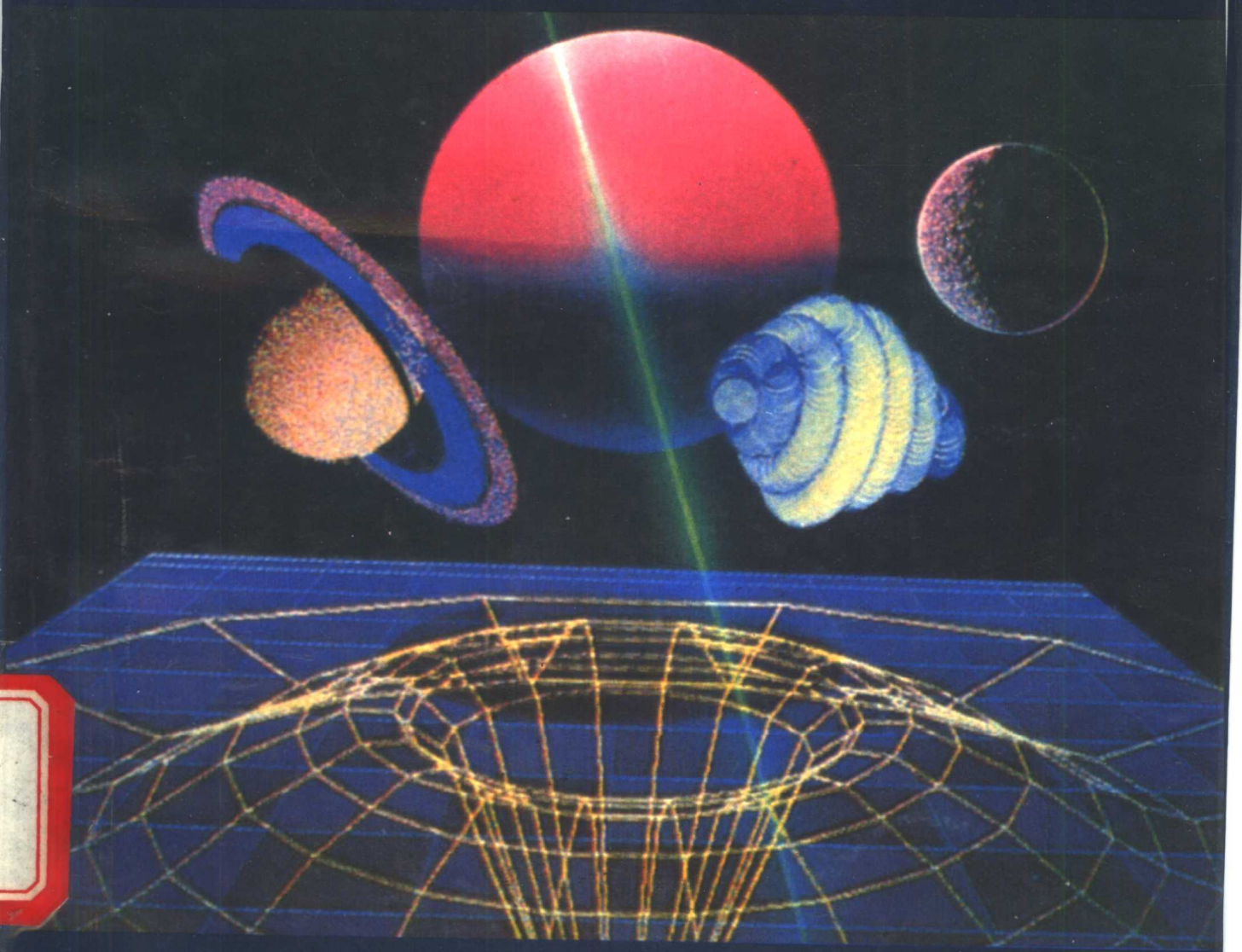
计算机软件工程基础

龚元明　陈为正　刘万春　编著



中国致公出版社

计算机软件工程基础

龚元明　陈为正　刘万春　编著

(京)新登字 196 号

图书在版编目(CIP)数据

计算机软件工程基础/龚元明著. —北京:中国致公出版社,1994.7
ISBN 7—80096—018—8

Ⅰ.计… Ⅱ.龚… Ⅲ.软件工程 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字(94)第 07655 号

中国致公出版社出版发行
北京市太平桥大街 4 号
(邮编:100810)
新华书店经销
北京文化艺术印刷厂印刷

开本:787×1092 1/32 印张:10.75 字数:270 千字
1994 年 8 月第 1 版 1994 年 8 月第 1 次印刷
印数:5 000 册 定价:12.80 元

