



普通高等教育
软件工程

“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

用户为中心的 设计与评估

张云鹏 武君胜 杨帆 ◎ 编著

*User-centered Testing
and Evaluation*



人民邮电出版社
POSTS & TELECOM PRESS



普通高等教育
软件工程

“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

用户为中心的 设计与评估

张云鹏 武君胜 杨帆 ◎ 编著

*User-centered Testing
and Evaluation*

人民邮电出版社

北京

图书在版编目 (C I P) 数据

用户为中心的设计与评估 / 张云鹏, 武君胜, 杨帆
编著. — 北京: 人民邮电出版社, 2014. 9
普通高等教育软件工程“十二五”规划教材
ISBN 978-7-115-36673-3

I. ①用… II. ①张… ②武… ③杨… III. ①软件设计—高等学校—教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2014)第186270号

内 容 提 要

本书既保持知识的普适性, 又保证前沿研究成果的新颖性, 由浅入深、循序渐进地介绍用户为中心的设计与评估及相关的编程基础知识, 使读者对以用户为中心的界面设计有一个完整的了解, 并在界面设计方面打下坚实的基础。

本书共 9 章, 主要内容包括: 用户为中心的设计与评估、用户客观直觉特性、用户主观直觉特性、快速原型方法、Visual Basic、Qt 界面开发、用户界面设计、启发式评估、出声思维可用式测试。

本书可作为高等学校计算机、软件工程等专业相关课程的教材, 也可作为软件工程领域从业人员及爱好者的参考用书。



-
- ◆ 编 著 张云鹏 武君胜 杨 帆
责任编辑 许金霞
责任印制 彭志环 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京中新伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 12 2014 年 9 月第 1 版
字数: 311 千字 2014 年 9 月北京第 1 次印刷
-

定价: 34.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前言

2003 年国家示范性软件学院成立不久,国内几所软件学院共同引进了美国卡耐基梅隆大学 (CMU) 的软件系统开发 (SSD, Software Systems Development Courses) 系列课程。CMU 的软件工程专业全球公认排名首位,由其主导开发的这套课程体系代表了软件工程专业的国际前沿水平,其中的 SSD4 (User-centered Testing and evaluation, SSD4 原名为 User-Centered Design and Testing)、用户为中心的设计与评估是全面介绍软件设计与评估相关知识的专业基础课程。经过十余年的吸收和本地化,该课程的教学思想已充分和中国的教育模式结合起来,今天,由原版英文教材过渡到中文教材的时机已经成熟。同时,也为了适应国外课程和国内教学情况的更好统一,我们在充分吸收 SSD4 相关知识的基础上,结合近年该领域最新的研究成果编写了本书,目的是让国外先进课程和国际前沿知识本地化,这样才能更好地为国内读者服务。

本书是在几位作者自 2003 年至今从事用户为中心的设计与评估课程教学实践、参考国内外多种教材的基础上编写而成的,编写过程中力求做到内容通俗流畅、简明扼要。书中对每个部分的内容从章节、实例、图表的精选都是按照步步启发的模式设计安排的,有利于施教者在教学过程中的发挥。

本书全面系统地介绍了 Visual Basic 和 Qt 编程语言的基础知识, 内容组织紧凑, 层次清晰, 由浅入深。本书各章节内容的设置可以有效提高读者的兴趣, 使他们在入门后不会感觉到后续课程较难。书中引入了大量 C/S、B/S 架构的 MIS 实例, 使读者可以快速地进行可视化开发网络应用程序、网络服务、Windows 应用程序和服务端组件。

本书的另外一部分内容是通过动机、观点、知觉、记忆、认知的处理,灵活的能力,错误和参考等方面对用户界面设计所涉及到的心理学进行了详细的描述,并提供了启发式评价和出声思维评价两种对界面可用性的测试方法。

为了保证讲义的先进性和实时性，本书包含了大量的网络链接，可以得到最新的网络资源。网络资源包含了书中涉及到的大部分内容，并补充了各种最新的人机界面的设计理念。广大读者可以通过这些链接获得相关领域最新、最丰富和最全面的资料。

本书的实践性内容是另外一个特色,通过各种方式将理论上抽象的教学内容与生活中的实际应用联系起来,做到了良好的理论与实践结合。对每一个启发式评价的内容都给出对应的实例。

