

软件设计的艺术

Bringing Design to Software

(美) Terry Winograd 等编 韩柯 等译



机械工业出版社
China Machine Press

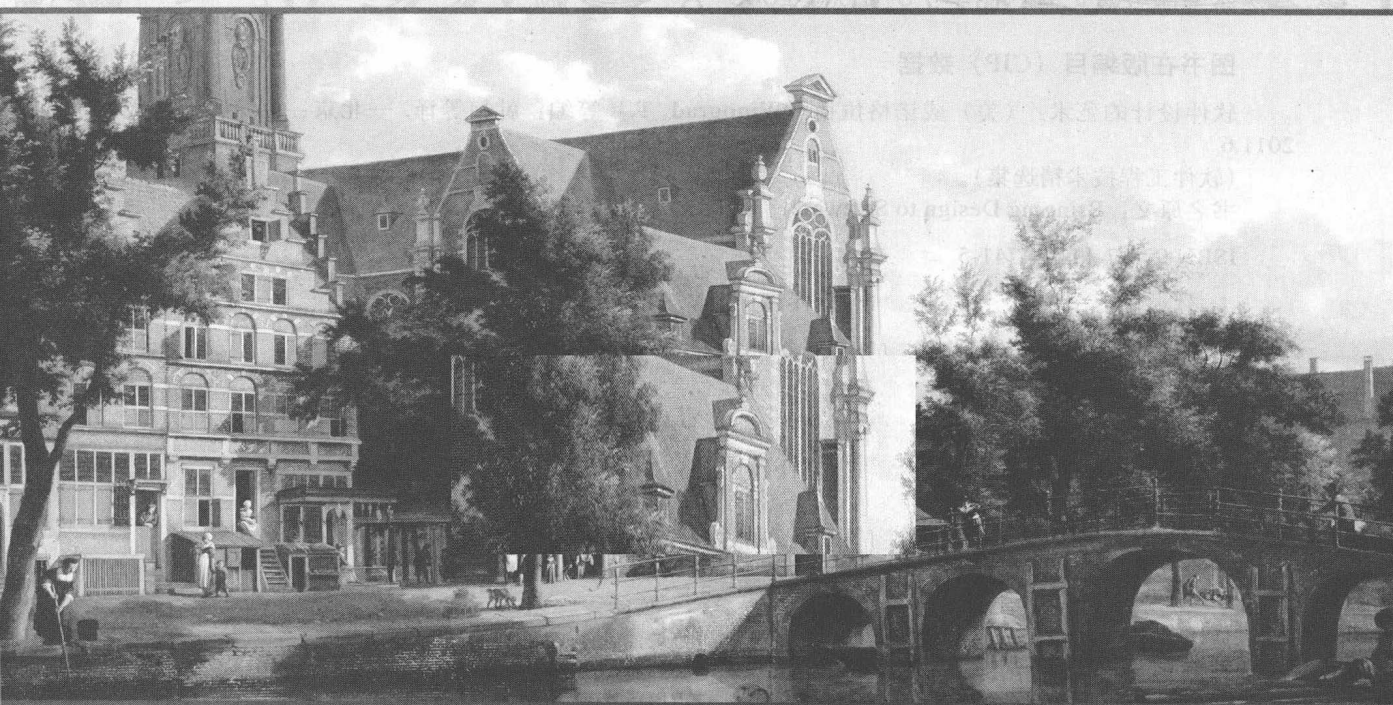


软 件 工 程 技 术 丛 书

软件设计的艺术

Bringing Design to Software

(美) Terry Winograd 等编 韩柯等译



机械工业出版社
China Machine Press

本书是一本经典著作，由众多卓越的软件和设计专业人员的论文及访谈录组成。本书的作者从不同角度，通过大量生动的实例，说明了什么是设计，给出了所有设计领域中共同的核心实践，展示了软件构建者如何运用这些实践产生使用户更满意的软件。

本书并没有很明确地提出解决软件设计问题的方法，而是引导读者站在不同的角度，在更高的层次上，对这些问题进行换位思考和更深入的分析，得出自己的结论，相信会对软件设计人员、项目经理、产品经理、系统分析员等相关人员有很大帮助。对于在校的学生来说，本书也是一本启发思路、拓展视野的好书。

Authorized translation from the English language edition, entitled BRINGING DESIGN TO SOFTWARE, 1E, 9780201854916 by WINOGRAD, published by Pearson Education, Inc., publishing as Addison-Wesley Professional, Copyright © 1996 by ACM Press.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and CHINA MACHINE PRESS Copyright © 2011.