前　言

建立一个大的计算机软件系统完全不同于写一个小的计算机程序。一个小程序不会过于复杂，可以由一个程序员来编写，所有的设计和调试的细节都在他们的头脑中，形成的一些有关说明往往是非规范的或不太齐全的。反之，一个大的软件系统的复杂程度往往不是一个人所能理解的，更不用说一个人能完全掌握起来。因此大的软件系统的开发必须有阶段、有组织地进行，并采用一些必要的技术和工具，更要做好项目开发的组织和管理工作。把一个软件系统的开发看成一项工程，就是所谓的软件工程。

软件工程这个名字最早是六十年代末期，一次讨论软件危机的会议上提出的。当时第三代计算机刚刚诞生，由于它们功能强、体积小、成本低，很快渗透到各个应用领域中。实践的结果对建立大的、复杂的软件系统提出了新的要求，促进了各种软件技术的迅速发展。

软件工程主要研究计算机软件开发的方法、工具、标准和规范。目前国内有关软件工程的定义不尽相同，各有所侧重，但是它们的共同点都是在系统开发中采用工程设计的原则来研制大、中型的计算机软件系统。

作者根据10年来软件工程的教学和科研工作，编写了这本既可作为高等学校软件工程课程的教材，也适用于有初步程序设计经验的读者阅读，并可作为软件实际工作者的参考书。它以目前软件界最为流行、也较实用的结构化方法为主要线索，同时也介绍了目前软件工程的新技术。

第一章概论，重点是软件生命周期的概念，它是本书的一个大纲，第二章说明了软件质量的评价，从第三章到第八章详细论述了软件生命周期各个阶段的任务、目标、开发工具和方法，第九章对软件工程新技术作了介绍。由于本书是一门实践性较强的学科，附录中给出了一个信息管理系统开发的全过程。

由于作者水平有限，经验不足，书中难免还存在一些缺点和错误，殷切期望广大读者批评指正。

内 容 提 要

本书结合作者多年教学和科研工作，介绍软件工程的基本概念、原则及软件典型开发技术。它以目前软件界最为流行也较为实用的结构化方法为主，叙述软件危机与软件工程、软件质量评价、需求工程和可行性分析、系统分析、系统设计、编程及测试、软件维护等内容，并附有实例和习题。同时本书还概述了软件工程的新技术——原型技术和面向对象的方法。

本书深入浅出，循序渐进可供软件开发人员阅读，也可作为高等学校计算机专业“软件工程”的教材或教学参考书。

目　录

第一章

概　论

在生产技术发展历史上，工程学科的进步一直是产业发展的巨大推动力。尤其1946年电子计算机的诞生，标志着人类正由工业化社会进入信息化社会，以计算机产业和计算机应用服务业为支柱的信息工业成为这个社会的主要基础之一。

60年代以来，计算机的应用愈加广泛，几乎涉及到社会生活的各个方面。工程学科中的一个新成员——软件工程，对软件产业的形成和发展起着决定性的推动作用。本章对计算机软件工程的形成、软件生命周期和软件开发的方法和工具等概念作简要的介绍，以使读者对软件工程有最起码的理解。

第一节　软件工程的形成

计算机软件是计算机应用的灵魂，软件相对于硬件的费用比例不断提高，图1－1为硬件和软件费用比例的变化。由图可见，50年代软件费用比例低于总费用的20%，70年代达60%，而80年代已达85%。

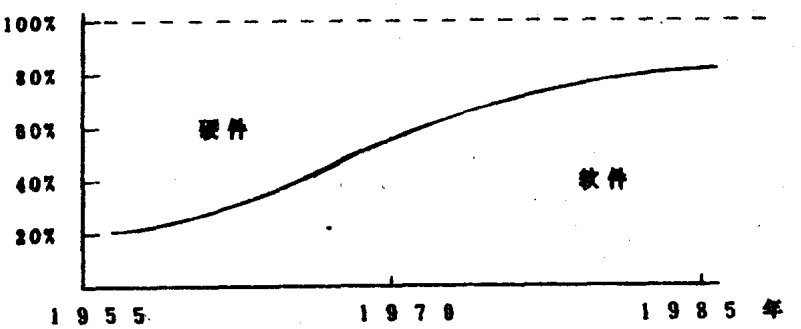


图1－1　计算机软件费用比例上升

硬件价格的下降表明计算机的广泛使用，对软件的需求也迅速上升。软件的开发是一个思考过程，难以管理，开发人员往往按各自的爱好和习惯工作，又往往以手工方式进行，缺乏统一的标准，对所需的经费和时间也难以预计。人们在开发大型软件系统时遇到了很大困难，有的系统甚至失败。