全书共分9章,张云鹏组织编写了第2、3、5、8章,武君胜组织编写了第1、4章,杨帆组织编写了第6、7、9章,李勇协助编写了第6章,全书由武君胜统稿,焦会琴参与了后期的校对工作。在编辑出版过程中得到了 iCarnegie Dr. Cheng Song 的大力支持,在此表示衷心的感谢。

由于编者水平有限，书中难免存在不当和疏漏之处，恳请读者批评指正。

编者

2014年3月

目 录

第 1 章 用户为中心的设计和评估 ... 1	1.2 以用户为中心的设计.....3
1.1 设计思想的变迁 1	1.2.1 用户为中心的目标4
1.1.1 以机器为本的设计观念..... 1	1.2.2 用户为中心的设计流程6
1.1.2 以人为本的设计观念..... 2	1.3 用户为中心的设计原则.....8

用户分析篇

第 2 章 用户客观知觉特性 14	3.2.3 不存在对信息知觉理解的 唯一标准 31
2.1 用户视觉特性 14	3.2.4 知觉能够产生错觉.....31
2.1.1 基本视觉特性 14	3.2.5 知觉功能具有一定的生理和 心理特性限度 31
2.1.2 视觉系统 18	3.2.6 注意力是一个很有限的心理资源 32
2.1.3 视觉寻找图标的特性 20	3.3 记忆 32
2.2 用户听觉特性 24	3.3.1 关于记忆 32
2.2.1 基本听觉特性 24	3.3.2 人类大脑如何进行记忆 33
2.2.2 听觉中的注意 25	3.3.3 人类大脑记忆特性的划分 33
2.2.3 听觉的预处理和听觉系统 26	3.3.4 记忆过程的实现 33
2.3 用户知觉特性 27	3.4 思维 35
第 3 章 用户主观知觉特性 29	3.4.1 思维的概念 35
3.1 现实世界的需求 29	3.4.2 思维的类型 35
3.1.1 新手用户 29	3.4.3 思维的基本特征 36
3.1.2 平均用户（一般用户） 30	3.5 理解 37
3.1.3 偶然用户 30	3.5.1 理解的含义 37
3.1.4 专家用户 30	3.5.2 理解的对象与内容 37
3.2 知觉的非理性 30	3.6 交流与合作 38
3.2.1 不存在传统意义上的 “标准知觉特性” 30	3.6.1 交流的概念 38
3.2.2 不存在唯一的 “知觉标准参数” 31	3.6.2 人的几种合作行动方式 40

开发工具篇

第 4 章 快速原型方法 43	4.1.2 人机界面的迭代开发过程 44
4.1 迭代式设计过程 43	4.2 快速原型方法 46
4.1.1 迭代设计的概念 43	4.2.1 建立原型的目的 46

