软件工程导论

张海藩 编著

清华大学出版社

软 件 工 程 导 论

张海藩 编著

清 华 大 学 出 版 社

内 容 提 要

本书系作者根据他最近几年在北京大学校内外多次讲授“软件工程”课程所用的讲义改写而成。书中较全面系统地介绍了软件工程的概念、原理以及典型的技术方法。本书的特点是既注重系统性和科学性，又注重实用性；既有广度（全面概括地介绍了各种常用方法），又有深度（具体详尽地讲述了一种方法论）；既有原理性论述，又有丰富的实例与之配合，特别是正文后面的两个附录，比较完整地讲述了用软件工程的方法开发两个实际软件的过程，对读者深入理解软件工程学很有帮助。本书正文共十章，第一章是概述，第二章至第九章顺序讲述软件生存周期各阶段的任务、过程、方法和工具，第十章集中讨论软件工程使用的管理技术。

本书可供有一定实际经验的软件工作人员和需要开发应用软件的广大计算机用户阅读，也可作为高等院校计算机系“软件工程”课程的教材或教学参考书。

软 件 工 程 导 论

张海藩 编著

责任编辑 贾仲良

☆

清华大学出版社出版

北京 清华园

铁道出版社激光照排排版

军事科学院印刷厂印装

新华书店北京发行所发行

☆

开本：787×1092 1/16 印张：20.75 字数：531千字

1987年6月第1版 1987年6月第1次印刷

印数：0001——12500

统一书号：平装15235·275 定价：平装3.45元

精装15235·276 精装5.45元

前　言

计算机工业发达国家在发展软件的过程中曾经走过不少弯路，受过许多挫折，至今仍然经受着“软件危机”的困扰。人们开发优质软件的能力大大落后于计算机硬件日新月异的进展和社会对计算机软件不断增长的需求，这种状况已经严重妨碍了计算技术的进步。

为了摆脱软件危机的困扰，一门研究软件开发与维护的普遍原理和技术的工程学科——软件工程学——从六十年代末期开始迅速发展起来了，现在它已经成为计算机科学技术的一个重要分支，一个异常活跃的研究领域。严格遵循软件工程方法论可以大大提高软件开发的成功率，能够显著减少软件开发和维护中的问题。

软件工程学（通常简称软件工程）研究的范围非常广泛，包括技术方法、工具和管理等许多方面，软件工程又是一门迅速发展的新兴学科，新的技术方法和工具不断涌现，真可谓五花八门层出不穷。因此，在一本书中不可能包含软件工程的全部内容。本书《软件工程导论》是软件工程的入门介绍，着重从实用角度讲述软件工程的基本原理、概念和技术方法，同时也尽量注意书的全面性和系统性。希望本书既能对实际的软件开发工作有所帮助，又能为读者在今后深入研究这门学科奠定良好的基础。

本书正文共十章。第一章概括介绍软件工程学产生的历史背景以及它的基本原理、概念和方法。第二章至第九章按软件生存周期的顺序介绍各个阶段的任务、过程、方法和工具。第十章着重讨论软件工程的管理技术。正文后面有两个附录，比较完整地介绍了两个实际软件的开发。附录 A 着重讲述从问题定义到实现的过程，把这个具体例子和课文前几章的内容结合起来学习，有助于加深对一些基本概念和方法的理解。附录 B 讲述一个交互式正文编辑程序的设计和描述，它是上机实习的好材料（例如，可以实习把设计翻译成程序、设计测试方案以及维护的方法）。

《软件工程导论》可以为多种读者服务。本书通俗易懂，实例丰富，既有对多种常见方法的全面概括介绍，又有对一种典型方法的深入详尽介绍，很适合于有一定实践经验的软件工作者和广大计算机用户参考或自学；对于高等院校计算机系高年级本科生和研究生来说，本书可以做为软件工程课程的教材。

本书根据编者最近几年在北京大学校内外多次讲授“软件工程概论”课所用的讲义改写而成，改写时充分考虑了在校内外讲授这门课程时广大学员的建议和要求，并且增加了最近收集到的大量新材料。

编者在美国马里蓝大学进修期间，听过该校朱耀汉（Yaohan Chu）教授讲授的软件工程课，并且在朱教授指导下从事过一些软件开发工作，这些都为本书的编写奠定了基础。本书编写前征求了复旦大学计算机系副教授潘锦平同志对写作大纲的意见，初稿写出后又征求了潘锦平和北大计算机系副系主任许卓群等同志对内容编排的意见，承蒙他们给予真诚的鼓励并且提出了许多宝贵的具体建议。此外，本书编写过程中董士海副教授曾经提供了许多很有价值的材料。谨在此向他们表示衷心的感谢！