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：01-2011-2885

图书在版编目（CIP）数据

软件设计的艺术/（美）威诺格拉德（Winograd, T.）等编；韩柯等译. —北京：机械工业出版社，2011.6

（软件工程技术精选集）

书名原文：Bringing Design to Software

ISBN 978-7-111-34741-5

I. 软… II. ①威… ②韩… III. 软件设计 IV. TP311.5

中国版本图书馆CIP数据核字（2011）第088012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：秦 健

北京市荣盛彩色印刷有限公司印刷

2011年6月第1版第1次印刷

186mm×240mm · 14.25印张

标准书号：ISBN 978-7-111-34741-5

定价：45.00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

译 者 序

自本书1996年第一次出版，到2002年10月已经重印了8次，由此可见它对软件设计界的影响程度。书中的作者从不同角度，通过大量成功和失败的生动例子，比较全面、客观地审视、分析了软件设计各个方面的问题。我们感到这些问题对于中国的大部分软件组织和开发人员来说也并不陌生，书中所描述的一些现象对于我们已司空见惯，所引用的话也常常挂在我们嘴边，当然所指出的苦头我们也尝过。

本书并没有很明确地提出解决设计问题的方法，而是引导读者站在不同的角度，在更高的层次上，通过似曾经历过的例子对这些问题进行更深入的分析，并自己得出结论，相信这会对软件设计人员、项目经理、系统分析员等有很大帮助。对于在校高年级学生来说，本书也是拓展视野的相当不错的参考书。

在翻译过程中，我们力求忠实原文。但由于译者的知识水平和实际工作经验有限，不当之处在所难免，恳请读者批评指正。参加本书翻译、审校和其他辅助工作的人员还有黄慧菊、屈健、刘芙蓉、王威、李津津、原小玲、韩文臣等。

韩柯

2004年9月

前 言

“软件”是一个有歧义的词。谈论软件的人心里想的也许是程序组件的结构，也许是应用程序的功能，也许是界面的外观，也许是硬件-软件环境的总体用户体验。所有这些视点中的每一个都会对所讨论的问题、对能够设计的内容、对恰当的工具和方法有自己的理解语境。

“设计”也是一个有歧义的词。在其很多种含义中，有一点是共同的，即当被设计的对象在实践中被体验时，把设计人员的意图和活动与所产生的结果联系起来。虽然设计原则千差万别，但是我们仍然可以发现能够适用于任何对象（不管这种对象是一张海报，一种家用电器，还是一座住宅）设计的共同考虑因素和原则。

本书的目标是通过从更宽的视点考虑设计，通过探索如何将所有设计领域内的经验教训运用到软件中，来改进软件设计的实践。这里，我们有意识地使用了“探索”这个词，其含义是“本书旨在启发开放式思维，而不是成为一种关于规则和方法的详细手册”。软件设计仍然是一个很新的领域，我们还远远没有得到对相关原则的清晰描述。软件设计是一个面向用户的领域，因此具有各学科（例如建筑和图形设计）的人员开放性，并不是一刀切式的刻板确定的工程设计。

本书收集了关于软件设计共同问题的不同观点。本书作者包括软件设计人员、其他领域的设计人员、研究人员、教师、软件行业的管理者和行业分析人员。本书每一章都将从不同角度讨论两组关键问题。

1) 设计是什么？在设计中会发生什么？

2) 我们怎样通过把对设计的广义理解运用到软件设计实践来改进软件？

在交流不同的观点时，本书的作者为具有不同职业背景并由于不同原因对软件发生兴趣的读者设计了自己的具体内容。本书的主要读者是把自己看做“软件设计人员”的越来越多的人，即日复一日地生产新的软件、界面和用户体验的人。使用一句现在很流行的话，他们的工作最终要“脚踏实地”。有效的软件设计需要有效的软件设计人员，本书将通过各章正文和背景简介中给出的软件设计例子来说明读者所关心的问题。

但是，个体设计人员并不能代表软件设计的整体情况。软件设计人员在组织中工作（和与组织一起工作），组织中包括各类管理软件开发、销售软件、开发硬件、编写文档和执行所有把软件带给用户所需的其他活动的人。对设计过程和对软件设计目标的更深理解，对于在软件企业工作的每一个人都会是很有价值的。本书的很多章节和例子对于管理设计组织的人和对于在这些组织中工作的人同样重要。

跳出具体的设计环境，本书还介绍了人机交互专业的学生与研究人员的观点。虽然本

书主要不是一本学术分析著作，但是它也给出了很多参考文献，而且本书每位作者都给出了专业理论领域方面的推荐读物。本书提出的问题要比所解答的问题多得多，列出参考文献有助于读者寻找问题并探索自己的答案。

出于进一步扩展读者范围的目的，我们希望本书能够使其他设计学科专业（例如建筑设计、图形设计、产品设计和城市设计）的人士感兴趣。本书包含了来自所有这些领域设计人员撰写的内容，以探索他们如何从自己学科的观点出发来进行软件设计。这些学科的同仁会发现这对于将其技能用于软件设计并反映其自己工作的性质很有价值。

最后，本书的一个总的主题是：有一类人员在软件设计中是最重要的，这类人员就是用户。虽然我们主要针对的是怎样设计好软件，但是本书还讨论了软件的选择、引入和使用。如果软件应用程序的每个采购者和使用者都对软件设计有很好的理解，那么这就会迫使软件界还以设计得更好的软件。我们希望那些关心软件能够为自己做什么的大量软件用户也是本书的读者，这样他们所使用的软件会变得更适合、更可用，使他们得到更大享受。

