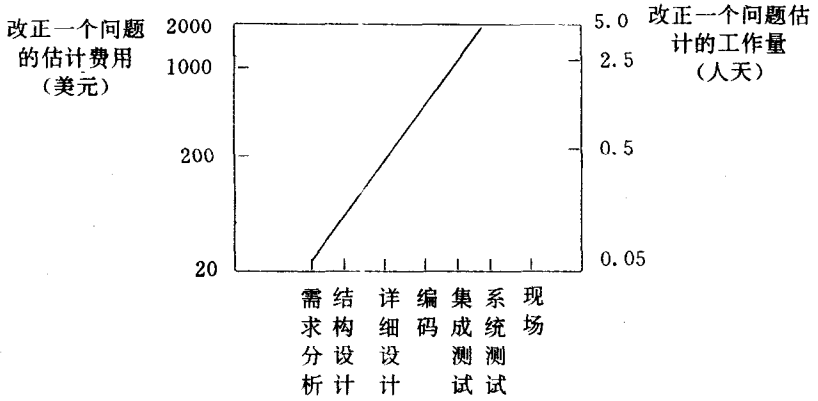


图1—2　改正一个问题需要付出的代价

了解产生软件危机的原因，澄清错误认识，建立起关于软件开发和维护的正确概念，还仅仅是解决软件危机的开始，而全面解决软件危机需要一系列综合措施。

3. 解决软件危机的途径

从上述软件危机的现象和软件危机的原因可以看出，要想彻底摆脱危机也不是一件易事，不是一朝一夕就能圆满解决的。但是在这期间如何根据软件的特点，它的应用范围，排除人们的传统观念和错误认识是非常重要的。

首先，软件开发不是个体劳动的神秘技巧，而应该是一个有良好的组织、实行严格管理、各类人员协同工作、共同完成的工程项目。因此要注意吸收其他工程项目中积累起来的丰富经验和科学原理、技术和方法。

从软件本身的特点看，也要不断地总结开发技术和方法，探索出一些软件工程中应遵循的原理过程和技术方法，从而指导软件工程技术人员，使软件工程学健康科学地发展。

另一方面，从计算机的硬件体系结构上能否来一次彻底革命，改变多年来冯·诺依曼计算机统治的天下，使得各个领域的用户用自然语言，把问题描述清楚（或简单的规则描述）？如是，计算机就能自动完成推理和计算，正确解决用户的问题。

总之，为了解决软件危机，既要有技术措施（方法和工具），又要有必要的组织管理措施。

1.2　软件开发工程化和软件生存周期

软件开发的工程化，打开了我们的眼界。我们认真分析软件开发的各个环节，从而得出了程序编码只是软件工程的一小部分工作，而在它产生的前后还有大量的工作要做；只有这样，才能保证软件的可靠性，可维护性，从而提高软件的生产率。经过人们多年工作经验，北大西洋公约组织的软件人员于1968年提出了“软件工程”的概念。从此以后，以此种思想为指导，软件工程学越来越成熟起来。广大软件工作者正采用工程的概念、原理、技术和方法，来开发与维护软件，并且把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。