

HZ BOOKS
华章教育

计 算 机 科 学 丛 书

PEARSON

《人月神话》作者最新力作
计算机科学大师探究设计原本

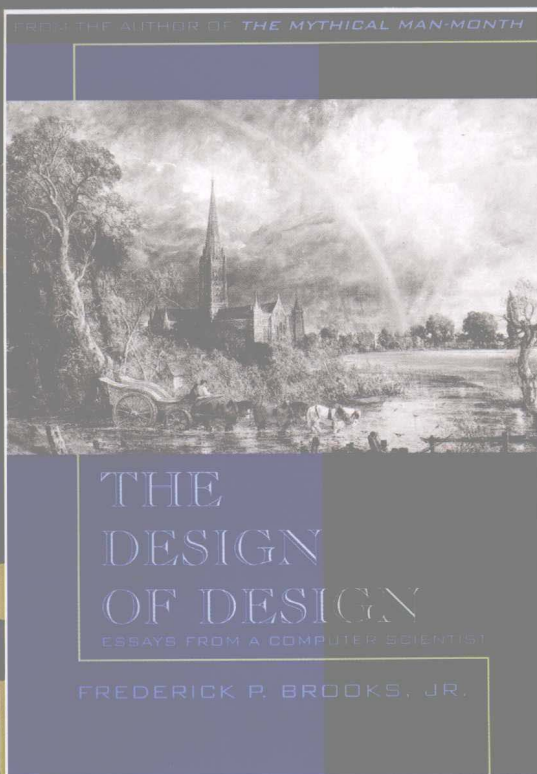
设计原本

计算机科学巨匠Frederick P. Brooks的思考

(美) Frederick P. Brooks, Jr. 著 InfoQ中文站 王海鹏 高博 译

The Design of Design

Essays from a Computer Scientist



机械工业出版社
China Machine Press

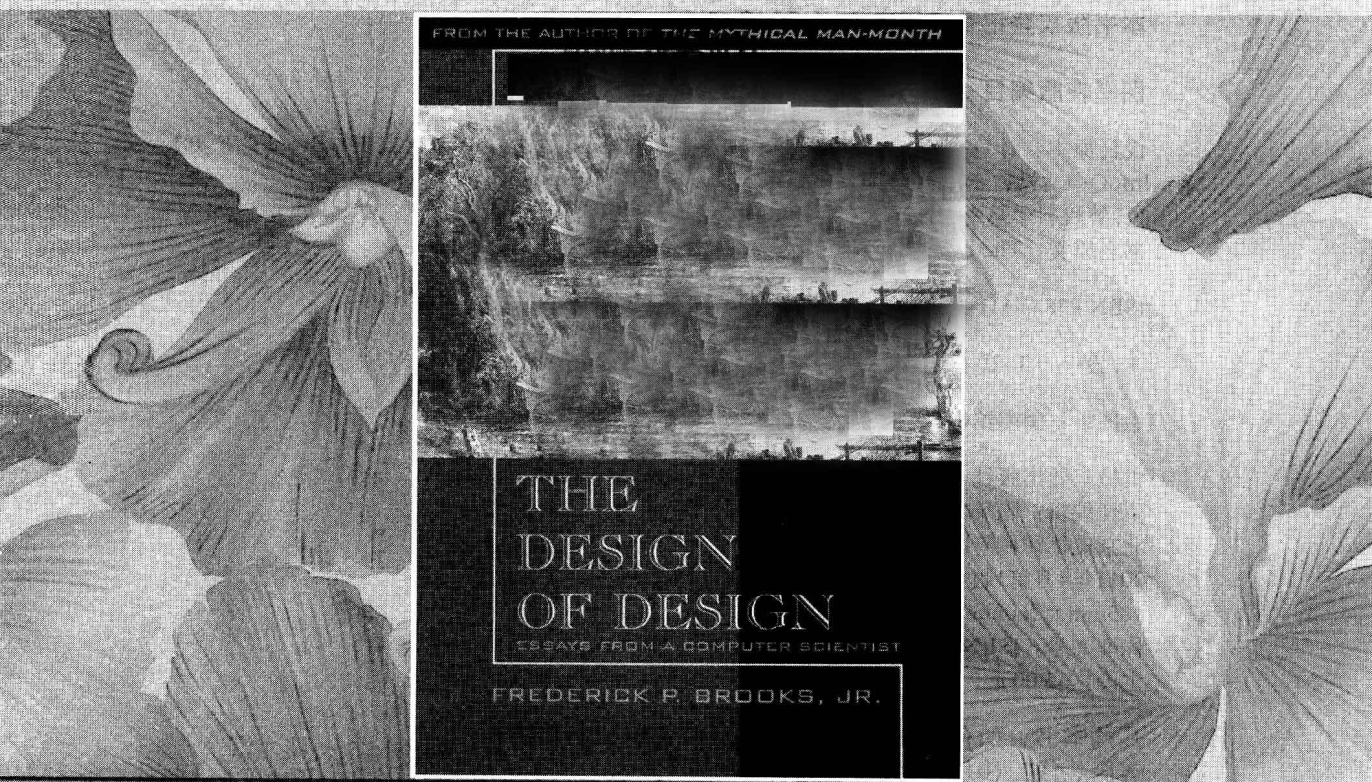
计 算 机 科 学 丛 书

设计原本

计算机科学巨匠Frederick P. Brooks的思考

(美) Frederick P. Brooks, Jr. 著 InfoQ中文站 王海鹏 高博 译

The Design of Design
Essays from a Computer Scientist



机械工业出版社
China Machine Press

这是一部在研究和教学中将设计领域探索心得和实践经验切磋琢磨、去伪存真、取其精华的反思之作。本书几乎涵盖了有关设计的所有议题：从设计哲学到设计实践，从设计过程到设计灵感，既强调了设计思想的重要性，又对沟通中的种种细节做了细致入微的描述，以及因地制宜做出妥协的具体准则等。特别深入分析了设计模型背后的工程思想，这对设计界的研究者和实践者而言无疑具有方向性的指导意义。

本书运用大量图表和案例，深入浅出地表达了复杂艰涩的设计思想，意图刺激设计者和设计项目经理，令其深入思考设计的过程，特别是设计复杂系统的过程。本书适合各类软硬件设计者、设计项目经理、设计研究人员等。

Simplified Chinese edition copyright © 2011 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *The Design of Design: Essays from a Computer Scientist* (ISBN 978-0-201-36298-5) by Frederick P. Brooks, Copyright © 2010.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2010-2468

图书在版编目（CIP）数据

设计原本：计算机科学巨匠Frederick P. Brooks的思考 /（美）布鲁克斯（Brooks, F. P.）著；InfoQ中文站，王海鹏，高博译. —北京：机械工业出版社，2011.1

（计算机科学丛书）

书名原文：The Design of Design: Essays from a Computer Scientist

ISBN 978-7-111-32557-4

I. 设… II. ①布… ②I… ③王… ④高… III. 软件设计 IV. TP311.5

中国版本图书馆CIP数据核字（2010）第228594号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：陈冀康 秦健

北京瑞德印刷有限公司印刷