我们的目标是揭示设计的共同和永恒的内容，同时也研究软件设计的当前状况，即在软件设计短短的历史中形成的传统案例和实例。本书是写给善于思考的读者，即受实际考虑的驱动，并且又有能力从高层审视哪些做法有效、哪些做法无效以及为什么会出现这种情况的设计人员。

本书的起源

本书的构思是从1992年夏天召开的一次软件设计研讨会开始的，这次研讨会由斯坦福大学组织，得到美国国家科学基金会、Interval Research公司和软件设计协会的支持，会议的主题是“人、计算机与设计”。30位软件设计人员、图形设计人员、工业设计人员、研究人员和教师汇聚到一起，就什么是软件设计、软件设计应该是什么、自己各自学科中的哪些设计原则有助于定义和提升软件设计的新理念等问题，进行了3天热烈的讨论。会议结束时，几位与会者同意合作出一本书，把各自的观点汇聚为一个完整的软件设计描述。从那以后，我们对本书的很多版本进行了大量的编写、修订和意见征询工作。最终形成了一个完整的论文和访谈集，所涉及的问题都是有关软件功能、易用性和价值等核心的内容。

致谢

像编写本书这样时间很长、范围很广的任何项目，都需要很多人的智慧和技能。我们的工作得到了同事们的帮助，衷心感谢他们的理解和所做的工作。首先要感谢每章的作者，感谢他们仔细地修改所撰写的内容，感谢他们贡献出很有见地的思想。编写本书的过程就是一种具有挑战性的对话过程，所有作者都非常投入和积极。

很多人在创办最初的研讨会时都作出了贡献，包括会议的组织委员会——Bradley

Hartfield、Mitchell Kapor、David Kelly、Donald Norman、Donald Schön、Andrew Singer 和 Terry Winograd，以及会议主席 Cynthia Lewis、副主席 Barry Polley。除了本书各章的作者，其他与会者还有：Jeanne Bamberger（麻省理工学院）、Nathaniel Borenstein（Bellcore 公司，即现在的 First Virtual 公司）、Larry Bucciarelli（麻省理工学院）、Robert Carr（Autodesk 公司）、Parvati Dev（斯坦福大学医学院）、Laird Foshay（Tabula Interactive 公司）、Nick Flor（加州大学圣地亚哥分校）、Chris Graham（微软公司）、John Hestenes（国家科学基金会）、Susan Kare（设计顾问）、Jason Lewis（Vivid 公司）、Kyle Mashima（Objective Software 公司）、Raúl Medina-Mora（Action Technologies 公司）、Gary Perlman（俄亥俄州立大学）、David Rine（George Mason 大学）、Kurt Schmucker（苹果计算机公司）、Lee Sproull（波士顿大学）、Suzanne Stefanac（*Macworld*）、Susan Stucky（学习研究所）、Bill Verplank（Interval Research 公司）、Patrick Whitney（伊利诺斯理工学院，设计研究所）、Sean White（Interval Research 公司）和 Kären Wieckert（斯坦福大学）。

研讨会和本书后续工作得到了 Interval Research 公司的慷慨支持。我们要特别感谢 Interval Research 公司的总裁 David Liddle，感谢他对本书工作的一贯支持，感谢他通过很多方式对软件设计开发的支持。本书得到了国家科学基金、计算机和信息科学与工程理事会的资助，资助编号 #CDA-9018898。

Robert Brunner、Allan Bush、Laird Foshay、Jonathan Grudin、Peter Hildebrand、Barbara Knapp、Donald Lindsay、Mary Miller、Barry Polley、Richard Rubinstein、Suzanne Stefanac、Howard Tamler、Bill Verplank 和 Sean White 为本书提供了宝贵的帮助并进行了审阅。Addison-Wesley 出版公司的 Peter Gordon 和 Lyn Dupré 积极参与了本书的出版工作，热情地提供了作者所需的编辑方面的帮助。

最后，把前面的“我们”换为“我”，表示我个人对几位合作者——John Bennett、Laura De Young 和 Bradley Hartfield 的衷心感谢和敬佩。他们对本书工作的投入程度不亚于一本书的单独作者。他们为本书各章付出了大量心血，阅读、评阅了初稿，参加了几十次的会议和讨论，以各种可能的方式对本书的出版做出了贡献。“没有他们，本书是不可能出版的”是句套话，但是对于本书来说，这句话绝对是真的。本书从头到尾都是集体努力的产物。

现在，要感谢的人扩展到了您，本书的读者，我们期待着您的积极参与。我们希望本书能够启发您拓展视野，在未来的对话中讲出软件是什么、软件能够做什么以及怎样才能更有效地设计软件，以满足人们的需要和期望。

