

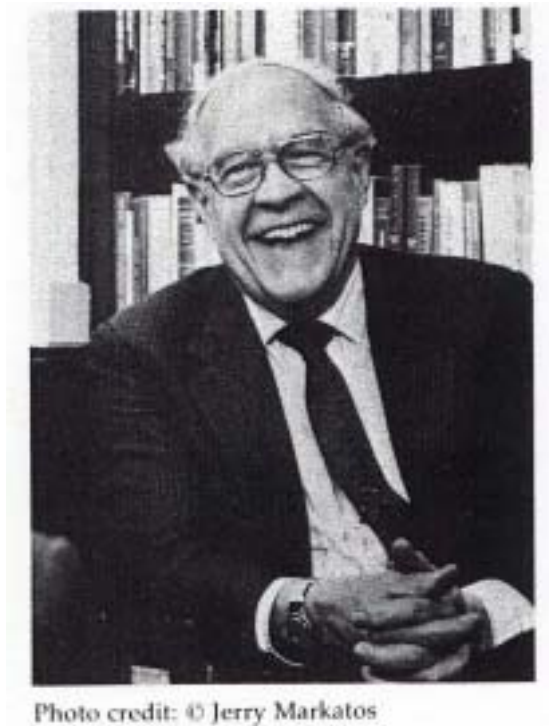


THE MYTHICAL MAN-MONTH

人月神话

FREDERICK P. BROOKS, JR.

翻译：[Adams Wang](#)



关于作者

Frederick P. Brooks, Jr. 是北卡罗来纳大学 Kenan-Flagler 商学院的计算机科学教授，北卡罗来纳大学位于美国北卡罗来纳州的查布尔希尔。Brooks 被认为是“IBM 360 系统之父”，他担任了 360 系统的项目经理，以及 360 操作系统项目设计阶段的经理。凭借在上述项目的杰出贡献，他、Bob Evans 和 Erich Bloch 在 1985 年荣获了美国国家技术奖（National Medal of Technology）。早期，Brooks 曾担任 IBM Stretch 和 Harvest 计算机的体系结构师。

在查布尔希尔，Brooks 博士创立了计算机科学系，并在 1964 至 1984 年期间担任主席。他曾任职于美国国家科技局和国防科学技术委员会。Brooks 目前的教学和研究方向是计算机体系结构、分子模型绘图和虚拟环境。

二十周年纪念版序言 (*Preface to the 20th Anniversary Edition*)

令我惊奇和高兴的是,《人月神话》在 20 年后仍然继续流行,印数超过了 250,000。人们经常问,我在 1975 年提出的观点和建议,哪些是我仍然坚持的,哪些是已经改变观点的,是怎样改变的?尽管我在一些讲座上也分析过这个问题,我还是一直想把它写成文章。

Peter Gordon 现在是 Addison-Wesley 的出版伙伴,他从 1980 年开始和我共事。他非常耐心,对我帮助很大。他建议我们准备一个纪念版本。我们决定不对原版本做任何修订,只是原封不动地重印(除了一些细小的修正),并用更新的思想来扩充它。

第 16 章重印了一篇在 1986 年 IFIPS 会议上的论文《没有银弹:软件工程的根本和次要问题》。这篇文章来自我在国防科学委员会主持军用软件方面研究时的经验。我当时的研究合作者,也是我的执行秘书,Robert L. Patrick 帮助我回想和感受那些做过的软件大项目。1987 年,IEEE 的《计算机》杂志重印了这篇论文,使它传播得更广了。

《没有银弹》被证明是富有煽动性的,它预言十年内没有任何编程技巧能够给软件的生产率带来数量级上的提高。十年只剩下一年了,我的预言看来安全了。《没有银弹》激起了越来越多文字上的剧烈争论,比《人月神话》还要多。因此在第 17 章,我对一些公开的批评作了说明,并更新了在 1986 年提出的观点。

在准备《人月神话》的回顾和更新时,一直进行的软件工程研究和经验已经批评、证实和否定了少数书中断言的观点,也影响了我。剥去辅助的争论和数据后,把那些观点粗略地分类,对我来说很有帮助。我在第 18 章列出这些观点的概要,希望这些单调的陈述能够引来争论和证据,然后得到证实、否定、更新或精炼。

第 19 章是一篇更新的短文。读者应该注意的是,新观点并不象原来的书一样来自我的亲身经历。我在大学里工作,不是在工业界,做的是小规模的项目,不是大项目。自 1986 年以来,我就只是教授软件工程,不再做这方面的研究。我现在的研究领域是虚拟环境及其应用。

在这次回顾的准备过程中,我找了一些正工作在软件工程领域的朋友,征求他们的当

目录 (Contents)

