

软件工程名著译述

论大型软件的开发

[美] 小弗雷德里克·P·布鲁克斯著

徐仁佐 方学芬 吕天玲译



武汉大学出版社

论大型软件的开发

[美] 小弗雷德里克·P.布鲁克斯 著

徐仁佐 方学芬 吕天玲 译

武汉大学出版社

1986

内 容 提 要

本书是论述软件工程的名著，是对作者本人在主持IBM-360操作系统设计过程中的经验教训所作的全面、系统的总结。作者始终根据软件生命周期各阶段的特点，阐述了进度管理、人员管理和文档管理中一系列带根本性的问题；讨论了影响软件生产率的诸因素并提出了不少提高软件生产率的方法。

作者被誉为美国“IBM-360之父”。书中所阐述的问题既有理论的探讨，又有实际背景的描绘，语言流畅，叙述生动。

本书可作为高等院校计算机专业研究生、本科生的教学参考书，也是从事软件开发和管理的各类人员颇有教益的读物。

论大型软件的开发

〔美〕小弗雷德里克·P.布鲁克斯 著
徐仁佐 学芬 吕无冷 译

武汉大学出版社出版

(武昌 珞珈山)

新华书店湖北发行所发行 武汉大学出版社印刷总厂印装

*

787×1092毫米 1/32 6印张 125千字

1986年10月第一版 1986年10月第一次印刷

印数：1—1,400

统一书号：13279·31 定价：1.05元

献 给

使我在IBM公司的岁月里过得特
别充实的两位领导人：

Thomas J. Watson, Jr. ,
在他的公司里仍到处洋溢着他关怀人
的深切之情；和

Bob O. Evans,
他的大胆领导使工作充满进取精神。

作者简介

小弗雷德里克·P. 布鲁克斯是北卡罗莱纳大学计算机科学系的教授，写作本书时他正兼该系系主任。在IBM/360系统的开发期间，及在随后的IBM/360操作系统软件项目的设计期间，他都是项目的主持人，因此以“IBM/360系统之父”而闻名。早期，他曾 是I BM Stretch和Harvest计算机的设计师。

布鲁克斯博士在北卡罗莱纳大学任教期间，参与了创建和指导三校计算中心和北卡罗莱纳教育计算服务中心的工作。他发表了《自动数据处理》、《自动数据处理的360/系统文本》，以及其它几部书的若干章节。

译 者 的 话

本书系根据阿迪生-韦斯利出版公司 (Addison-Wesley Publishing Company) 出版的小布鲁克斯 (F.P. Brooks, Jr.) 所著“软件开发的神话” (The Mythical Man-Month: Essays on Software Engineering) 1975年版译出。

本书第一至五、十五章由徐仁佐同志翻译, 第六至十章由方学芬同志翻译, 第十一至十四章由吕天玲同志翻译。原文中个别叙述上的错误, 已根据我们的理解作了更正, 行文中不再一一指出。全部译稿由徐仁佐同志作了文字上的统一处理。

在本书的翻译过程中, 得到何克清同志的大力支持, 并由他为本书撰写了译序。代大为同志审阅了本书的全部译稿, 并提出不少修改意见。

武汉大学经济管理系许俊千教授于百忙中审阅了译稿的部份章节, 并提出不少宝贵的意见。特此致谢。

由于我们的水平有限, 错误与不当之处教请读者指正。

译 者

1984年9月于珞珈山

译 序

有人说,《软件开发的神话》一书的发行量反映了一个国家、一个地区软件工程发展的水平。仔细想起来,还是十分在理的。

本书是IBM/360操作系统软件开发经验的总结和结晶。它神奇般地提出了软件开发的一系列重要的问题与课题,生动地叙述了软件开发的各种困难,明确地指出了各种方法和解决问题的途径,是一本学习和实践计算机软件工程的极其重要的指导书。

结合我国当前软件开发的实际状况一读,发人深省,颇有教益。可以相信,软件工程的开发、研究人员将十分喜爱她。

武汉大学软件工程研究所

何克清

1984、9、13,

序 言

管理一个大型的计算机程序设计项目，如同管理任何一个大型的企业一样，它们在许多方面是很类似的；但是许多程序设计员尚未充分认识到这一点。然而，在另外不少方面，它们又是很不相同的；可是许多专门的管理人员却对其类似的方面估计过高。