编者：Terry Winograd <Winograd@CS.Stanford.edu>

参编者：John Bennett <Bennett@PCD.Stanford.edu>

Laura De Young <Laura@Windrose.com>

BradleyHartfield <BradleyH@HartfieldDG.com>

于美国加利福尼亚州的 Palo Alto

引 言

在当今这个充满计算机的世界中，人们很难记住微处理器革命使计算机走出象牙塔，成为亿万人日常工作和娱乐的一部分仅仅才几年的时间。从那时起，软件业从计算机（硬件）业微不足道的附属行业，发展成为重要的经济产业，并造就了世界上一些大型的公司。

现在我们正在见证计算机及其软件在人们生活中扮演角色的第二次转变。硬件技术和软件设计的进步正在改变着软件产品的定义。我们正在从以操作系统和一体化字处理软件为代表的软件产品时代，走向一个以交互式多玩家游戏、信息网络和个人通信设备为代表的新阶段。

在加速超越20世纪80年代传统应用程序的过程中，一种新的软件问题在20世纪90年代开始变得突出，它就是我们可以与“设计”这个词联系起来的软件问题。

在20世纪90年代的一次大型软件制造商会议上，Lotus 1-2-3的设计者Mitchell Kapor号召其软件公司的经理们重视设计。他用“软件设计宣言”的形式提出挑战（参见本书第1章）：

尽管个人计算机取得了巨大成功，但是使用计算机的日常经历对于大多数人来说，往往还是充满困难、痛苦和障碍……软件没有易用性和糟糕的程序设计，是软件界的隐痛。

Kapor的号召不是对牛弹琴。他提出的问题正在引起人们深思，他的观点引起了软件开发界的共鸣。他的宣言发表没有几年，软件设计就得到了很大发展：

- 1992年，一个新的专业组织“软件设计协会”（ASD）成立了，协会的使命是“转换和提升软件设计活动的状态和质量”。ASD在整个美国的多个城市成立了分会，并启动了教育活动正式计划。
- 1994年，计算机协会（ACM）出版了一本新杂志《Human-Computer Interaction》，ASD和ACM还联合成立了人机交互特别兴趣组（Special Interest Group on Computer-Human Interaction, SIGCHI）。杂志编辑John Rheinfrank和Bill Hefley在创刊号（1994年1月，第88页）上指出：

看起来我们的思想已经超越了使计算机“有用”只不过就是设计“人和机器”之间好的界面的观念。我们的思想已发展为人员的丰富经验应该放在前台，计算环境应该成为服务于这种经验的背景。

- 《Human-Computer Interaction》创刊号上通篇是就John Seely Brown和Paul Duguid

一篇论文的对话，这篇论文是关于语境（context）在设计中的角色的。

- 在1994年的SIGCHI大会（CHI'94）上Kapor发表了主题演讲。在这次演讲中，他提出设计对于人机交互是第一位的。会上有相当多的论文和研讨会是专门针对设计问题的。CHI'95大会创立了一份《设计简报》，专门刊登著名的设计，并讨论这些设计是怎样达到这种程度的。
- 软件界已经开始重视软件的易用性。1995年，微软公司试图斥资15亿美元购买Intuit的财务管理程序Quicken，主要是因为客户发现与微软公司自己的软件相比，Quicken的可使用性和可理解性更强。这笔交易被政府阻止了，因为这样会使微软公司太强大，从而垄断财务软件市场。
- 经过25年的发展，因特网全球计算机网络突然呈现在人们的眼前，使人们激动万分。因特网的普及在很大程度上是由于像Mosaic及其后续产品Netscape界面的影响，这种界面使因特网能够被最大多数的人所使用。

这些都说明软件设计正在引起人们的重视。不过我们需要冷静地思考一下。软件设计到底是什么？软件设计与程序设计、软件工程、软件体系结构、人员因素和界面设计怎样区分？软件设计与自称为设计的其他领域（例如工业设计、图形设计、信息设计、城市设计、甚至时装设计）有怎样的联系？创造新标签是很容易的，但是真正的工作还是要带来能够产生新方向和新思想的视角的变化。

什么是软件设计

软件设计协会（ASD）很高兴地发现有这么多一直从事软件设计的会员，尽管他们公司的人事档案还把他们叫做软件工程师、程序员、程序经理、人员因素顾问或很多各种各样的其他头衔。ASD的会员手册给出了以下定义：

软件设计位于所有计算机学科（硬件与软件工程、程序设计、人员因素研究、人类环境工程学）的交叉路口。软件设计是关于人、机器以及将人和机器连接起来的各种接口（包括物理接口、感知接口和心理接口）的交叉学科。

每当创建供人使用的对象时，都要进行设计。设计可以是系统的，也可以是很随意的；可以是自觉的，也可以是自发的。但是，当人们创建软件或任何其他产品时，都要做出决策，都要构建对象，通过对象体现人们要做什么，以及如何感知和使用这些对象的意图。

计算机职业教育常常关注计算机制的理解，关注寻求保证这类机制按照程序员的意图运转的工程方法。这种教育的焦点是被设计的对象：硬件和软件，其主要考虑是高效地实现所描述的功能。当软件工程师或程序员说一段程序“成功”时，一般是说这段程序是健壮的、可靠的，是可以满足其功能规格说明的。这些考虑确实重要。任何设计者忽略这些问题都会面临灾难性的风险。