二十周年纪念版序言 (PREFACE TO THE 20 TH ANNIVERSARY EDITION)	I
第一版序言 (PREFACE TO THE FIRST EDITION)	III
目录 (CONTENTS)	V
焦油坑 (THE TAR PIT)	1
编程系统产品.....	1
职业的乐趣.....	3
职业的苦恼.....	4
人月神话 (THE MYTHICAL MAN-MONTH)	6
乐观主义.....	7
人月.....	8
系统测试.....	10
空泛的估算.....	11
重复产生的进度灾难.....	12
外科手术队伍 (THE SURGICAL TEAM)	16
问题.....	16
MILLS 的建议	17
如何运作.....	20
团队的扩建.....	21
贵族专制、民主政治和系统设计 (ARISTOCRACY, DEMOCRACY, AND SYSTEM DESIGN)	22
概念一致性.....	22
获得概念的完整性.....	23
贵族专制统治和民主政治.....	24
在等待时，实现人员应该做什么？	26
画蛇添足 (THE SECOND-SYSTEM EFFECT)	29
结构师的交互准则和机制.....	29
自律——开发第二个系统所带来的后果	30
贯彻执行 (PASSING THE WORD)	33
文档化的规格说明——手册.....	33
形式化定义.....	34
直接整合.....	36
会议和大会.....	36
多重实现.....	38

焦油坑 (*The Tar Pit*)

岸上的船儿，如同海上的灯塔，无法移动。

- 荷兰谚语

Een schip op het strand is een baken in zee.

[A ship on the beach is a lighthouse to the sea.]

- DUTCH PROVERB

史前史中，没有别的场景比巨兽在焦油坑中垂死挣扎的场面更令人震撼。上帝见证着恐龙、猛犸象、剑齿虎在焦油中挣扎。它们挣扎得越是猛烈，焦油纠缠得越紧，没有任何猛兽足够强壮或具有足够的技巧，能够挣脱束缚，它们最后都沉到了坑底。

过去几十年的大型系统开发就犹如这样一个焦油坑，很多大型和强壮的动物在其中剧烈地挣扎。他们中大多数开发出了可运行的系统——不过，其中只有非常少数的项目满足了目标、时间进度和预算的要求。各种团队，大型的和小型的，庞杂的和精干的，一个接一个淹没了焦油坑中。表面上看起来好像没有任何一个单独的问题会导致困难，每个都能被解决，但是当它们相互纠缠和累积在一起的时候，团队的行动就会变得越来越慢。对问题的麻烦程度，每个人似乎都会感到惊讶，并且很难看清问题的本质。不过，如果我们想解决问题，就必须试图先去理解它。

因此，首先让我们来认识一下软件开发这个职业，以及充满在这个职业中的乐趣和苦恼吧。

编程系统产品

报纸上经常会出现这样的新闻，讲述两个程序员如何在经改造的简陋车库中，编出了超过大型团队工作量的重要程序。接着，每个编程人员准备相信这样的神话，因为他知道自

己能以超过产业化团队的 1000 代码行/年的生产率来开发任何程序。

为什么不是所有的产业化队伍都会被这种专注的二人组合所替代？我们必须看一下产出的是什么。

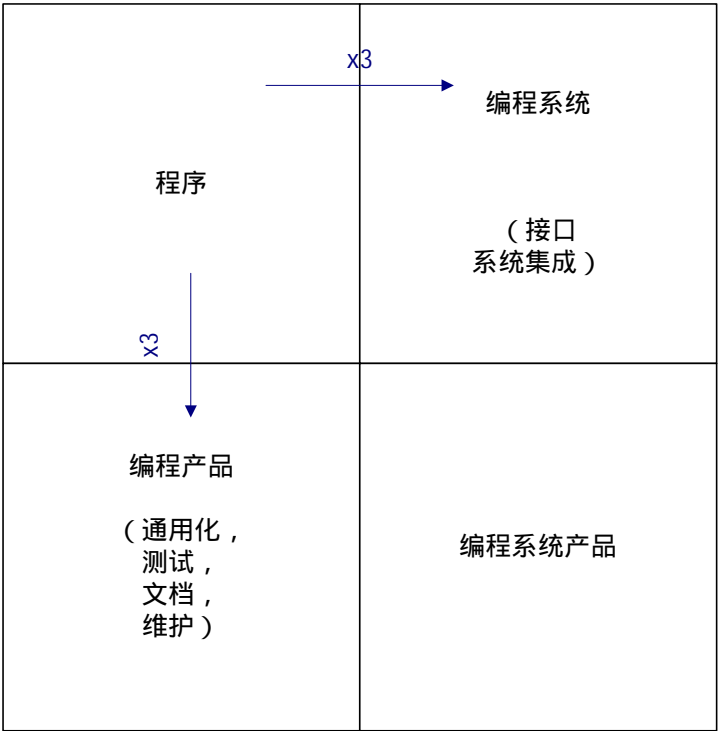


图 1.1：编程系统产品的演进

在图 1.1 的左上部分是程序 (Program)。它本身是完整的，可以由作者在所开发的系统平台上运行。它通常是车库中产出的产品，以及作为单个程序员生产率的评价标准。

有两种途径可以使程序转变成更有用的，但是成本更高的东西，它们表现为图中的边界。

水平边界以下，程序变成编程产品 (Programming Product)。这是可以被任何人运行、测试、修复和扩展的程序。它可以运行在多种操作系统平台上，供多套数据使用。要成为通用的编程产品，程序必须按照普遍认可的风格来编写，特别是输入的范围和形式必须扩展，以适用于所有可以合理使用的基本算法。接着，对程序进行彻底测试，确保它的稳定性和可靠性，使其值得信赖。这就意味着必须准备、运行和记录详尽的测试用例库，用来检查输入的边界和范围。此外，要将程序提升为程序产品，还需要有完备的文档，每个人都可以加以使用、修复和扩展。经验数据表明，相同功能的编程产品的成本，至少是已经过测试的程序的三倍。