本书不当之处敬请广大读者不吝赐教。

编者　1986 年

目　录

第一章　软件危机和软件工程

在工业发达的国家中计算机系统已经经历了三个不同的发展时期。随着计算机应用日益普及和深化，正在运行使用着的计算机软件的数量以惊人的速度急剧膨胀，而且现代软件的规模往往十分庞大，包含数百万行代码、耗资几十亿美元、花费几千人年的劳动才开发出来的软件产品，在七十年代已经屡见不鲜了。

由于微电子学技术的进步，计算机硬件成本每五年下降两至三个数量级，而且质量稳步提高；与此同时，计算机软件成本却在逐年上升，质量没有可靠的保证，软件开发的生产率也远远跟不上普及计算机应用的要求。软件已经成为限制计算机系统发展的关键因素。

在计算机系统发展的早期时代所形成的一些错误概念和做法，已经严重地阻碍了计算机软件的开发，更严重的是，用错误方法开发出来的许多大型软件几乎根本无法维护，只好提前报废，造成大量人力物力资源的浪费。西方计算机科学家把软件开发和维护过程中遇到的一系列严重问题统称为"软件危机"，并且在六十年代后期开始认真研究解决软件危机的方法，从而逐步形成了计算机科学技术领域中的一门新兴的学科——计算机软件工程学，通常简称为软件工程。

1.1　计算机系统的发展简史

过去人们往往按照计算机硬件的演变来划分计算机系统发展的时期，然而为了了解计算机软件发展演变的过程，特别是为了了解软件危机是怎样产生的，又是如何加剧的，从而探索解决软件危机的途径，我们应该更全面地回顾计算机系统发展的简短历史，按照计算机应用领域的演变而不是仅仅根据硬件特点的演变来划分计算机系统发展的时期。

从世界上出现第一台计算机到六十年代中期是计算机系统发展的早期时代。在这个时期人们用很大力气研究和发展计算机硬件，经历了从电子管计算机到晶体管计算机的变革；然而，人们对计算机软件的研究和发展却不够重视，认为软件开发是事后考虑的问题，因此基本上没有系统化的软件开发方法存在。所谓软件开发在那个时期不过是根据应用的需要写出能够运行的程序。由于硬件价格昂贵，运算速度低，内存容量少，所以当时的程序员非常强调"程序设计技巧"，把缩短每一微秒 CPU 时间和节省每一个二进制存储单元，作为程序设计（也就是那个时期的软件开发）的重要目标。

在计算机系统发展的早期时代，大多数系统采用批处理方式工作，只有个别系统是交互式系统。当时通用硬件相当普遍，软件却是为每个具体应用而专门设计的，而且大多数软件的开发者和使用者是同一个（或同一组）人。编写程序的人要设法使程序在机器上运行起来，并且负责使用这个程序解决预定的问题，如果使用过程中程序发生错误也由这个人设法修改。由于这种个体化的软件环境，使得软件设计通常是在人们头脑里进行的一个隐含的过程，而且除了程序清单之外，一般没有其他文档资料保存下来。

从六十年代中期到七十年代中期是计算机系统发展的第二代时期。在这个时期硬件经历了从晶体管计算机到集成电路计算机的变革，CPU 速度和内存容量都有了很大提高，从而为

计算机在众多领域中的应用提供了潜在的可能性。为了更有效地利用硬件的能力，计算机系统普遍采用多道程序多用户的分时工作方式；能够从多个数据源采集数据，高速进行处理，并在很短的时间内(通常只有几毫秒)产生输出信息，从而控制实际过程进行的实时系统，开辟了计算机应用的一个新领域；联机辅助存储设备的进展导致出现了第一代数据库管理系统。

这个时期的另一个重要特征是出现了“软件作坊”，广泛使用产品软件。这是因为随着计算机应用的普及和深化，需要的软件往往规模相当庞大，以致单个用户无力开发；此外，许多不同的部门和企业往往需要相同的或类似的软件，各自开发就会使人力浪费很大。在这种形势下“软件作坊”就应运而生了，许多用户不再自己开发软件而是购买或定做软件。不过在这个时期“软件作坊”开发软件时，基本上仍然沿用早期时代形成的开发方法。