2011年4月第1版第2次印刷

185mm×260mm · 18.5印张

标准书号：ISBN 978-7-111-32557-4

定价：55.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反

馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



HZ Books

华章教育

华章科技图书出版中心

Frederick P. Brooks, Jr. 论设计原本^①

很少有人像Frederick P. Brooks, Jr. 一样对软件开发的实践（而不是学术理论）产生那么大的影响。他在《人月神话》中记录了他作为IBM System/360计算机以及Operating System/360项目领导者的经验，这些经验不断地促使我们形成项目管理的观念。很难找到比他的“没有银弹：软件工程中的根本问题和次要问题”（《Information Processing 1986, Proceedings of the IFIPS Tenth World Computer Conference》，H.-J. Kugler编辑。Amsterdam: Elsevier Science, 1069-1076）一文被引用更多的文章。“银弹”的概念代表了对困难的软件和编程问题的某种近乎神奇的解决方案，这是我们的词汇表中不可缺少的一个词。

Brooks的这本新书拓展了他以前的思想，添加了对设计的本质和重要性的新领悟。毫无疑问，这本书将会成为所有专业开发者书架上必备的书籍。

这本书的副书名是“计算机科学巨匠Frederick P. Brooks的思考”。书中包含了可以作为独立的文章进行阅读的20章，每一章与其他章的耦合都比较松散。这是好事，因为大多数读者都希望阅读本书的每一章（不一定要按顺序），然后在继续下一章阅读之前思考一下所讲的内容。除了这些文章外，这本书还提供了8个案例，涉及的范围从海滩小屋的设计到IBM System/360的架构。这些案例阐释了本书中的一些重要概念。

什么是设计？Dorothy Sayers（英国作家和戏剧家，Brooks引用了他的话）说设计有3个阶段：概念构造的形成，在真实媒质上的实现，与真正用户的交互。Brooks在他的“银弹”一文中指出，第一个阶段（即概念构造）是软件工程最困难的阶段。但在1986年时，“概念构造”似乎更侧重计算机在执行指令时内部发生了什么。在这本新书中，关注的重点更多地转移到架构、外观，以及程序工件与环境的交互上。有趣的是，Brooks在第1章的开始引用了培根的话：

（新思想来自于）将一门艺术中的领悟联系并应用到另一门艺术中，历经若干次这样的经历而有所悟，脑海里自然就孕育出了“新思想”。

这清晰地表达了跨学科的灵感和锻炼。Brooks没有深入这个概念，即使在后面的内容中讨论如何“创造”了不起的设计师时，他也没这样做。

有几章讨论了设计和设计过程的模型。Brooks在这里严厉批评了流行的“理性主义者”设计模型（读者可能最熟悉的就是“瀑布方法”），并断言“理性模型过于简单”。书中指出了理性模型的诸多缺陷，包括“对理性模型的最具破坏性的批评可能是，最有经验的设计完全不是这样工作的”。（在本书中提到David Parnas多次，但没有提到他的著名文章“The Rational