4.2.2 水平原型和垂直原型.....	47	6.7.1 主窗口.....	96
4.2.3 抛弃型原型和进化型原型.....	48	6.7.2 菜单.....	97
4.2.4 书面原型和电子原型.....	49	6.7.3 工具栏.....	97
4.3 原型评价.....	50	6.7.4 动作.....	98
第 5 章 Visual Basic.....	53	6.7.5 停靠窗口.....	98
5.1 使用 Visual Basic.....	53	6.7.6 对话框.....	99
5.1.1 初识 Visual Basic.....	53	6.7.7 交互式帮助.....	100
5.1.2 使用控件.....	62	6.7.8 多文档界面.....	100
5.1.3 编写代码.....	71	6.7.9 向导页.....	100
5.1.4 生成可执行文件.....	72	6.7.10 设置.....	101
5.2 变量和常量.....	73	6.7.11 多线程.....	101
5.2.1 变量.....	73	6.7.12 桌面集成.....	102
5.2.2 常量.....	75	6.8 Qt Designer.....	102
5.3 Visual Basic 程序.....	77	6.8.1 使用 Qt Designer.....	102
5.3.1 在代码窗口中编写源代码.....	77	6.8.2 Qt Assistant.....	103
5.3.2 使用选项对话框.....	80	6.8.3 GUI 应用程序示例.....	104
5.4 使用 Visual Basic 帮助文件.....	80	6.9 国际化.....	106
5.4.1 查看 Visual Basic 帮助文件.....	80	6.9.1 文本输入和显示.....	107
5.4.2 使用帮助菜单查看帮助.....	81	6.9.2 翻译程序.....	108
5.4.3 使用 F1 键.....	82	6.9.3 Qt Linguist.....	108
5.5 Visual Basic 与面向对象编程.....	83	6.10 布局.....	109
5.5.1 理解面向对象编程.....	83	6.10.1 内建布局管理器.....	110
5.5.2 完全面向对象的 Visual Basic7.....	84	6.10.2 嵌套式布局.....	110
第 6 章 Qt 界面开发.....	86	6.11 样式和主题.....	111
6.1 Qt 历史.....	86	6.11.1 内建样式.....	112
6.2 Qt 产品概述.....	87	6.11.2 窗体的样式表.....	112
6.3 Qt 的安装.....	89	6.11.3 自定义样式.....	113
6.4 Qt 开发社区.....	89	第 7 章 用户界面设计.....	114
6.5 窗体.....	90	7.1 用户界面的设计过程.....	114
6.5.1 内建窗体.....	90	7.1.1 开发一个 OOED 应用程序.....	114
6.5.2 定制窗体.....	92	7.1.2 构建用户界面.....	120
6.6 信号和槽.....	94	7.1.3 编码、测试、调试和编写文档.....	127
6.6.1 信号和槽示例.....	94	7.2 界面设计原则.....	130
6.6.2 元对象编译器.....	95	第 8 章 启发式评估.....	133
6.7 GUI 应用程序.....	95	8.1 启发式评估方法介绍.....	133

8.2 启发式评估准则	133	8.7.1 帮助和文档的区别	145
8.2.1 可见性准则	134	8.7.2 始终如一	146
8.2.2 预防准则	134	8.7.3 便于搜索	146
8.2.3 认识恢复准则	134	8.7.4 可用性评价报告实例	146
8.2.4 帮助准则	134	8.8 启发式评估：计算机系统与	
8.2.5 现实匹配准则	134	现实世界的匹配	147
8.2.6 统一准则	135	8.8.1 现实世界中的概念	147
8.2.7 自由准则	135	8.8.2 使用用户熟悉的语言	148
8.2.8 提示准则	135	8.8.3 现实世界中的约定习俗	148
8.2.9 高效准则	135	8.8.4 文化差异对计算机	
8.2.10 美观准则	135	系统造成的影响	148
8.3 如何完成可用性评价报告	136	8.8.5 可用性评价报告实例	149
8.3.1 报告标识	136	8.9 启发式评估：一致性和标准化	149
8.3.2 简要描述	136	8.9.1 不要对用户造成干扰	150
8.3.3 缘由	137	8.9.2 保持系统中的一致性	150
8.3.4 界面描述	137	8.9.3 保持应用程序的一致性	150
8.3.5 解释	137	8.9.4 一致性和标准的其他体现	150
8.3.6 好的可用性说明/不好的		8.9.5 可用性评价报告实例	151
可用性说明	137	8.10 启发式评估：用户的控制和自由	152
8.3.7 可能/潜在的解决方案	137	8.10.1 让用户去控制	152
8.3.8 同其他可用性评价报告的关联	137	8.10.2 撤销操作	152
8.4 启发式评估：系统状态的可见性	138	8.10.3 需要确认回复	153
8.4.1 提供反馈意见让用户		8.10.4 提供退路	153
了解进程所处的阶段	138	8.10.5 可用性评价报告实例	153
8.4.2 不要让用户去猜测	139	8.11 启发式评估：识别而不是记忆	154
8.4.3 可用性评价报告实例	139	8.11.1 长期记忆和工作记忆	154
8.5 启发式评估：防止错误	141	8.11.2 记忆和识别对界面设计的影响	154
8.5.1 预防错误的发生	141	8.11.3 可用性评价报告实例	155
8.5.2 对用户潜在破坏性操作的警告	142	8.12 启发式评估：适用性和效率	156
8.5.3 可用性评价报告实例	142	8.12.1 提供键盘快捷键操作	156
8.6 启发式评估：帮助用户		8.12.2 允许自定义	157
认识错误并从错误中恢复	144	8.12.3 可用性评价报告实例	157
8.6.1 帮助用户认识错误		8.13 启发式评估：美观和简明设计	158
并提供建议的解决方法	144	8.13.1 美观和简明设计	158
8.6.2 可用性评价报告实例	144	8.13.2 可用性评价报告实例	158
8.7 启发式评估：帮助和文档	145	8.14 小结	160