关于软件工程这一领域的知识正在日积月累地积聚起来。已经召开了一系列的会议。在美国信息处理学会联合会（AFIPS, American Federation of Information Processing Society）召开的会议上，发表了不少的专著及论文。但是，这并不是说它已经成熟到可以写一本系统的教科书来处理这一领域的内容。然而笔者认为提供这一本质上反映个人观点的小书，似乎还是恰当的。

虽然一开始我的工作是在计算机科学的程序设计方面，可是在一九五六——一九六三年间，我的工作却多半是致力于硬件的结构设计。也正在此期间，人们开发了自主的控制程序和高级程序设计语言的编译系统。一九六四年，我成为OS/360系统的主持人，这时我发现：由于早几年在软件方面的进展，程序设计领域已极大地改观了。

主持OS/360系统的开发，是一次极具教育意义的经历，虽然是一次使人遭受挫折的经历。在当时的的工作小组中，包括有F. M. 特拉泊内尔（F. M. Trapnell），他后来接替

我成了主持人，我为这个小组感到万分的骄傲。OS/360系统具有许多设计和实现方面的优点，并且也成功地获得了广泛的应用。其中的某些想法，最可注意的是与设备无关的输入/输出，以及外部程序库管理技术，现在都是被广泛加以应用的技术革新。现在，该系统相当可靠，具有合理的高效率，并且对于各方面的要求都能应付裕如。

但是，对于该系统所作的努力还不能说是完全成功的。任何一个OS/360的用户很快就会意识到它应当可以好到什么样的程度。设计和实现方面的漏洞特别出现在控制程序中，而语言的编译系统就不是这样。这些漏洞的大多数都要追溯到一九六四——一九六五年的设计期间，因此都应当由我来负责。此外，产品出来得太迟，它所占的存储量比计划的要多，成本是估算的几倍；并且，直到首次发表的版本几经修改之后，该系统才能得以很好地运行。

正如我接手OS/360系统时就允诺下来的那样，于一九六五年，我离开了IBM公司，来到北卡罗莱纳大学执教。此后我就开始着手分析OS/360系统的经验，看看从中可以得到什么样的管理和技术方面的教训。特别地，我想解释在IBM/360系统的硬件开发和OS/360系统的软件开发过程中，所碰到的十分不同的管理经验。本书就是对汤姆·沃特森的众多调查问题（如管理程序设计为什么是困难的等等）的一个太迟的答复。

在这一探索的过程中，我从与R. P. 卡斯(R. P. Case, 1964——1965年的助理主持人)和F. M. 特拉治内尔(1965——1968年的主持人)俩的长谈中，获益匪浅。我还将我的结论与其它项目主持人的作了比较，他们是：麻省理工学院的F. J. 科巴托(F. J. Corbato)，贝尔电话实验室的约

翰·哈尔(John Harr)和V.威索茨基(V. Vyssotsky), 国际计算机有限公司的查尔斯·波特曼(Charles Portman), 苏联科学院西伯利亚计算研究所的A.P.艾晓夫(A. P. Ershov), 以及IBM公司的A.M.皮特拉圣特(A.M. Pietrasanta)。

我自己的结论体现在随后我打算为专业程序设计人员, 专业管理人员, 特别是专门管理程序设计员的人员所写的论文之中。

虽然写成了互不相连的论文集的形式, 但是它们都是围绕着中心议题来写的, 尤其是在第2——第7章, 更是如此。简言之, 我认为大的程序设计项目与小的项目所遇到的管理方面的难题是不相同的, 这主要是由于工作的划分而引起的。我还认为关键在于保持产品本身概念上的完整性。本书前面几章探索了要达到这一统一的困难, 以及做到这一点的方法。后面几章探索软件工程管理的其它方面的问题。

这一领域的文献并不十分丰富, 大都散见于许多的资料之中。因此, 我力图给出的参考文献既能说明特定的问题, 又能引导有兴趣的读者去阅读其它有用的文章。我的许多朋友都阅读过本书的手稿, 他们中的一些人还提出了广泛而有益的意见, 其中那些虽有价值但并不适合于写入正文中去的意见, 我都将它们归并在注释之中。