^① 原文参见：<http://www.infoq.com/articles/brooks-design-book-review>。

Design Process: How and why to fake it”，这篇文章也对理性模型提出了严厉批评。）本书对其他一些设计模型也进行了讨论，包括：

- 迭代演进式开发，与此最接近的是Brooks对敏捷方法的讨论。
- Boehm的螺旋模型。
- 开源的、Raymond的“集市模型”。

这些讨论的结论是：设计需要某种过程以及该过程的模型（主要是为了沟通并支持协作），Barry Boehm的螺旋模型是Brooks认为最有希望的模型。

本书中的其他章节关注了以下几个问题：

- 协作与团队设计，包括远程协作所引发的问题。
- 关于“理性主义”与“经验主义”的讨论，Brooks自认是一个经验主义者。
- 约束条件的价值。其中印证了许多“设计”文献，这些文献来自工业、产品和图形设计师。设计草案（用于启动设计过程的文档）必须足够模糊，允许自由思考和表达，但必须清楚定义所有的约束条件。约束条件勾画出边界，创造性的、有想像力的、创新的设计将在这个边界之内发生。
- 讨论美与风格。
- 一些关于设计技术和实践的讨论，Brooks发现它们是有价值的。
- 需求、罪过与合约。
- 一个案例，讲述了为什么任务控制语言（Job Control Language, JCL，如果你没有在大主机上工作的愉快经历的话）可能是“有史以来最糟糕的编程语言”！

本书末尾用两章的篇幅讨论卓越的设计和卓越的设计师的问题。Brooks果断指出，卓越的设计来自卓越的设计师，而非卓越的设计过程。Brooks说，我们教育和培训软件专业人员的方式无助于培养出卓越的设计师。他指出，培养卓越的设计师的一个主要障碍就是缺少持续的实践和批评。

InfoQ有机会对Brooks提出了一些跟进问题，本文包括了这次访谈的内容。问题与回答如下：

开发项目有一个生命周期，要么是经典的线性瀑布式，要么是迭代增量式（敏捷）。设计是在生命周期特定阶段发生的事情，还是分布在所有阶段？

它集中发生在迭代开发的前面几次循环，但有时候发生在所有迭代中。

如果设计发生在所有的开发阶段中，是否在每个阶段中具有不同的形式、实质或要点？

当然是这样的！在第一次迭代中，总体架构是中心问题。在接下来的迭代中，设计工作集中于更精细的层面，除非是在满足最后期限之后的回溯，或者人们意识到需求的改变或新的机会。

约束条件在设计过程中扮演什么角色？（本问题的背景知识：其他领域的设计师常常依赖一份“草案”，他们预期/需要草案中有模糊之处，以便能够自由地“设计”，但他们需要清楚表达约束条件，这些约束条件定义了目标区域，最优设计只能在这个区域中产生。）

它们既促成了整体架构，又在较小程度上促成了细节设计。

是否有明显可以确定的设计“错误”，它们是否能在犯下时就可以意识到？（或者，这样

的错误是否像代码中的缺陷，通常需要在犯下之后许久才发现？)

大错是在开始时犯下的，而且很少意识到。如果最终被发现，通常是在现场实施之后。较小的错误是在编码开始时出现，或者是在第一个真正的用户测试原型时被发现。

大多数设计师认为他们的活动是高度协作的，至少是客户与设计师之间的协作，但设计更多时候涉及一个团队。您是否同意设计是一种协作？如果是这样，设计团队的地理分布或临时分布会带来什么影响？（显然，在今天的离岸外包环境中寻求并行设计，以应对软件开发的一般挑战。）

我用了两章讨论协作和远程协作。是的，今天大多数的设计都涉及团队。通用产品的设计不涉及与客户的协作，如iPad。对于为一个客户设计定制的产品，我非常喜欢对原型和代用品（如建筑的虚拟环境模型）尽早地、激烈地、频繁地、不断地进行用户测试。但是，我不认为这是与用户和客户进行协作设计。

您是否有一份清单，例如6点建议，可以总结您的书向设计者提供的最重要的经验？

- 1) 专心研究以前设计者的工作，看看他们如何解决问题。
- 2) 尝试弄明白他们为什么做出那样的设计决定，这是对您自己最有启发性的问题。
- 3) 仔细研究以前设计者的风格。最好的方式是尝试用他们的一些风格勾画设计草图。
- 4) 保存一本“草图本”，将您的想法、设计和局部设计记录下来，不论使用何种媒质。
- 5) 在开始设计时，写下您对用户和使用方式的假定。
- 6) 设计、设计、设计！

在您的“银弹”文章中，您谈到了“概念构建”和人类在头脑中完成这项工作时遇到的巨大困难。您觉得在这本书中一些思想是否关注并解决了概念构造这一基本困难？

肯定关注了，肯定没解决。

在第3章中，您漂亮地批评了理性设计过程，特别是瀑布式方法所包含的理性设计思想，并指出这种思想是有害的，必须抛弃。您对这种有害模型的持续存在有何看法？是否人们就是很倔强？或者尽管开发者更了解，但管理层会犯错？是否有一些文化上的偏见（尤其是西方文化上的偏见）阻碍人们抛弃瀑布模型？