第 9 章 出声思维可用性测试..... 161

9.1 出声思维测试介绍	161
9.1.1 什么是出声思维可用性测试.....	161
9.1.2 出声思维测试的经验法则.....	164
9.2 出声思维测试实践	166
9.2.1 定义研究框架.....	166
9.2.2 选择观测对象.....	167

9.2.3 观测准备.....	168
9.2.4 给参与者介绍要执行的活动.....	170
9.2.5 执行测试活动.....	173
9.2.6 结果分析.....	175
9.2.7 根据结果重新设计.....	178
9.2.8 编写可用性分析报告.....	180
9.3 出声思维测试与启发式评价的比较.....	181

第 1 章

用户为中心的设计和评估

1.1 设计思想的变迁

1.1.1 以机器为本的设计理念

工业革命以来，产生了以前没有的价值观念：效率和利润。大多数机器的基本设计思想是无限提高机器效率。因此以机器功能为设计过程考虑的核心问题，首先确定机器功能的设计，然后根据剩下的选择余地再补充一个外部的操作部分。这样，操作部分必然受机器功能限制，只能跟随机器的功能变化。它迫使操作者必须遵循机器的行为方式、运行速度、精度和强度。人成为机器的一个附件，完成机器无法实现的输入功能和反馈控制功能。这种设计没有考虑使机器适应人的生理特性、心理特性和认知特性。这种设计价值观念叫“以机器为本”。

20 世纪 50 年代以来，计算机设计基本延续了这种设计思想。虽然从 20 世纪 80 年代中期西方开始认识到这个问题，尝试从人机界面去弥补这一问题，但是没有根本改变计算机控制人、制造商控制用户的特性。“以机器为本”的设计在当前计算机设计中的主要表现如下。

1. 把布尔逻辑运算作为主要的或唯一的思维方法

布尔（George Boole）认为人的思维不是随意的，而是按一定规则进行的，这一规则被称为计算（Computation）。它的基本目的是描述人脑的逻辑思维的规律。1854 年布尔发表了一篇论文《思维规律的研究》（A Investigation Of the Law of Thought），建立了计算机中的逻辑计算数学。他在该文中提出用计算的符号语言去表示和建立大脑的各种操作。他用代数型的运算方法建立了人类思维的演绎过程，并认为按照这一共同的推理方法，其结论就是正确的。它与推理内容无关，也不因各人解释而异，这种方法符合西方科学要求的“客观性”。这样他把逻辑真理的判断从内容含义转变到形式符号和连接符、转变到关系运算和抽象法则，使逻辑句法重于语义，从而创立了形式逻辑的发展方向。他认为，他的逻辑学“完全对应”人类思维时所使用的逻辑。他的长远目的是从逻辑和概率这些数学理论中找到思维规律，并确定人的思维规律。他的理论为计算机的发展起了重要作用。

例如，有人根据这种逻辑推理方式得出如下结论。

- （1）思维是一种计算。
- （2）图灵机可以计算任何可计算的函数。
- （3）数字计算机实现了图灵机。

因此，数字计算机可以思维。

这个结论明显不符合事实。计算机可以进行推理计算，但是计算机推理与人的思维方式不同。因此，对心理学的研究往往首先在哲学范围中进行。人是按照自己的理解去解释的，事实上可能完全是另外一个样子。

现在人们普遍明白了人脑思维过程远比这种逻辑复杂得多。人脑的逻辑思维过程包括这些逻辑元素，但是人脑还包括许多非逻辑的思维过程，例如，探索尝试性思维、临场发挥的想法、人脑的遗忘、思维的跳跃、因人而异的触景生情、思维出错后的修正、情绪对思维的支配等。不能把全部思维过程仅仅简化成逻辑运算。

2. 在计算机发展过程中一直存在两种不同的设计观念

第一种观念认为，计算机的计划和发展的目的是为了模仿人脑的思维功能。这种观念往往认为计算机与人的思维方式基本一致，因此人适应计算机就是适应自己，学习计算机操作是应该的。这种设计的最终目的是制造出像人脑一样的机器。