IBM公司1963－1966年开发OS/360系统时所遭受的挫折可作例证。该系统耗资几千万美元，花费5000多人年，用了几年才交付使用，但投入运行后其结果令人沮丧，人们在程序中发现的错误就达2000多个。该系统负责人F. D. Brooks曾生动地描述了开发过程中的困难和混乱：

"…像巨兽在泥潭中作垂死挣扎，挣扎得越猛，泥浆就沾得越多，最后没有一个野兽能逃脱淹没在泥潭中的命运…程序设计就像是这样一个泥潭…一批批程序员在泥潭中挣扎…没有人料到问题竟会这样棘手。"

软件开发的高成本及产品低质量的尖锐矛盾，就是所谓的软件危机(Software Crisis)。

一 软件危机的表现

1 计算机应用系统越来越大，软件复杂程度越来越高。以程序员各自为阵的小生产者手工业方式(或小作坊式的生产)已无法完成大型软件系统的开发。

2 软件发展的历史较短，没有大量的历史数据和模型作为系统开发的依据，导致进度和费用估计的粗略，整个软件缺乏有效的管理措施。

3 产品质量低劣，出错率高，可靠性差，无法满足用户的真正需要。

4 应用系统的需求量日益增长，整个社会对软件人员的需求急剧上升，而软件人员奇缺，供求关系失调。即使在软件产业最发达的美国，软件需求量每年大约递增12%，而软件人员的劳动生产率每年只增加4%。

5 已开发软件的维护最为麻烦。软件维护费用占整个软件生存周期总费用的70—80%。随着时间的推移，系统可靠性的下降，以修补的办法来改进现有系统，难以满足用户的要求。

二 软件危机的原因

造成软件危机的原因是多方面的，主要有以下两点：

1 软件本身的特点　在计算机系统中软件是逻辑部件，而硬件是物理部件。物理部件允许有一定误差，如100个键盘即使5个不合格，还有95个可以用。逻辑部件不允许有一点误差。硬件用坏了可以换，软件不会旧也不会坏。软件的问题往往在软件开发时引入，测试时没有发现错误，软件的维护通常包括对设计的更改和修正。软件生产是一个思维过程，它的进度和质量是不可见的。

2 人为的因素　软件是逻辑产品。软件生产集中于开发而不是制造，起决定作用的是人的智能的高效率发挥，并且与管理人员的技术水平和工作作风有极大关系。

三 问题

软件危机的出现，向人们提出了如下问题：如何进行软件开发？如何维护现有的软件？如何适应社会对软件日益增长的需求？为解决这些问题，1968年，在北大西洋组织的一次学术会议上，有人首次提出"软件工程"(Software Engineering)的概念，并对软件工程化的技术进行了讨论。

假如一个人在一天内能完成100行代码的设计，按此效率，一个10000行代码的程序是否一个人花100天或一百个人一天就能完成呢？回答是否定的。程序的规模由100行增加到10000行，程序的复杂性的程度却远远不止增加一百倍。尤其是当许多人共同设计一个由许多模块组成的大型系统时，人和人之间必须准确地进行协商讨论，软件的开发更为复杂。另外，从产品使用情况来看，大型系统开发耗费了大量的人力物力，人们一般不会轻易抛弃，而是在旧程序的基础上一再修改，希望延长它的使用期限，随着时间的推移会出现"多个版本"。

Parnas认真分析了开发大型软件和编制小型程序之间的差别，把软件工程定义为多版本软件的多人结构(Multi—person Construction of multi—version Software)。这个定义抓住了软件工程的实质，以及程序设计和软件工程间最明显的不同。一个程序员写一个完整的程序，一个软件工程师写一个软件部件，将同其它软件工程师写的软件部件结合，构成完整的软件系

统。程序设计基本上是一个人的活动，而软件工程涉及一组人的活动。

作为一门学科，软件工程研究如何应用一些科学理论和工程技术指导大型软件的开发，以较低的成本研制出具有较高质量的软件。软件工程将促进计算机推广应用的步伐，直接或间接地产生巨大的经济效益和社会效益。