本书是一本论文集而不是教科书, 因此我将全部的参考文献和注释都放在本书的末尾, 并且奉劝读者在第一次阅读本书时, 忽略它们。

我谨向萨拉·伊丽莎白·摩尔(Sara Elizabeth Moore)小姐, 大卫·瓦格纳(David Wagner)先生以及丽贝卡·伯里斯(Rebecca Burris)夫人致以深切的谢意, 感

谢他们在我准备手稿的过程中所给予的帮助。并且还要感谢约瑟夫·C. 斯隆 (Joseph C. Sloane) 教授对于插图、例证所提供的建议。

小 F. P. 布鲁克斯

1974年10月于北卡罗莱纳大学

目 录

译者的话	(I)
译序	(Ⅲ)
序言	(I)
第1章 沼泽地	(1)
第2章 神奇的人-月	(10)
第3章 外科小组	(26)
第4章 独裁、民主与系统设计	(37)
第5章 第二系统效应	(48)
第6章 传播信息	(55)
第7章 巴别塔为什么半途而废	(65)
第8章 确定目标	(77)
第9章 在装五磅的袋子里装十磅	(85)
第10章 文档生成的前提	(92)
第11章 计划废弃	(99)
第12章 精良的工具	(109)
第13章 整体与局部	(121)
第14章 延迟与对策	(132)
第15章 其它的方面	(140)
跋	(157)
注释与参考文献	(158)
人名索引(英汉对照)	(172)

第1章

沼泽地

搁浅于海滩上的船，就是建于海
边上的一座灯塔。

荷兰谚语



史前的景象中没有哪一幕能像巨兽在沼泽地里作垂死挣扎更使人触目惊心了。人们在想象中看到恐龙，猛犸象，还有剑齿虎，它们为了免遭沼泽地的吞噬而作徒劳的挣扎，挣扎越烈，越往下陷，没有一头巨兽有力量和机智能够逃脱，终于陷入了无底深渊。

过去的十多年中，大系统的程序设计正是这样的：“沼泽地”，许多强有力的巨大“猛兽”已经深陷其中。正在运行的系统也大都有问题，仅少数系统达到最终的目标，如期在预算内完成。无论是大的还是小的、实力坚实的还是实力不足的开发小组，都相继陷入了“泥沼”。看起来，困难并不是某一件事情引起的——任何一只特定的“爪子”都能够拔出泥沼地。但是，一些同时发生、相互影响的因素都使进展越来越慢。所有的人似乎都被问题的顽固性弄得迷惑不解，而要想弄清其本质，却非常困难。然而，要解决这个问题，就先要想方设法去理解它。

因此，还是让我们从了解系统程序设计技巧以及它本身所固有的乐趣和苦恼着手进行研究吧。

程序设计系统的产品

人们不时可在报上读到这样的新闻报导：两个程序员如何在一间经过改建的汽车库里编出了一个重要的程序，他们工作的成果超过了许多大的程序设计小组的成果。而每一个程序员都会相信这故事的，因为他们知道，他们自己也会以比由软件产业组织所报告的每年1,000条语句的速度快得多的速度来编写任何程序。

那么，为什么所有的程序设计产业组织不由那些一对对献身于汽车库的程序员们来取代呢？这我们得看看生产出来的是什么样的产品。

图1.1所示的是一个程序。它本质上是完全的，准备由它的作者在某个系统上来运行，而这个程序也正是为在该系统上运行而开发出来的东西，是个体程序员用以估计生产率的物体。