第4章讨论了软件工程中的瀑布模型的持续存在（在其他学科中并不常见）是因为设计者过早期望得到有约束力的合同和确定的需求。

在第20章中，您批评了我们的教育系统（温和，但确是事实），并建议设计者需要参与“批评性实践”。Richard Gabriel长期以来一直主张计算机科学/软件工程大纲应该采用他在取得诗歌硕士学位时一样的大纲（他在很久之前获得了计算机科学博士学位）：大量的练习（每天至少一首诗）、大量的批评（来自同学和导师）、勤奋学习大师和大师的诗歌、不断自省、周期性的反省。这似乎与您的建议相似，那么您是否主张大学提供一个“软件好艺术硕士”学位？

不。熊恩在《the Education of the Reflective Practitioner》一书中提出了同样的建议，还有例子，更适合设计。所有的工程学大纲都应该强调这种方式。

哪些其他领域的研究将有助于毕业生变成卓越的软件设计师？

- 1) 算法和数据结构是最重要的基础课程。

- 2) 计算机硬件架构。
- 3) 应用领域，特别是商业数据处理、数据库技术和数据挖掘。
- 4) 心理学，特别是知觉心理学，因为用户是最重要的。

理性设计过程，实际上是所有计算机科学和软件工程，有意识地采用了宇宙的基本模型(20世纪的物理学)，即确定性的、机械的、理性的宇宙。如果那就是自然或现实，那么理性的、搜索树式的设计过程模型就很有意义。如果宇宙实际上是复杂的适应性系统，那么理性模型就会失效。您是否走得更远，奠定了复杂系统的设计或修改过程的基础，如文化或商务企业？

我不会自大到建议这样的东西。

IDEO是一家非常有名的设计公司，它的总裁Tom Kelly写了一些关于设计、设计过程和设计思考的书。他最好的一本书是《Ten Faces of Innovation》，其中他确定了每个设计团队需要的10个角色(人类学家、实验员、嫁接能手、跨栏运动员、合作者、指导者、用户体验设计师、布景师、专业护理人员 and 讲故事的人)。如果您知道这本书，是否类似的角色应该出现在软件设计团队中？怎样做到？

我不了解这本书。听起来有趣，而且有用。

您提到了Christopher Alexander和他的影响并在注释中提到了《The Synthesis of Form and A Pattern Language》一书。那代表了“理性的Alexander”，但“Timeless Way of Building”和他的杰作“Opus, Nature of Order”却更为“神秘化”。“神秘的Alexander”是否对您的设计和设计过程思想有所影响？

我受到了《The Timeless Way of Building》一书的影响。

软件领域的设计将大多数注意力放在人工制品上，即执行程序的计算机。越来越多的实践(但学术界还没开始)开始关注“用户体验设计”，即计算机所处的系统和生态环境。对于用户体验设计者如何从本书中获益，您是否有一些建议？

我觉得前面给出的建议同样适用于用户体验设计，但这个领域与软件工程领域相比，自由式教育更为重要。

您能列举出值得所有人学习的3名设计大师(任何领域)和3名软件设计大师，4至5个设计杰作，并简单说明为什么吗？

- 巴赫，作曲家
- 伦勃朗，画家
- Seymour Cray，超级计算机设计师
- Christopher Wren，建筑，特别是他在伦敦设计的教堂
- Nicholas Wirth，计算机语言
- Donald Knuth，算法

我不认为所有的人都应该学习同样的例子。人们需要接受范例的设计原则方面的基本教育，这样才能深入研究一个范例，欣赏这些设计问题和解决方案。但是，即使是门外汉也能欣赏一致性和设计概念的统一性。

……《诗》有之：“‘高山仰止，景行行止’。虽不能至，然心向往之。”

——司马迁（西汉），《史记·孔子世家》

大师之作不仅仅是指一部出自大师笔下的著作，更是特指大师的心血凝结之作。Frederick P. Brooks, Jr., 美国“两院”院士、A.M. 图灵奖和IEEE先驱奖获得者¹、软件工程界至今脍炙人口的奠基之作《人月神话》的作者，这位令人高山仰止的大师，在创作了《人月神话》35年之后，才在2010年初推出了本书。如果说《人月神话》是Brooks刚刚完成若干个改变了全球计算系统格局的重大项目，在人生和事业的巅峰时期的激情之作，那么本书则是Brooks功成名就之后，在研究和教学中将先前在设计领域中的探索心得和实践经验切磋琢磨、去伪存真、取其精华的反思之作。可以说，比起锐气有余的《人月神话》，本书更多了几分高屋建瓴的大局观以及数十年如一日积淀而成的丰富材料，是设计领域真正的大师之作。

本书几乎涵盖了所有有关设计的议题：从设计哲学到设计实践，从设计过程到设计灵感，既强调了设计思想的重要性（第8章），又对沟通中的种种细节做了细致入微的描述（第6、7章），并且也谈到了因地制宜做出妥协的具体准则（第9、10、11章）。其中，Brooks特别强调的是设计的概念完整性（第6章），这不仅对于设计过程中步骤流转时的信息传递十分关键，并且也是沟通中最需要重点注意的地方。使用同一个术语表达不同的概念，或使用不同的术语表达同一个概念，都会给设计带来剧增的成本，甚至灾难性的后果。这一点是贯穿始终的主线之一。另一条主线就是Brooks对于理性模型的批判（第2、3、5章）。由于在现行的软件工程著作和研究论文中，对理性模型导致的直接模型——瀑布模型的推崇可谓甚嚣尘上。Brooks在此处着了重墨，深入分析了理性模型的工程师心理学渊源，解释了它何以根深蒂固，然后剖析了它的实质——以拓展设计树的方式来暴力遍历问题的解空间，最后对它的种种不足提出了针对性的批评，并指出在哪些受限的条件下方可运用理性模型（瀑布模型），而在其他场合中有哪些更好的设计模型，尤其是Boehm提出的螺旋模型。这些对于设计模型的长篇论述，特别是对其背后的工程思想的深入分析，无疑将对设计界的研究者和实践者起到方向性的指导意义。