第二节　软件生命周期

软件和软件生命周期（或称为生存期）是软件工程中两个重要的概念。

过去，人们一般认为所谓软件就是指程序，所谓开发软件就是编写程序，这种理解对于大型软件系统是不合适的。

软件生命周期就是从提出软件产品的开发开始，直到该软件产品被淘汰的全过程。研究软件生命周期是为了更科学、更有效地组织和管理软件的生产，使产品更可靠、更经济。

一个软件生命周期可以分为若干个彼此既有联系又有区别的阶段。每个阶段中的工作都以上一阶段工作的结果为依据，同时又为下一阶段的工作提供前提。

我们的基本思想是尽早地发现并纠正错误。工作中造成的差错越是发生在生命周期的后阶段，造成的损失就越大，纠正错误所花费的代价也越大。因此，上一阶段的工作没有做好，决不要轻率地进入下一阶段，这样将有助于提高软件质量和节省开发成本。

关于软件生命周期的划分方法，至今尚未有统一的认识，但各种划分方法大同小异。这里我们把软件生命周期划分为 6 个阶段。

一　问题定义和可行性分析

一个大的软件系统的研制是从用户提出要求开始的。当用户感到计算机的运行存在问题，或满足不了要求时，就可能提出改进原有的系统或者开发新的软件。设计人员必须弄清楚用户存在的问题及提出的要求，作初步的调查研究，称之为问题定义。

接着，应该进行可行性分析。在人力、财力、物力和时间各方面的允许范围内，分析有无必要和可能开发这个软件系统，并进行效益和成本的估算。可行性分析的内容是：

1　确定所研究系统的范围，即系统的边界。系统边界之外的内容，不予考虑。

2　确定目前计算机系统存在哪些问题，在硬件、软件上有哪些缺陷，哪些地方满足不了用户的要求。

3　确定新的系统要达到什么样的要求，主要目标是什么。

4　在用户的经费、时间允许的范围内提出几个方案，以便选择。并对每一种方案的优缺点进行比较，作出评价。

5　开发软件的初步计划。

6　同生产研制部门和用户进行协商，使双方对以上各点取得一致的看法。

该阶段应写出可行性研究报告。

二　软件需求分析

以可行性研究为出发点，对系统进行较为详细的分析。使用数据流程图（DFD）为主要工具对系统进行逻辑设计，解决一个 What to do? 的问题，从而明确系统的目标。

这一阶段的产品称为功能规格书（functional specification），它是新系统的逻辑要求，这些要求包括处理数据的方式和所要求的数据库。然后，对功能规格书进行讨论和审核，并从功能

规格书来分析系统的成本和收益。

三 软件系统设计

这个设计是解决如何去实现功能规格书的问题，即是一个 How to do ? 的问题。它包括两个阶段：

(一)结构式系统设计

根据实际的技术条件和经济条件来确定系统的实施方案，即物理模型。设计师的目标是在保证实现逻辑要求的前提下，尽可能提高软件的可靠性、质量、效率和可变更性。它以结构图为工具，进行从顶向下的设计。这种设计是总体设计，好像是一幢大楼的总体安排。

(二)结构式程序设计

对软件系统中各个模块进行设计。模块的划分应该保证模块的凝聚度(指模块本身内容的内在联系)要高，而模块之间的耦合要小。这种模块设计好像是一幢大楼中每个房间的设计。系统设计的结果是产生结构规格书。

四 代码编写

使用适当的高级语言或者汇编语言编制程序，并对如何执行这个系统有一个合适的策略。例如，人机对话方式可以根据用户的情况采用菜单式、回答式、命令式。对表格的设计要符合用户的要求，并在安排上要简明、扼要。主要信息要突出、醒目。

五 软件测试

及时发现软件中的错误，并设法排除。

六 运行和维护

设计出来的软件系统一旦通过验收交付使用，就进入运行和维护阶段。

在经过一段较长时间的运行以后，用户会提出新的要求，希望对现行的软件系统作出大的改进或者开发新的软件系统，这表明原有软件系统的废弃，新的研制工作开始。以上从用户提出要求直到软件系统的开发、运行和废弃的整个过程，非常类似于人的生命过程，我们称之为软件生命周期(The Sortware Life Cycle)。