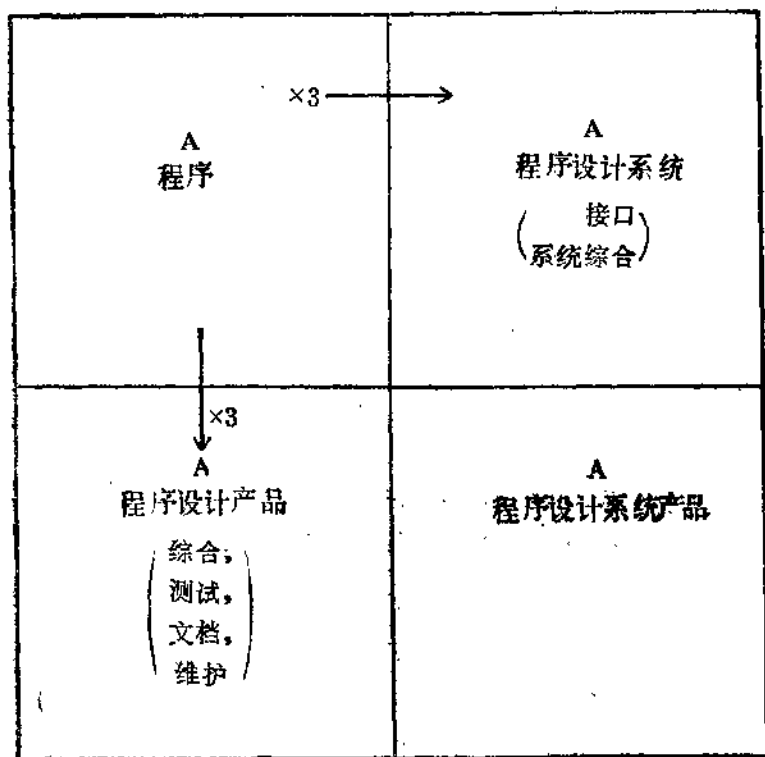


图1.1 程序设计系统产品的展开

这里有两种方式，可以将一个程序转换成更有用的，但成本也更高的程序。这两种方式由图中的分界线表示。

垂直向下移动，越过分界线，一个程序就变成一个程序设计产品。该产品仍是一个程序，它可以由任何一个人来运行、测试、修改和扩充。它对于许多操作环境都是可用的，并且也适合于许多的数据集合。为了变成一个普遍适用的程序设计产品，一个程序必须要用一种普遍能接受的方式来书写。特别地，其输入的范围和输入的形式，在基本的算法所允许的范围内必须尽可能一般化。然后，该程序必须经过彻底的测试，因而，能够确信它是正确的。彻底的测试就意味着：对于能够显示它的输入范围和探测它的边界的众多测试程序的数据和条件，必须充分地作好准备，然后将它投入运行并且记录下来。最后，将一个程序转换成程序设计的产品还要求彻底的文档化，以便于任何人来使用它，修改它，和扩充它。凭经验，我估计一个程序设计产品的成本，至少三倍于具有相同功能的、经过排错的程序的成本。

——向右平行移动，越过分界线，一个程序就变成一个程序设计系统的部件。这个系统是一个在功能上协调、形式上统一的，相互作用、相互影响的许多程序的集合体，因此就构成了为完成一项大型任务的完整体系。为了变成一个程序设计系统的部件，一个程序必须在语法和语义上使每一个输入和输出都与精确定义好的程序之间的接口配合一致。而且，在该程序设计时，就必须考虑到资源——存储空间、输入-输出设备、计算机时间——的分配，以使得它仅仅在事先安排好的资源分配方案的支配下运行。最后，它必须与系统的其他部件一起来加以测试，在测试的过程中还得考虑到所有预期的可能的组合形式。由于对程序进行测试的条件和数据教

目随组合部件数的增长而增长，这一测试一定相当广泛而浩大。由于事先未预料到的、排错的各部分之间的相互影响，产生了一些微妙的错误，所以，这是一件十分费时间的工作。一个程序设计系统部件的成本，至少要三倍于具有同样功能的、一个单个的程序的成本。如果该系统有着更多的部件，则其成本还要有所增加。

在图1.1中的右下角表示的是程序设计系统产品。它从上面所述各方面未说都与一个简单程序不同。它的成本至少要九倍于简单程序的成本。但它是真正有用的程序，是绝大多数系统程序设计的努力所追求的产品。

程序设计的乐趣

为什么程序设计本身充满欢乐？程序设计者可以从中获得什么样的乐趣呢？

首先是进行创造的纯粹的喜悦。正如儿童们醉心于他们做的泥巴玩具一样，成年人也做各种东西，特别是制作他们自己设计出来的东西。我认为这一喜悦一定就是上帝在创造世界时的喜悦的一种反映，这种喜悦在每一片树叶、每一片雪花的千姿百态中表现出来。

第二是在创造对别人有用的事物时的喜悦心情。我们每个人在内心深处都希望别人会使用我们工作的成果，并希望别人都发现这些成果对他们自己是很有帮助的。在这一方面，程序设计系统与孩子的刻着“供爸爸的办公室使用”字样的第一枝自制的粘土铅笔杆基本上是相同的。

第三是构造复杂而又象谜一般的东西并看着它运转时的