随着计算机系统数量的不断增加，计算机软件库就开始膨胀。但是在程序运行中发现错误时必须设法改正；当用户的要求有改变时也必须相应地修改程序；当买进新硬件或操作系统的新版本时通常必须改变程序以适应新的环境。上述种种软件维护工作，以令人吃惊的比例耗费资源。更严重的是，许多程序的个体化特性使得它们最终成为不可维护的。”软件危机”就开始出现了！一九六八年北大西洋公约组织的计算机科学家曾在联邦德国召开国际会议讨论软件危机问题，在这次会议上正式提出并使用了“软件工程”这个名词，一门新兴的工程学科就此出现了。

计算机系统发展的第三代从七十年代初期开始，一直沿续到八十年代。这个时期硬件发展的特点是从集成电路计算机进步到超大规模集成电路计算机，高性能低成本的微处理机大量涌现，发展日新月异。

这个时期计算机应用又有了进一步的普及和深化，开辟了更多新的应用领域。分布式系统使用多台计算机共同解决一个问题，各台计算机并行工作分别完成各自的子任务，同时各台间进行必要的通信和同步，这大大增加了计算机系统的复杂程度；利用计算机硬件提供的简单算术运算和逻辑运算的能力，模拟人类复杂的思维解题过程的人工智能系统，需要十分复杂的计算机软件才能实现。

硬件的迅速发展已经超过人们提供支持软件的能力，然而，硬件只提供了潜在的计算能力，如果没有软件来驾驭和开发这种能力，人类并不能有效地使用计算机。在计算机系统发展的第三代，软件危机进一步加剧了。软件维护费用占数据处理总预算的百分之五十以上，软件开发生产率远远满足不了对于新系统的需求。为了对付日益严重的软件危机，计算机科学家和软件工程师开始认真研究和采用软件工程学。

那么，什么是软件危机的确切含义呢？为什么会出现软件危机？软件工程学提出了哪些基本原理和策略来对付软件危机呢？这些正是本章下面几节要讲述的问题。

1.2 软件危机

1.2.1 什么是软件危机

软件危机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。这些问题绝不仅仅是“不能正常运行的”软件才具有的，实际上几乎所有软件都不同程度地存在这些问题。概括地说，软件危机包含下述两方面的问题：如何开发软件，怎样满足对软件的日益增长的需求；如何维护数量不断膨胀的已有软件。具体地说，软件危机主要有下述一些表现：

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。这种现象降低了软件开发组织的信誉。而为了赶进度和节约成本所采取的一些权宜之计又往往损害了软件产品的质量，从而不可避免地会引起用户的不满。

2. 用户对“已完成的”软件系统不满意的现象经常发生。软件开发人员常常在对用户要求只有模糊的了解，甚至对所要解决的问题还没有确切认识的情况下，就仓促上阵匆忙着手编写程序。软件开发人员和用户之间的信息交流往往很不充分，“闭门造车”必然导致最终的产品不符合用户的实际需要。

3. 软件产品的质量往往靠不住。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）还没有坚持不懈地应用到软件开发的全过程中，这些都导致软件产品发生质量问题。

4. 软件常常是不可维护的。很多程序中的错误是非常难改正的，实际上不可能使这些程序适应新的硬件环境，也不能根据用户的需要在原有程序中增加一些新的功能。“可再用的软件”还是一个没有完全做到的、正在努力追求的目标，人们仍然在重复开发类似的或基本类似的软件。

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，还应该有一整套文档资料。这些文档资料应该是在软件开发过程中产生出来的，而且应该是“最新式的”（即和程序代码完全一致的）。软件开发组织的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；软件开发人员可以利用它们作为通信工具，在软件开发过程中准确地交流信息；对于软件维护人员而言，这些文档资料更是至关重要必不可少的。缺乏必要的文档资料或者文档资料不合格，必然给软件开发和维护带来许多严重的困难和问题。

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。图 1.1 描绘了美、日两国计算机硬件成本和软件成本在计算机系统总成本中所占比例逐年变化的情况，可以看出在 1985 年软件成本大约占总成本的 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。软件产品“供不应求”的现象使人类不能充分利用八十年代计算机硬件提供的巨大潜力。