图 1－2 表明了软件生命周期，图 1－3 表明了人的生命周期。

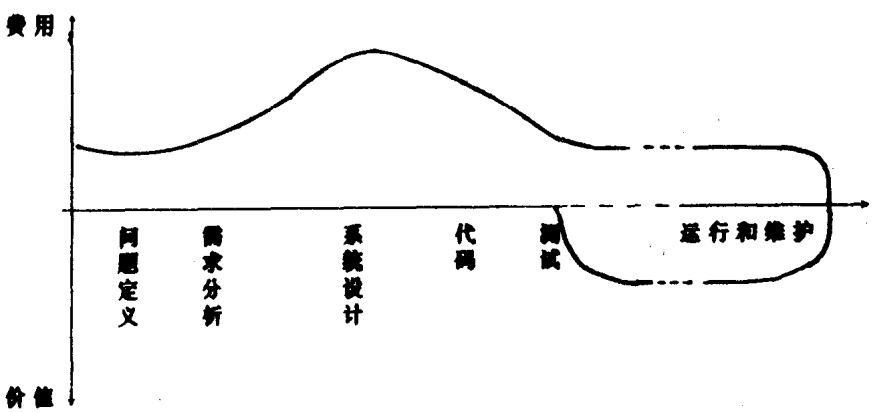


图 1－2 软件生命周期

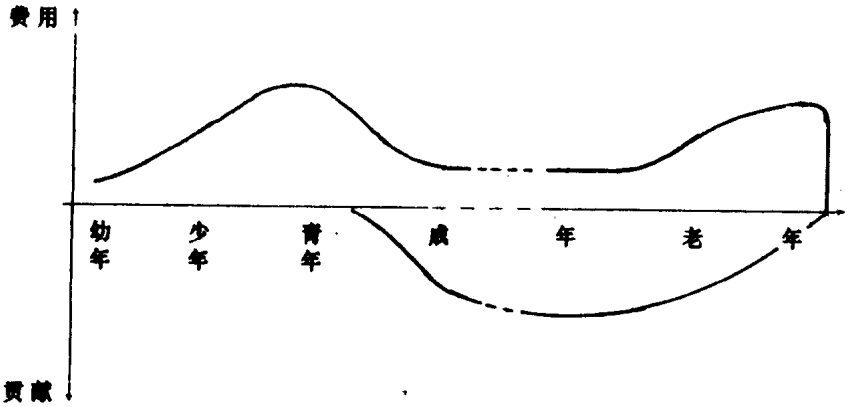


图 1－3　人的生命周期

综上所述，软件工程涉及软件需求分析、设计、编码、测试和维护的原理、方法及工具，是一门研究和应用的学科。它要求管理和技术并重，编写程序仅仅是软件生命周期很小的一部分，它不等效于软件的开发。

表 1－1 给出了软件开发各个阶段所需费用占用的比例。

表 1－1　软件开发的成本

	研制过程的成本	总的系统成本
需求分析	15%	6%
软件设计	20%	7%
代码编写	20%	7%
软件测试	45%	15%
运行和维护	—	65%

由表 1－1 可知，系统总成本的 80%用于调试和维护。软件的错误发现得越晚，校正的代价就越大。

据统计，假定在可行性分析时发现错误，校正它所需的代价为 1；在系统分析阶段校正所需代价为 5，在系统设计阶段校正所需代价为 25；而在调试阶段发现，校正的代价为 125。因此应该采用系统工程的方法，进行软件的开发，厂家和用户密切联系，尽早发现并解决问题，尽量不让错误留到后期，这是节省成本，避免大的返工的重要想法，这样做可以把损失减少到最小。

第三节　软件开发各阶段的文档

软件是由程序和文档组成的，两者缺一不可。由于没有程序就不能使计算机运转起来，人们往往重视程序而忽视文档。

在信息社会中，信息系统的环境常常变化，用户的需求也必须改变，随之就要修改程序。为了提高软件的利用价值，延长软件的寿命，系统的修改也是不可避免的。

如果原编制人员已调动工作，或正从事新的系统开发，让其他人员来修改原系统，没有文档是极其困难的。即使对编制人员本人而言，天长日久，也会忘记，因此文档对其本人也是不可

缺少的。目前软件的鉴定中，技术文档是否齐全是一个重要的指标。

文档的种类很多，主要有以下 3 种：

(一)规格书

用来在开发各阶段之间传递信息，由上一阶段向下一阶段发出指示。

规格书有系统需求规格书、功能规格书、详细设计规格书、程序规格书等等。例如，功能规格书明确表示出该系统应完成哪些功能？各个功能内容是什么？所有功能之间的关系如何？至于该功能如何实现，内部结构如何，可以不涉及。功能规格书是功能设计阶段的产物，也可作为详细设计阶段的输入。可见，这类文档的目的是把开发的各阶段有机地连接起来，高效率地产生可靠的软件。