但是这种关注功能和结构的向内看的视点只是一个方面。为了设计确实“成功”的软

件，需要从构建者的视点转移到设计者的视点，把系统、用户和语境合起来作为起点。当设计者说什么东西（例如，一本书的封面设计或一个住宅设计）是“成功的”时，实际上有更宽的含义。好的设计产生一种就价值和需要而言对人们是成功的对象，产生高质量的结果和令人满意的使用经历。

什么是软件

对于一本关于软件设计的书来说，很明显我们考虑的是设计软件。但是仅仅从字面上理解还是远远不够的。可以从很多不同视点研究软件，每种视点都有自己的设计含义。在本书中，我们强调软件是一种创造虚拟环境（virtuality）的媒介，是一种软件使用者感知、操作和响应经历的世界。

虚拟世界的创建可以通过计算机游戏马上看出来。这种虚拟世界吸引玩家探索无尽的空间，在峡谷中战斗，寻找宝藏，积极主动地生活在游戏设计者能够想象到并且描绘出来的任何世界上。在桌面界面、电子表格和因特网中也有虚拟世界。界面和世界之间的差别已经被Xerox Star（图形用户界面的鼻祖）的设计者们发现，他们关注的是被David Liddle（请参阅本书第2章）称作“用户观念模型”的东西。后续项目使用了其他术语，例如“概念模型”、“认知模型”、“用户模型”、“接口隐喻”、“用户幻觉”、“虚拟环境”和“存在论”，所有这些术语都包含这样的意思：要设计的是一种存在空间，而不是一组设备或影像。“虚拟环境”这个词说明了这个世界是虚拟的这个特点，在这个空间中既没有用户的精神结构，也没有设计者的精神结构。

今天我们都熟悉标准图形用户界面虚拟环境，熟悉这种虚拟环境的窗口、图标、文件夹等。虽然这些虚拟对象大致可以与现实世界相比拟，但是它们还是存在于自己的独特世界中，具有自己的特殊逻辑和潜能供使用者采取行动。内部程序操纵磁盘磁道、网络地址、数据缓存和程序段。这些基础内容并不出现在用户所使用的桌面虚拟环境中，至少在一切正常运转时不会出现。

软件并不只是用户与之交互的设备，软件还是用户在其中生活的空间生成器。软件设计与建筑一样——当建筑师设计一栋住宅或办公大楼时，要描述的是一种结构。不过更重要的是，要形成大楼住户的生活模式。人被看做大楼的住户，而不是大楼的用户。本书把软件用户看做住户，关注用户如何在设计师所创建的空间中生活。我们的目标是把设计人员的成果置于用户世界中。

本书采用的这种软件视点并不是惟一合适或有价值的视点。设计人员只有通过对比各种视点理解软件才可以掌握软件，就像掌握任何技术一样。

软件工程

“软件设计”这个词常常被用来刻画又叫做“软件工程”的原则，即关注构建高效、可靠、健壮和易于维护的软件。有关把软件设计作为一种工程活动的参考文献（例如，

Pfleeger, 1991; Rumbaugh, 1991; Blum, 1992; Brooks, 1975, 1995) 与本书考虑的问题互为补充。Blum (1966) 从设计角度明确地叙述了软件工程问题。

接口设计

本书书名可能会产生一些误解。《Bringing Design to Software》的含义是设计的对象是软件，没有提不可避免地要嵌入到用户软件中的接口设备。我们不可能将设计很好地划分为软件设计部分和设备设计部分。软件实现的可能性既是由物理接口创造的，也是由物理接口约束的。在今天这个计算机应用世界中，绝大多数应用程序都是以某种标准方式把自己表示给用户的，即可视显示加上键盘和点击设备。但是未来的计算环境将会把更丰富的资源带给物理人机交互。有些最先进的设备已经在一定范围内使用，例如基于笔的个人数字助理 (PDA)、虚拟现实目镜和手套、嵌入到各种机电设备中的计算机。研究人员正在探索进一步的可能性，包括触摸输入和输出设备、融入式环境、音响空间、可穿着计算机以及与现在的个人计算机和 workstation 没有多少共同之处的大量设备。现在有关人机交互的很多教科书和文献 (例如 Shneiderman, 1992; Dix 等, 1993, Preece 等, 1994; Baecker 等, 1995) 都涉及了接口设备。随着各种各样设备的不断增多，以设备和软件新组合为基础的交互设计会成为一本重要的新书，我们暂时叫它《Software Design》吧。