全书的案例研究是另一大亮点，这不仅包括专门的案例章节（第21~27章），而且在进行抽象的模型和思想论述时，Brooks也时时注意运用图表和案例说话，深入浅出地表达复杂艰涩的思想。并通过层次丰富的案例，展示了设计既能治大国，又可烹小鲜的强大力量和无穷魅力。比如，为了论述抵制需求蠕变的必要性时，他首先以和美国军方要员的一段对话切入话题（第4章），给读者以直观且深刻的印象。这不仅表明了即使在军事领域，设计的准则和影响仍然适用，也不经

意间揭示了作者和美国国防部——全世界最尖端的科研和工程的研发和实践基地——的深入合作关系。Brooks以揶揄的方式对待这些大企业的高管们的案例并非仅此一隅，在讲述僵死、拙劣的规格如何造成最终产品的可笑妥协时，他又举一例，美国联邦航空局的一个令人匪夷所思的系统规格造成了在最终产品中不得不以禁用一个完整系统的部分功能的方式“凑合”成一个虽然符合规格却造成不小浪费的系统，而这些最终都是由纳税人买单（第11章）。要知道，这些都是动辄涉及上百亿美元的大项目，Brooks在其中的谈笑风生绝对是一种举重若轻的大将风度。可是另一方面，Brooks又会在讲述设计中的约束时，在多处提到对自家房屋进行建筑设计和拟订整修方案时遇到的各种困难，并一一指出如何应对：有些约束只要改变一下思路就会消失，有些约束虽然无可避免但可以最小化，有些约束反映了原始设计方案中的方向性错误，等等（第1、3、17、18章）。这种十分贴近读者生活实际的例子不仅一下拉近了作者和读者的思想距离，同时也更说明Brooks热爱设计工作到何种程度，连一般人视作生活小节之处也不愿意放过，而是把它作为设计工程来研究一番²。顺便说一句，Brooks在建筑设计方面也决非业余，他曾经参与他工作至今的北卡罗来纳大学的西特森厅设计，是设计委员会的正式成员³，这在本书中也有提及（第4章）。

当然，Brooks毕竟是软件工程界的先驱，正如在他的《人月神话》或任何一本主要文献中一样，我们都能够在他的作品的字里行间感受到计算机体系结构刚刚诞生的黄金年代充满了怎样的设计思想和工程实践的生机和活力。而Brooks也十分擅长专业史料的记载和整理，并且以他独有的方式为读者展示出极为清晰的脉络。比如，他主持设计的System/360系统不仅是当代操作系统在实际意义上的先祖和典范，而且它本身仍然活跃在历史舞台上：Brooks在书中指出，System/360和OS/360上的应用程序至今仍然可以完全兼容地运行在当今的后续体系结构之上，包括晚至2007年发布的64位IBM Z/90机型——一种System/360体系结构的直接后裔机型（第24章）。正是通过这样上承开天辟地之洪荒巨擘，下接耳熟能详之主流系统和应用的史诗式描述，让我们在充分领略软硬件发展史的无限风光的同时，也深切地感受到用心设计会带来数十年前后一贯的、可持续发展的产品，而这些产品及其反过来促进的设计思想和方法论又怎样彻底地改变了我们每一个人生活、工作和沟通的方式。这不也正是包括我们自己在内的一代代设计师和架构师投身于此的原动力吗？

本书由我、王海鹏与InfoQ中文站的张龙、黄璜共同翻译完成。虽然与几位的合作还是第一次，但是整个过程进行得非常顺利和愉快。在全书的翻译实践中，我本人收获极大。能够逐字逐句地研读本书，已经是充分的精神享受——Brooks的文字无疑是值得一读再读的。再经过整个团队的协作和努力，把它的内容和意义带给中国的千百万读者，这对我们翻译工作者来说，已经是无上的嘉赏和成就了。当然，由于我们的水平所限，缺点和错误在所难免，希望广大读者不吝指正，以便在再版时予以修订。

在本书的翻译过程中，机械工业出版社和InfoQ中文站给予了我们很大的支持、鼓励和帮助。UMLChina于2010年6月下旬举办了一场“Brooks新作暨《人月神话》35周年讨论会”⁴，在本书出版前提供了一个与读者交流和讨论的机会。本书译稿成稿之前，多位友人阅读了翻译初稿并给予本人许多可贵的修正意见，尤其是和我一起工作的张龙、黄璜、王海鹏，盛大创新院的郭忠祥院长和刘海平工程师，SAP中国的范德成工程师，MBK Partners私募基金公司的章子琦分析师，富士康中国研发集团的Carl Giardina总监，以及微软总部的米琦工程师，在此一并致谢。当然，家父高学栋博士也通读了全稿并给予了我不少有关文法和表达的中肯意见。我也想借此机会向在工作和生活中给了我莫大支持的父母和家人表达我内心最深处的敬意和谢