(二)说明书

说明书是系统的重要输出产品，它与程序一起构成软件。它向用户传递有关程序的信息。例如，系统使用手册或用户手册(针对用户)、程序维护手册(针对系统维护人员)和系统操作说明书(针对系统操作员)等等。

(三)分析报告书

是记录系统开发经过的文档。在软件开发中往往要进行讨论和评价。对于庞大的系统，有时会忘记是否已议论、评价过，而产生重复评议或遗忘。记录系统开发的文档可以防止重复评议的浪费和由于疏忽忘记评议而造成的错误。这种文档不仅可以在以后开发同一类系统时作参考，而且对于系统的推广也起到作用。这类文档往往称为系统分析书或系统分析报告等。

在软件开发的各阶段，文档作业情况见图 1—4·

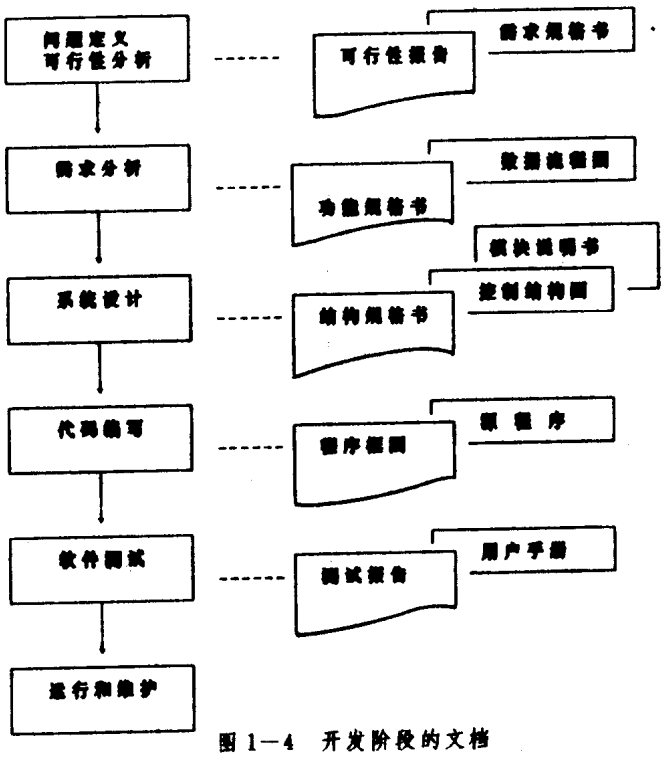


图 1—4　开发阶段的文档

以前，一般系统开发中，系统设计员进行系统设计，设计结束时以自由方式归纳出一份程序规格书交给程序员。程序员进行程序设计、编码调试之后，再作出没有标准格式的程序维护文档。这种自由方式书写的文档理解困难，同时没有标准格式的文本也不能作为产品投入市

场。因此，文档必须标准化容易理解。文档是软件的重要组成部分，开发很费工时，如何得到高质量的文档，怎样对文档进行标准化的研究，也是软件工程的一个领域。

第四节　模糊的观念和正确的认识

为了按期获取高质量的软件产品，必须贯彻软件工程方法。通常存在的一些模糊或错误的观念如下：

(一)只要对总的目标有一般的描述就可以编程，细节可以以后再补充。

人们通常忽视前期工作，急于进入编程阶段。然而，初期定义不准确是软件开发失败的主要原因。一个关于功能、性能、接口、设计约束和有效性标准的正式而详细的描述是不可缺少的，这些只能在用户和软件开发人员之间进行全面充分的讨论之后才能确定。没有进行详细的调查研究和方案论证就仓促上马必然会出现问题，从而引起用户的不满。有些项目钱花了，进度却一拖再拖，有些项目鉴定结束，文章发表，但实用性很差，只好束之高阁。要满足用户的要求必须充分与用户合作，作为管理者更要重视这一点。

(二)写出了程序并且能运行就算完成任务

软件开发不等于编写程序，要按照软件生命周期规定的各个阶段进行，每个阶段都必须有详细的文档，要保证程序的可读性和可更改性。

(三)软件需求不断变化，软件灵活性大，修改容易

确实，对软件的要求经常有变化，但修改所需的费用在不同阶段差别很大(见表 1—1)，越到后期所花销费用越大，如果前期工程仓促上阵，不考虑后期修改的工作量和费用，总费用将大大超过预算。