以上列举的仅仅是软件危机的一些明显的表现，与软件开发和维护有关的问题远远不止这些。那么，为什么会发生软件危机呢？

1.2.2 产生软件危机的原因

在软件开发和维护的过程中存在这么多严重问题，一方面与软件本身的特点有关，另一方面也和软件开发人员的弱点有关。

软件不同于硬件，它是计算机系统中的逻辑部件而不是物理部件。在写出程序代码并在计算机上试运行之前，软件开发过程的进展情况较难衡量，软件开发的质量也较难评价，因此，管理和控制软件开发过程相当困难。此外，软件在运行过程中不会因为使用时间过长而被“用坏”，如果运行中发现错误，则很可能是遇到了一个在开发时期引入的在测试阶段没能检测出来的故障。因此，软件维护通常意味着改正或修改原来的设计，这就在客观上使得软件较难维护。

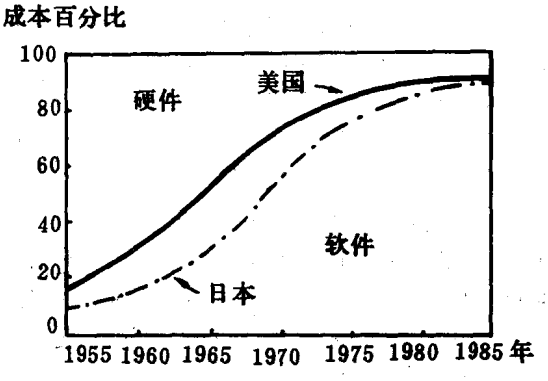

图 1.1 美、日两国计算机硬件和软件成本变化趋势

软件本身独有的特点确实给开发和维护带来一些客观困难，但是人们在开发和使用计算机系统的长期过程中，也确实积累和总结出了许多成功的经验。如果坚持不懈地使用经过实践考验证明是正确的方法，许多困难是完全可以克服的，过去也确实有一些成功的范例[3]；但是，目前相当多的软件专业人员对软件开发和维护还有不少糊涂观念，在实践过程中或多或少地采用了错误的方法和技术，这可能是使软件问题发展成软件危机的主要原因。

与软件开发和维护有关的许多错误认识和做法的形成，可以归因于在计算机系统发展的早期时代软件开发的个体化特点。今天这些错误认识仍然迷惑着不少软件人员，因此有必要对若干主要的错误认识做一番剖析，以树立正确的概念来指导软件开发和维护工作。在下面的叙述中引号内是典型的错误认识，在每个错误认识后面是对它的剖析。

“有一个对目标的概括描述就足以着手编写程序了，许多细节可以在以后再补充。”

事实上，对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。只有用户才真正了解他们自己的需要，但是许多用户在开始时并不能准确具体地叙述他们的需要，软件开发人员需要做大量深入细致的调查研究工作，反复多次地和用户交流信息，才能真正全面、准确、具体地了解用户的要求。对问题和目标的正确认识是解决任何问题的前提和出发点，软件开发同样也不例外。急于求成仓促上阵，对用户要求没有正确认识就匆忙着手编写程序，这就如同不打好地基就盖高楼一样，最终必然垮台。

“所谓软件开发就是编写程序并设法使它运行。”

一个软件从定义、开发、使用和维护，直到最终被废弃，经历了一个漫长的时期，这就如同一个人要经过胎儿、儿童、青年、中年、老年，直到最终死亡的漫长时期一样。通常把软件经历的这个漫长的时期称为生存周期。正如上面刚刚讲过的，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；然后要进行可行性研究，决定该问题是否存在一个可行的解决办法；接下来应该进行要求分析，也就是深入具体地了解用户的要求，在所要开发的系统（不妨称之为目标系统）必须做什么这个问题上和用户取得一致的看法。经过上述软件定义时期的准备工作才能进入开发时期，而在开发时期首先需要对软件进行设计（通常又分为一般设计和详细设计两个阶段），然后才能进入编写程序的阶段，程序编写完之后还必须经过大量的测试工作（需要的工作量通常占软件开发全部工作量的 40～50%）才能最终交付使用。所以，编写程序只是软件开发过程中的一个阶段，而且在典型的软件开发工程中，编写程序所需的工作量只占软件开发全部工作量的百分之十至二十。

另一方面还必须认识到程序只是完整的软件产品的一个组成部分，在上述软件生存周期的每个阶段都要得出最终产品的一个或几个组成部分（这些组成部分通常以文档资料的形式存在）。Boehm 曾经指出：“软件是程序以及开发、使用和维护程序需要的所有文档。”这也就是对软件的定义。所以，一个软件产品必须由一个完整的配置组成，应该肃清只重视程序而忽