意，希望本书的出版能给他们带去快乐。

高博

2010年11月

于盛大创新院上海总部

译者简介

高博 2004年毕业于上海交通大学计算机系，在微软公司和惠普公司有多年项目和管理经验。对程序设计语言、云计算、软件测试方法学、软件架构设计和软件项目管理方向有浓厚兴趣。近年来翻译出版了《C++：99个常见编程错误》、《微软的软件测试之道》等多本书籍。联系方式：feedback@gaobo.org，订阅博客：feed.gaobo.org。

王海鹏 1994年毕业于华东师范大学。拥有双学士学位。咨询顾问、培训讲师、译者和软件开发人员。已翻译20本软件开发书籍，主题涵盖敏捷方法学、需求工程、UML建模和测试。拥有15年软件开发经验，目前主要研究领域是软件架构和方法学，致力于提高软件开发的品质和效率。

张龙 同济大学软件工程硕士，InfoQ中文站Java社区编辑，满江红开放技术研究组织成员。热衷于编程，对新技术有强烈的探索欲，对Java轻量级框架有一定研究。目前对Mac、iPhone/iPad、Android开发、动态语言及算法具有浓厚兴趣。翻译出版了《Dojo构建Ajax应用程序》、《Spring高级程序设计》等书籍。拥有5年的Java EE培训讲师经验。联系方式：zhanglong217@yahoo.com.cn，博客地址：<http://blog.csdn.net/ricohzhanglong>，新浪微博：<http://t.sina.com.cn/fengzhongye>，欢迎follow。

黄璜 2007年毕业于重庆邮电大学。曾在某外包公司工作3年，目前在一家初创公司做开发工作。担任InfoQ中文站社区编辑两年，对于业界新动向和新技术有强烈的兴趣。目前专注于网站架构、分布式以及算法。联系方式：alexhuang1984@gmail.com。

注释

1. http://www.cs.unc.edu/~brooks/FPB_BIO.CV.04.2007.pdf，最近访问于2010年11月。

2. http://www.cs.unc.edu/~brooks/DesignofDesign/kitchen_design_notes.pdf，这份扫描件展示了作者把每一个建筑细节都当做设计问题来研究的严谨工作和生活态度。http://www.cs.unc.edu/~brooks/DesignofDesign/house/start_here.html，他甚至还把详细的设计树数字化并放入网站供读者了解每一个设计细节以及它们在决策中的定位，这种一丝不苟的工作习惯着实令人感动。最近访问于2010年11月。

3. <http://www.cs.unc.edu/~brooks/DesignofDesign/experiences.html>，“Buildings”部分，最近访问于2010年11月。

4. <http://www.umlchina.com/Chat/mmm100620.htm>，包括讨论会幻灯片与对话的音频与文字记录（会议由王海鹏主持），最近访问于2010年11月。

前言

我写这本书是为了刺激设计者和设计项目经理，让他们深入思考设计的过程，特别是设计复杂系统的过程。本书从工程师的视角关注实用性与有效性，同时也关注效率和优雅性¹。

谁应该阅读这本书

《人月神话》的目标读者是“职业程序员、职业经理，特别是管理程序员的职业经理”。在那本书中，我讨论了团队开发软件时，实现概念完整性的必要性、困难和方法。

本书在相当大的程度上扩大了范围，并添加了我35年来学到的经验。设计经验让我确信，各种不同设计领域的设计过程包含一些不变的因素。因此本书的目标读者是：

1) **各类设计者**。排除直觉的系统化设计将得到普普通通的跟随式产品和仿冒产品，没有系统的直觉设计将得到充满缺陷的、不切实际的产品。如何将直觉和系统化方法融合在一起？如何成长为一名设计师？如何在一个设计团队中发挥作用？

虽然我针对非常多的系统进行了论述，但我期望读者是偏重于计算机软件和硬件的设计者，面对这样的读者我可以提供具体的阐述。因此我在这些领域的某些例子中会涉及技术细节。其他读者可以跳过这些细节。

2) **设计项目经理**。要避免灾难，项目经理在设计他的设计过程时，必须融合理论与口口相传的经验，而不是仅仅照搬某种过于简化的学术模型，也不能临时设定一个过程，而不参考他人的理论或经验。

3) **设计研究人员**。对设计过程的研究已经成熟，这是好事，但并不是一切都好。已发表的研究成果越来越关注更狭窄的主题，大问题讨论得越来越少。对精确的期望和对“设计科学”的期望可能使得科学研究之外的出版物受阻。我建议设计思考者和研究者重新关注这些大问题，即便是在社会科学方法没有太大帮助的时候。我相信他们也会思考我的论述是否具有通用性，我的观点是否正确。我希望为他们的学科领域提供服务，将他们的一些研究结果带给实践者。