(四)评审是多余的，太费时间

评审是目前唯一可以进行软件管理控制和技术控制的手段。在软件生命周期中各个阶段应坚持严肃认真的评审工作，否则无法管理控制软件的质量。

(五)进度落后了，只要加人就可以赶上去

软件开发与机械制造、建筑工程不同，不是单纯的机械作业或手工作业过程，而是知识密集的逻辑过程。在一个拖了进度的软件项目中增加人员往往会使它更加拖延，当然有计划、有层次地增加经过严格训练、有经验和易于合作的人员会对工作有促进作用。

(六)软件工作只到软件交付使用为止

实际使用中维护的工作量是很大的，预计今后这个比例还会增加。

产生上述错误认识的原因是多方面的，有些是计算机发展本身带来的，有些是人们的习惯势力影响，但更主要是缺乏对软件工程的教育，管理人员缺乏这方面的知识。

80 年代以来，各国都把软件工程作为计算机科学与工程领域的重要研究领域。我国软件的发展虽然比国外落后多年，但我们可以借鉴国外的经验，少走弯路，使软件工程这一领域获得较快的发展。

第五节　软件工程与其它计算机领域的关系

软件工程是计算机科学的重要领域，它与许多其它计算机科学领域之间有着协同关系，这些领域影响到或受到软件工程的影响。

一 与程序设计语言的关系

软件工程对程序设计语言有明显的影响。在软件开发中程序设计语言作为编程工具使用，软件工程所开发软件的目标也影响程序设计语言的采用和发展。例如，Ada 程序设计语言支持“程序包”的开发，把它作为一个组件使用，有可能像硬件装配那样，选择一些程序包来装配软件。

反之，软件工程也受到程序设计语言的影响。要尽可能使用逻辑严密、机器易于处理的语言（类似程序设计语言）来描述软件的需求和设计。

多年来对程序设计语言形式化和自动化的研究技术也应用到软件工程中，进行格式化说明和自动软件生成的发展是程序设计语言对软件工程的另一种影响。

二 与操作系统的关系

操作系统对软件工程的影响强烈，它是实际建造的大型软件系统之一。操作系统中采用的一些技术，虚拟计算机、分层抽象化、策略和技巧分开等都可以应用于任何大型软件系统，尤其是把策略和技巧分开与软件工程中把说明（What）和执行（How）分开的原则相对应。

同样，软件工程也影响到操作系统，例如 UNIX 操作系统企图规定一个软件开发的环境，这个环境中的工具支持了软件配置的管理。

三 与数据库的关系

数据库技术是另一个大型软件开发技术，它对软件工程的最重要影响是数据独立性的概念，它允许使用数据而不必关心数据的内部表示方法，这也是一种说明同执行相分开的例子。

软件工程也影响到数据库技术的发展，传统的数据库技术处理软件工程中提出的一些问题显得无能为力，例如存贮大结构目标（源程序或用户菜单）、大型非结构目标（目标码和执行码）、同一目标的不同版本等用传统数据库技术很难解决。另外，传统数据库支持短的事务处理（transactions），而软件工程中往往要解决很长的事务处理（例如，多模块的大编辑系统、测试大的程序等）。目前，数据库领域的发展已考虑到上述问题。

四 与人工智能的关系

人工智能是另一个影响软件工程的领域，所建软件系统通常是复杂的系统。从某种意义上说，它们不同于一般的软件系统，往往涉及探索性、不精确性。这些概念同传统的软件工程中要求精确定义用户需求的做法相反。但人工智能中研究的一些新技术，例如不确定性研究、逻辑型（logic）语言的使用、自然语言的接口等都开始影响到软件工程用户接口设计。

软件工程技术也采用在最新的人工智能系统。例如，专家系统中对已知的事实（facts）和系统处理事实所用的规则（rules）划分得非常清楚，这种划分使得建造和商品化专家系统工具成为可能，用户能使用专家系统工具来建造专家系统。

人工智能和软件工程交叉发展，例如，程序设计助手成为程序设计人员的好帮手，它注重通常的编程语法或系统需求，可以用来进行软件测试。

五 与计算机理论的关系

计算机理论发展了很多模型，成为软件工程的有用工具。例如，有限状态机可用作软件说明的技术基础，也可以作为软件设计和构造的模型。

软件工程也影响计算机理论的发展。软件工程需求推动了代数说明和抽象数据理论。在规范化领域方面，软件工程更多地集中在非一阶逻辑理论上（例如，暂态逻辑）。软件工程师同计算机理论研究人员不同，他们对一个理论的有效性和可理解性更感兴趣，暂态逻辑与一阶逻