人机交互

只要设计与人交互的软件，设计工作都会超出软件本身，都会涉及人们在接触和使用该软件中的经历。接触任何人工产品的人都要运用知识和理解，这些都要以涉及人们感知、记忆和行动能力的各种认知机理为基础。人机交互方面的研究人员研究过计算机用户的心理世界，正在探索预测交互性质和支持接口设计的方法和途径。虽然还不能说人机交互认知分析是已经被普遍接受、并得到广泛使用的方法，但是对于设计交互式软件的任何人来说，有很多文献 (例如 Card 等, 1983, Norman 和 Draper, 1986, Helander, 1988, Carroll, 1991) 还是很有价值的。一些机构，如 ACM SIGCHI、人员因素与人类环境工程学学会 (请参阅 Perlman 等, 1995)、易用性专业协会等，也促进了人机交互的实际使用。

艺术

人参与和计算机系统的交互，并不限于人机交互主流文献所探索的认知领域。作为人，我们还要有美学和情感层面的参与。由于计算机最初是在实验室和办公室中使用的，因此非认知方面的问题在很大程度上一直被忽略，但计算机游戏的设计者除外。只要人们使用一段程序，不管是电子表格软件还是物理仿真软件，都会有自然人的反应。他们可能会体验到美感、满足感和乐趣，也可能会体验到相反的感觉。

随着计算环境逐渐与娱乐技术结合，随着典型用户从有严格管理的办公室转向家庭休闲房间，软件设计者需要更多地关注人员反应的情感问题 (请参阅 Laurel, 1993)。我们可以通过其他人员沟通媒介的历史进行学习，可以借用小说家、电影制作人、作曲家、视觉

艺术家和很多其他被称作“艺术”的设计人员的原则和手段。全面考虑人员体验的设计可能是下一代软件设计要讨论的主题。

什么是设计

甚至比定义什么是“软件”还要困难的是定义什么是“设计”。字典提供了几个有些重叠的定义，浏览一下图书馆或书店的设计专题划分更会使人感到困惑。虽然我们把设计用作一个名词，但是设计并不是一个“东西”。我们会经常问的问题是“设计活动”的。本书各章的作者并没有给出一个简单定义，而是通过提供人们在设计过程中所作所为的一种视点回答这个问题。

设计是有意识的

只要对象能够很好地适应所处的环境，人们就会认为这个对象是经过良好设计的，即使这种适应性源自无意识的进化过程。本书关注的是当设计人员反映并把注意力放在了设计活动上会发生什么。复杂系统没有一致的精心设计也可以发展，例如城市（请参见 Alexander, 1964）和因特网（请参见 Krol, 1994），但即使是这样，有意识的设计也会在创建构成整个整体的单个部件和关系中发挥作用。

关于设计的有意识性并不意味着要运用某种形式化的、一致的或全面的设计理论或某种统一的方法论。系统化的原则和方法有时可以用于设计过程，但是现在还没有出现与那些广泛适用于数学和传统工程学科的合理化理论同样有效的适用于设计过程的理论。设计的有意识性仍然弥散着直觉、默认知识和大胆反应。

设计使人关注核心问题

所有工程和设计活动都要求进行折衷管理。现实世界的问题很少有像适合数学问题或教科书上的练习那样的“正确解决方案”。设计人员在由竞争价值和所需的资源构成的候选方案中寻求创造性的解决方案。在经典工程学科中，这种折衷常常是可以量化的——材料强度，构建成本，磨损率，等等。在设计学科中，折衷更难识别和度量。设计人员的一只脚站在技术上，另一只脚站在主观考虑领域中，而这两个世界的比较是不容易的。

举一个例子，软件设计人员很容易陷入对某种需求的全身心追求，在这种情况下，易于使用（特别是对新手用户）成了神圣的目标。但是什么是“易于使用”？易于使用对谁重要？重要程度如何？小提琴对于新手来说是很难演奏的，但是不能愚蠢地因此认为就应该用自鸣琴替代小提琴。一件手工产品的价值可能要通过艺术大师高度熟练的使用来体现，也可能通过某类特殊用户（例如儿童和残疾人）的易于使用来体现。就像马力强劲的赛车和自动传导的小面包车可以共存一样，软件也是不同的。

在本书第5章中，Paul Saffo 研究了高技术消费者关注的一些因素，并引入了一种他叫做“愤怒阈值”的概念，即在折衷曲线的某个点上，对于不同用户组来说，感知到的价值

被感知到的争议压倒。用户关注问题的多方面考虑对于理解新软件和电子技术会在实践中带来什么至关重要。

设计是与材料的会话

现行设计过程在两个层次上是迭代式的。设计人员对当前的一件开发任务进行的迭代，以及整个团队不断进行的发现新的中间机会的迭代。单个设计人员的行动反映循环的详细信息可参见对Donald Schön的采访（参见本书第9章），Shahaf Gal的一个学生开展的机械设计项目对此也进行了解释（参见本书第11章）。正如Schön和Gal所指出的那样，设计在本质上是复杂的，设计人员作出的每个选择都既是有意识的结果，也是无意识的结果。设计并不完全是一种精心策划和执行的过程，因为设计是一种会话，其中会话伙伴就是被设计的对象本身，它能产生未预料到的中断和贡献。设计人员不仅要形成设计，而且还要倾听正在完成的设计的声音。