为什么要再写一本关于设计的书

创造东西是一种快乐，是一种极大的满足。J. R. R. Tolkien说上帝给了我们发明创造的能力，作为一件礼物，纯粹是为了让我们快乐²。毕竟，“千山上的牲畜也是我的。……我就算饿了，也不用告诉你。”³设计本身就是快乐的。

很多设计者从心理上和实践上都没有对设计过程进行很好的理解。这不是因为缺少研究。

许多设计者反思了他们自己的设计过程。研究的动机之一就是，在所有的设计领域，最佳实践和平均实践之间存在着巨大的鸿沟，平均实践和半吊子实践之间也是如此。大部分的设计成本是返工，即纠正错误，这通常达到总成本的1/3。平庸的设计肯定是浪费了世界的资源，破坏了环境，影响了国际竞争力。设计很重要，设计教育也很重要。

所以，根据推理，系统化设计过程将提升平均实践的水平，而结果也确实如此。德国的机械工程设计者们显然是首先采用了这一规划⁴。

随着计算机和之后人工智能（AI）的出现，设计过程的研究受到了极大的刺激。最初人们希望，AI技术不仅能够过去人类主宰的领域中承担许多例行设计的工作，甚至能够产生杰出的设计⁵。这种希望迟迟没有实现，而我本人觉得不可能实现。设计研究形成了一门学科，有一些专门的学术会议、期刊和许多研究项目。

既然已经有了这么多认真的研究和系统的处理，为什么还要再写一本书？

首先，设计过程自第二次世界大战以来，有了非常大的变化，而人们很少讨论这些变化。对于复杂产品的设计，团队设计越来越成为常态。团队常常在地理上是分散的。设计者越来越脱离产品的使用和实现，通常他们不再亲手打造他们设计的东西。各类设计者现在都陷在计算机模型中，而不是陷在图纸中。正式设计过程的教育越来越广泛，而且通常是雇主强制要求的。

其次，仍然存在许多误区。当我们试图教学生怎样做好设计时，我们在理解上的差异就变得很明显了。Nigel Cross是设计研究领域的一位先行者，他追踪了设计过程研究变化的4个阶段。

- 1) 规定（prescription）一个理想的设计过程
- 2) 描述（description）设计问题的内在本质
- 3) 观察（observation）设计活动的现实
- 4) 反思（reflection）设计的基本概念⁶

我在我人生的60年时间里涉及了5种设计领域：计算机架构、软件、房屋、图书和组织机构。在每个领域，我都承担过团队中主设计者和协作者的角色⁷。我对设计过程的兴趣由来已久，我在1956年的论文是“The analytic design of automatic data processing systems（自动化数据处理系统的解析设计）⁸”。也许现在是时候进行成熟的反思了。

这是一本怎样的书

令我非常吃惊的是，以下这些过程极为类似！思维的过程、人与人的交互、迭代、约束条件和劳动，都有很大的相似性。本书中反思的东西可能是隐藏在这些设计活动背后不变的设计过程。

虽然计算机架构、软件架构的历史不长，对它们的设计过程的反思也不多，但建筑设计和机械设计已经有很长的历史和荣耀的过去。在这些领域，设计理论和设计理论家都很多。

我是一名职业设计师，我所工作的领域中对设计的反思还不多，而在那些得到长期深入反思的领域，我是一名业余设计师。所以我将尝试从历史较长的设计理论中提取一些经验，应用于计算机和软件的设计。

我相信“设计科学”是一个不可能完成的目标，实际上也是一个具有误导性的目标。这种解放思想的怀疑论让我们能够从直觉和经验的角进行探讨，包括其他设计者的经验，他们很客气地和我分享了他们的领悟⁹。

所以我提供的既不是一本教科书，也不是一本包括一致论证的专著，而是一些观点文章。虽然我试图补充一些有用的参考和注解，探索一些隐秘的小路，但我仍建议读者先从头到尾阅读每篇文章，忽略这些注解和参考，然后再回过头来探索这些小路。所以我将它们藏在每章的末尾。

某些案例研究提供了一些具体的例子，文章中参考了这些例子。选择这些例子并不是因为它们很重要，而是因为它们体现出了某种经验，我基于这些经验得出了结论和观点。我特别喜欢关于房屋功能设计的那些经验，任何领域的设计者都可以参考它们。

作为主设计师，我完成了3所房屋的功能设计（详细的平面图设计、照明、电气和管道）。将房屋功能设计过程与复杂计算机硬件和软件的设计过程进行比较和对比，这帮助我提出了设计过程的“精髓”，所以我用它们作为我的案例，并且相当详细地介绍了这些过程。

通过反思发现，许多案例研究具有惊人的共同特点：**最大胆的设计决定，不论是谁做出的，都为优秀的结果作出了巨大的贡献。**这些大胆的决定有时是因为远见，有时是因为绝望。它们总是在赌博，要求额外的投入，以期得到好得多的结果。

致谢

本书的书名借鉴自40多年前Gordon Glegg的一部著作，他是一位富于创新精神的机械设计师、一个有风度的人、一位有吸引力的剑桥讲师。我很荣幸地在1975年与他共进午餐，感受到了他对设计的热情。他的书名准确地概括了我的尝试，所以我怀着感激与尊敬之情，沿用了这一书名。¹⁰