辑相比，其方式更为紧凑和自然。

第六节　软件开发方法和工具

软件工程的最终目的是以较少的投资获取较高质量的软件产品，即高产优质。它开发的规范化和自动化涉及到软件开发的方法和工具。

一　软件开发方法

几十年来，程序设计是一种个体手工艺方式的劳动，每个程序员都可以按自己的方式编写程序，没有什么规范需要遵循。当软件进入大规模生产时，则要集体劳动，面临的第一个问题是规范化。

为使软件研制走上工程化的轨道，我们必须寻找一些标准的规程，作为指导和约束，使开发人员按一定的规范来理解和处理问题，这是使开发获得成功的关键所在。

软件开发方法(Software Development Methods)是一种标准规程，但过程相当复杂，涉及因素很多，所以软件方法也有一定的灵活性和试探性。一般来讲，一个软件方法往往规定了明确的工作步骤、具体的描述方式以及确定的评价标准。

(一)明确的工作步骤

一个软件系统的开发涉及很多问题，正确的方法是把复杂问题分解为简单问题，按先后次序一步步去做，每一步集中精力解决一个问题。软件开发方法提出了处理问题的基本步骤，包括每一步的目标、工作成果及应注意的问题等。

(二)具体的描述方式

工程化生产必须强调文档的重要性，每一步工作成果一定要以书面方式记录下来，以保证开发人员之间有效地交流，有助于维护工作的顺利进行。软件开发方法规定了描述软件产品的格式，包括每一步应产生什么文档，文档的内容及格式等。

(三)确定的评价标准

对于同一个问题，要选取最佳方案。有些软件开发方法提出了确定的评价标准，可以指导人们对各个具体方案进行评价，从中选取较好的方案。

近十几年来，软件工作者陆续研究总结出各种软件开发方法。70 年代初出现了编写程序的结构化程序设计方法；70 年代中期人们认识到编程仅仅是软件开发的一个环节，合理建立软件结构比编写程序更为重要，研究重点转移到设计阶段，提出了结构化设计和 Jackson 方法；70 年代后期，人们进一步意识到前期工作的重要性，研究重点又转移到分析阶段，提出了结构化分析，SADT(结构化分析和设计技术)、SREM(软件需求工程方法)等方法。目前，用什么方法来分析和说明用户的要求，用什么方法对软件进行测试和维护仍是软件工作者感兴趣的问题。

各种软件方法的适用范围不一定相同，例如 SREM 方法适用于实时控制，结构化设计和 Jackson 方法适用于数据处理。各种方法的风格也互不相同，有些较为抽象，有些较为具体。

本书以目前流行的结构化方法(包括结构化分析、结构化设计、结构化程序设计等)为主来介绍软件开发方法。但它们还不是尽善尽美的，在不同软件系统中对软件开发的方法要求也不同，开发人员必须结合具体情况灵活使用。

二 软件开发工具

软件开发是智力密集的工作，为提高生产效率和质量，人们很自然地希望实现开发过程的自动化。实践证明，用计算机来完成数据处理工作很有效。软件开发本身也可以看作是一种数据处理工作，其中机械性的、规律性的工作也可以用计算机来完成。

软件开发工具(Software Development Tools)就是指软件开发、维护和分析中使用的程序系统。例如，编辑程序、编译程序、排错程序、跟踪程序等都是用于程序编写及测试的工具。

近年来，人们又为设计阶段和分析阶段研制了一些工具，例如 PSL/PSA(问题说明语言/问题说明分析程序)等。众多的软件工具组成了工具箱，在软件开发的各个阶段，人们可以根据不同的需要选择合适的工具使用。但是软件开发的过程是复杂的，很多工作需要人的创造能力，全部自动化是不可能的，计算机只能起到辅助的作用。

软件开发方法和工具间有着密切的联系，方法是主导，工具是辅助。软件方法提出了明确的工作步骤和标准的文档格式，是设计软件工具的基础，而工具的实现又将促进方法的发展。

目前人们热心于研究软件工程环境，它是软件开发方法和工具的结合。人们希望研究出一套系统的软件方法、一组成套的软件工具，再加上一个由计算机管理的文档库，为开发人员提供了一个能覆盖整个软件生命周期的良好工作环境。

习题一

1 什么叫软件危机？如何摆脱软件危机？
2 什么是软件生命周期？它分为哪几个阶段？
3 试论述开发阶段文档的重要性。
4 在软件开发中有哪些不正确的认识？
5 软件工程和其它计算机领域有何关系？
6 为什么要使用软件开发的方法和工具？