设计总意味着某种构建方法，而新技术又带来新的设计领域。就我们所知，建筑学的出现是由于出现了采用石料进行建筑的技术，图形设计的出现是由于印刷技术的出现，现代产品设计则由于塑料技术的发展扩展了能够设计的日用产品的各种外形而欣欣向荣。计算机产生了“交互设计”领域，在这个领域中可以以前所未有的速度创建和修改新型的虚拟环境（virtuality）。新的原型化媒介对整个设计界所产生的影响，可参见Michael Schrage在本书第10章“原型文化”中的讨论。

设计是创新

制定一个良好设计的标准是一回事，而做好设计是另外一回事。人们写了很多书讨论设计的系统化方法问题，尤其是界面和交互系统的设计（例如：Hix和Hartson，1993，Newman和Lamming，1995）。这些文献提供了有用的指南，但是设计是一种不能完全归结为标准步骤的创新性活动，这种活动甚至都不能被理解为“问题解决方法”。正如David Kelley在第8章所论述的那样，设计人员没有组织完备的工程准则来进行恰当的约束，因为设计本质上是混乱的。设计要求解决问题中的创新能力，但不仅限于此。设计从发现问题这种创新活动开始，预想人们有但是还没有意识到的需求。

正如Gillian Crampton Smith和Philip Tabor在第3章所描述的那样，对于“艺术家—设计者”，交互设计更像是艺术，而不是科学，因为这种设计是自发的、不可预测的和难以定义的。艺术家—设计者的技能不能归结为一套方法，不能像科学和工程学科那样，通过结构化的课程学习获得。另一方面，这种设计也不是神秘的天赋。在设计领域已经有了很长时间的的教育传统，这使交互设计介于学习者和教育者、设计者和评论者之间。

设计是沟通

前几节讨论了用户与所在世界之间的交互，以及设计人员与自己材料之间的交互。这两种交互哪种更重要呢？虚拟环境并不只是设计人员所想象的环境，也不是任何特定用户

所想象的环境。虚拟环境存在于设计人员和用户的不断沟通之中。为了获得设计的成功，我们需要理解设计人员怎样传递自己的思想。

从外层看，所设计的产品要与内容融会贯通。正如Crampton Smith和Tabor所指出的那样，即使是像列车时刻表那样简单的对象也会传递多个层次的含义。各个领域的熟练设计人员都懂得如何管理这些层次的含义。此外，一种主动的人工产品，不管是计算机程序还是咖啡机，都是与用户就这种产品的使用进行沟通。设备就像是门，通过约定与用户进行沟通。门上大约与肩同高的位置上挂上一个写有“推！”的牌子，带有把手的门上写上“请拧这里”。虽然这些消息都受物理机制的约束，但是也需要约定和学习，例如每个旅游者都会发现需要学会对付自己不熟悉的文化背景下的日用设施。John Rheinfrank和Shelley Evenson（请参见本书第4章）解释了这些物理设施是更宽的设计语言所要说明的内容之一，需要沟通多种含义，包括功能、感知、内涵和美学上的含义。只要设计人员构建的是供人使用的对象，就需要利用团队和文化共有的设计语言背景。

与所有人类语言一样，通过文字能够看到的只是所有内容的一小部分。听到“我要杀了你！”这句话，如果不知道说话的上下文环境，就不能解释其真正含义——是在走向网球场的路上对朋友说的，是写给总统的匿名信上的一句话，还是父母对又把脏衣服丢在卫生间地板上的孩子说的？字面含义只是上下文环境中含义的影子。

对于我们所构建的产品（包括软件也是这样）：人们不能孤立地理解它。每个对象都出现在一种预期的语境中，这种语境是由以前的对象和体验的历史，以及由外围（即该对象要面对的物理、社会和历史语境）的环境来产生的。John Seely Brown和Paul Duguid在本书第7章描述了每个设计人员在创建某种新东西时都会利用的“边界资源”，而这些资源不可避免地要继承以前的含义。他们认为，在软件设计中对外部环境的漠视会导致一些设计，这些设计可以扩展传统形式的字面功能，但是却不适合更大的人员环境。

设计具有社会影响

有关软件设计的大量讨论都使用通用的有大量用户的软件例子，例如字处理程序、操作系统、电子表格程序、图形程序和游戏程序。虽然软件设计的很多关键考虑因素都可以通过这些应用程序说明，但是也还有些考虑因素不能通过这些应用程序说明。这些高度可视应用程序是各种软件的一部分，还有很多专用应用程序（例如医院计费程序）以及专门针对某些工作场所环境设计的系统。在这些更有针对性的应用程序中，设计人员可以看到并考虑设计对将使用软件的人员的具体影响。

Sarah Kuhn将在本书第14章介绍软件设计的组织问题，讨论在为具体组织和工作场所构建集成化的计算机系统时应该考虑的核心问题。这些工作场所中的人员需求和考虑因素会导致复杂有时甚至是矛盾的设计考虑因素，这些问题只有通过把社会和政治因素明确地纳入分析和设计对话中才能解决。设计人员不仅要进行技术设计，而且要进行组织设计，并通过诸如“共享设计”这样的手段使员工直接参与到这个过程中来。