我感谢Ivan Sutherland对我的鼓励，他在1997年建议将一本讲义发展成一本书，并在10多年后对草稿提出了尖锐的批评，促进了巨大的改进。这使我后来的智力发展历程受益良多。

如果没有北卡罗来纳大学教堂山分校资助的3个研究项目以及系主任Stephen Weiss和Jan Prins的支持，这本书是不可能完成的。我在剑桥大学受到了Peter Robinson的亲切接待，在伦敦大学学院受到了Mel Slater以及他们的系主任和同事的亲切接待。

美国国家科学基金会（NSF）的计算机与信息科学工程（CISE）理事会的设计科学（Science of Design）项目由助理理事长Peter A. Freeman发起，该项目为本书的完成和相关网站的准备提供了最有用的资助。这笔资助让我能够访谈许多设计者，并能够在过去几年里将主要工作集中在这些文章上。

我非常感谢许多真正的设计师，他们与我分享了他们的领悟。我用一张致谢表列出了受访者，并在末尾列出了评阅者。有几本书提供了特别多的信息，对我产生了很大的影响，我在第28章中列出了这些书。

我的妻子Nancy是其中一些工作的共同设计者，她一直为我提供支持和鼓励，我的孩子Kenneth P. Brooks、Roger E. Brooks和Barbara B. La Dine也是一样。Roger对手稿进行了仔细的复查，并对每章内容提出了几十项建议，从概念到标点符号都有。

我要感谢在北卡罗来纳大学获得的强大后勤支持，这种支持来自Timothy Quigg、Whitney Vaughan、Darlene Freedman、Audrey Rabelais和David Lines。Peter Gordon是Addison-Wesley的出版合作者，他提供了特别的鼓励。Julie Nahil是Addison-Wesley的全职产品经理，Barbara Wood是文字编辑，他们提供了无比专业的技能，付出了特别的耐心。

John H. Van Vleck是诺贝尔物理奖获得者，当我在哈佛大学工程和应用学院Aiken的实验室读研究生时，他是那儿的院长。Van Vleck非常注重工程实践要建立在牢固的科学基础之上。他领导了美国工程教育从设计向应用科学的转变，这种转变富有朝气。就像钟摆摆到极点，反作用就会出现，教授设计从那时起一直引起争论。我非常感谢我在哈佛的3位老师，他们从未丧失对设计重要性的深刻理解并教授设计。他们是Philippe E. Le Corbeiller、Harry R. Mimno和我的导师Howard H. Aiken。

注释

1. 本书的封面图片是基于Smethurst (1967) 《The Pictorial History of Salisbury Cathedral》，他说：“……除了圣保罗大教堂之外，索尔兹伯里 (Salisbury) 大教堂是仅有的一座英国教堂，其所有内部结构都是按照一个人（或一个两人团队）的设计建造的，并且没有中断地完成。”

2. Tolkien (1964), 《Tree and Leaf》。

3. 诗篇50:10,12。强调是添加的。

4. Pahl和Beitz (1984)，在1.2.2节中追踪了这段始于1928年的历史。他们自己的书《Konstruktionslehre》历经7版，可能是最重要的系统化总结。我在所有领域中区分设计过程的研究和设计规则的研究。这些历史已经很悠久了。

5. 主要的专论是Herbert Simon的《The Sciences of the Artificial》(1969, 1981, 1996)，非常有影响力。

6. Cross (1983), 《Developments in Design Methodology》。

7. 具体的设计经历表包含在网站的附加材料中：<http://www.cs.unc.edu/~brooks/DesignofDesign>。

8. Brooks (1956), “The analytic design of automatic data processing systems,” 哈佛大学博士论文。

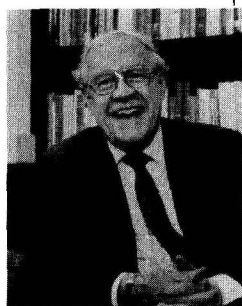
9. 我因此没有对下面网站中所说的设计方法学目标作出贡献：http://en.wikipedia.org/wiki/Design_methods (我在2010年1月5日访问了这一网址的内容)。

我们的挑战是将个人的经验、框架和观点转变成共享的、能理解的知识，最重要的是，转变成可传递的知识。Victor Margolin指出了3点困难，(其中一点是)：“……个人设计经验的著作太关注个人叙述，导致了个人的观点，而不是关键的群体共享观点。”

对此我必须承认，“你说对了”。

10. Glegg (1969), 《The Design of Design》。

作者简介



Frederick P. Brooks, Jr. 是北卡罗来纳大学计算机科学系的Kenan教授。他因担任IBM System/360系统开发的项目经理而以“IBM System/360之父”而闻名于世，后来又担任了Operating System/360软件项目设计阶段的项目经理。因为这些工作，他、Bob Evans和Erich Bloch荣获了1985年的美国国家技术奖。在此之前，他曾是IBM Stretch和Harvest计算机的架构师。

在北卡罗来纳大学教堂山分校，Brooks博士创建了计算机科学系，并从1964年~1984年担任该系的系主任。他曾在美国国家科学委员会和防御科学委员会工作。他目前的教学和研究工作集中于计算机架构、交互式计算机图形和虚拟环境。