Peter Denning和Pamela Dargan将在本书第6章介绍一种他们叫做“以行动为中心的设

计”的技术，这种技术将工作内容作为设计的中心焦点来陈述。他们使用传统软件工程技术的经验说明这些技术有失败的地方，设计人员可以通过关注工作结构，而不是支持工作的信息系统的结构来解决这些问题。

设计是一种社会活动

如果只关注单个设计人员的活动，可能会得出“设计的整体质量主要决定于有创造性的个人的素质和活动”这个错误结论。**Kelley**指出，设计人员是在更大的背景环境下工作，要受到与他人互动的促进和制约。

Donald Norman在本书第12章生动地叙述了大型组织中的设计是如何受到个体设计人员考虑之外的各种因素和力量的影响。他指出，在两个层面上组织限制了设计的可能空间一个明显的层面是大型组织中的很多部门具有不同的目标和要求，另一个默认的层面是组织所特有的文化。

Norman关于组织结构如何会使设计活动变得复杂的论述得到了**Laura De Young**有关软件开发组织能够采取什么措施促进面向客户的设计分析的补充（请参见本书第13章）。**De Young**从她当顾问的工作以及在家庭财务应用软件获得巨大成功的**Intuit**产品**Quicken**的使用体验中提取例子，找出任何组织内部的设计人员都可以遵循的、关注以用户使用质量为基础的设计过程的原则。

软件和设计怎样才能配合到一起

正如前面各节已经说过的，你在本书中将不会找到软件设计的简单定义。虽然本书最初确实想收集整理出这样的定义，但是交流的结果是给出一些不同的观点，这些观点都以独特的方式扩展了这个问题的共同核心。在把这些观点整理成书的过程中，我们一直与书中各章的作者进行深入交流。本书每一章都以各自作者的口吻探索自己所关心的问题。在各章之间穿插了简短的框架，每个框架都描述了一个使本书观点在实践中得以印证的成功的项目或程序。

本书有许多重要内容需要读者仔细领会。本书不是简单地剪剪贴贴就整理为软件设计人员能够很方便地总结和使用的一体化专著。读者应该思考本书提出的问题该怎样用于解决你自己的问题，怎样用于指导你自己的活动。本书所有作者的共同之处是关注设计所具有的性质，即对人的极为丰富和多样的背景环境的敏感性。

目 录

译者序

前言

引言

第1章 软件设计宣言1

背景简介：软件设计与建筑学7

第2章 概念模型的设计12

背景简介：Alto与Star22

第3章 艺术家—设计者的角色25

背景简介：Kid Pix39

第4章 设计语言42

背景简介：Macintosh人员界面指南54

第5章 不同消费者类型58

背景简介：Mosaic与万维网67

第6章 以行动为中心的设计71

背景简介：业务过程映射82

第7章 保持简洁86

背景简介：Microsoft® Bob™97

第8章 设计者的立场101

背景简介：IDEO公司110

第9章 与材料的反应式对话114

背景简介：苹果公司的计算机界面设计

计划124

第10章 原型文化127

背景简介：HyperCard、Director和Visual

Basic137

第11章 设计的立足点142

背景简介：电子表格151

第12章 成熟的设计154

背景简介：日用物品的设计165

第13章 软件设计的组织支持168

背景简介：Quicken178

第14章 为一线人员设计181

背景简介：参与式设计192

反思195

参考文献199

： 第 1 章 ：

软件设计宣言

Mitchell Kapor

计算领域内最重要的进化应该是把软件设计人员角色定义为用户经验的斗士……设计是什么？……设计就是一只脚踩在两个世界，即技术的世界和人及其意愿的世界，设计人员努力要做的是把两者结合到一起。

Mitchell Kapor是微型计算机界最早把自己的工作定义为“设计软件”的人中的一个。当他在20世纪80年代初开发Lotus 1-2-3时，就承担了设计交互界面的任务，但是他并不是实际编程的人。他与熟练程序员Jonathan Sachs密切协作，互动地开发出所有细节层次上的设计，从最宽的体系结构到命令菜单的结构组织和菜单项命名。

他说有一次他的儿子问他在做什么工作，他没有办法给出一种好的回答——他不是程序员，但是他正在开发程序。另一方面他也不是经理，他干的是详细设计工作，而不是指挥其他员工。他干的是软件设计，但是这种工作没有一个头衔。本章重新给出了他的宣言。过去几年的其他几次谈话和写作中，他都令人信服地说明需要把软件设计看做一种“职业”，而不是经理或程序员的附属工作。

Kapor 1990年在汇集了微型计算机界领袖的个人计算机论坛Esther Dyson上提出了自己的宣言。人们对此褒贬不一，有人大力赞同，有人则提出“为什么你要抱怨——我们在销售很多软件吗？”一年以后，这个宣言发表在《Dr. Dobbs Journal》上（Kapor, 1991），