第二种观念认为，计算机仅仅是一种工具，是脑力劳动的一种辅助工具，称为认知工具。计算机的“思维”方式与人的不同，使用计算机给人造成很大的记忆负担、思维负担和精神压力，因此应该改进它的人机界面设计，使计算机操作适应人的生理和心理特性。

3. 强调用户必须学习计算机的操作方法

如今似乎已经成为一个法则，即要能使用计算机，必须对用户进行大量培训。计算机是人类发明的最复杂的大众机器之一，学习操作计算机需要花费大量的时间和精力。学习操作的主要目的是改变人的思维方式和行为方式。要记忆大量的计算机概念，了解计算机的行为特性，适应计算机的行为过程。

1.1.2 以人为本的设计观念

20 世纪 80 年代中期人们提出了“对用户友好”或“以人为本”的概念，以改进人机界面为主要目的。计算机内部按照机器特性去行为，人按照自己的方式去思维和行为，在这两者之间需要一个转换，把人的思维和行为转换成机器可以接受的方式，把机器的行为方式转换成人可以接受的方式，这个转换就是人机界面。然而要使计算机适应人，机器内部的结构和功能也需要改变，目前很难完全达到这一目标。

使计算机在人机界面上适应人的思维特性和行动特性，这种目的就叫“以人为本”的设计观念。迄今为止，实现这种设计目标主要以动机心理学、认知心理学、符号学为基础。动机心理学主要研究人行动的基本特性，例如，一个行动包含哪些基本因素，在设计人机界面时适应人的这些特性。认知心理学主要研究人脑力劳动的特性，例如，记忆、理解、语言交流等方面，在设计中，减少人的这些认知负担。符号学主要研究符号（文字和图形）的构成、符号的语义以及符号使用的基本特性。在设计中，使计算机符号符合人的表达和交流特性。

从 20 世纪 80 年代到 2000 年前后，人们研究的主要范围是研究人操作计算机时的知觉特性，减少用户记忆负担，减少学习操作的时间。例如，在屏幕上发现目标的过程中人的知觉有什么特性，显示多少个菜单项目比较符合人的知觉特性，知觉对文字的感知特性，键盘字母的排布方式对手操作有什么影响，鼠标形状会引起手腕的什么生理问题，各种键盘命令的构成方式对用户操作有什么影响，寻找取代键盘的方法，改进屏幕显示，把记忆操作命令的 DOS 系统改变为屏幕显示菜单的直接操作系统等。

改变屏幕上显示的菜单符号布局能够改善用户的一些操作特性。在改进计算机“思维”特性

以适应人的思维方面进展不大。用户的思维不得不适应计算机的“思维”方式，适应计算机的控制过程，这是引起用户负担的主要原因之一。如果不能明显减少思维负担，就没有从根本上改变“以机器为本”的状态，用户仍然受计算机的控制。人们逐步理解到，要把计算机的“思维”方式转变成人的思维方式，要把计算机的设计从“以机器为本”转到“以人为本”，仅仅改变表面屏幕上的东西是不够的，必然涉及计算机系统内部的功能与结构以及计算机语言的功能和结构。人机界面的设计研究正在促进这一根本变革。

目前，计算机技术和软件技术的快速发展，计算机的处理器速度、存储器容量得到飞速提高，使在其上运行复杂图形化人机系统成为可能。伴随着人机交互技术，如鼠标、窗口系统、超文本、浏览器等交互设备和环境的不断出现和发展，更多的领域和更多的人必须掌握计算机系统的使用。计算机也从 20 世纪 80 年代多人同时使用一台计算机的“主机时代”，迅速发展到了“一人一机”的 PC 时代；进而又快速发展到当今的“普适计算时代”。

在主机时代，计算机被视为高深莫测之物，只有在高端领域提供给超智和奇才使用。进入普适计算时代，计算机系统无处不在，已应用于各行各业，乃至个人和家庭生活的方方面面。计算机系统的发展及其广泛应用反过来又推动着计算机产品设计观念从“以机器为中心”不断向“以人为本”、“以用户为中心”的观念转变，从而从最早的“人适应计算机”转变为“计算机适应人”，如图 1-1 所示。

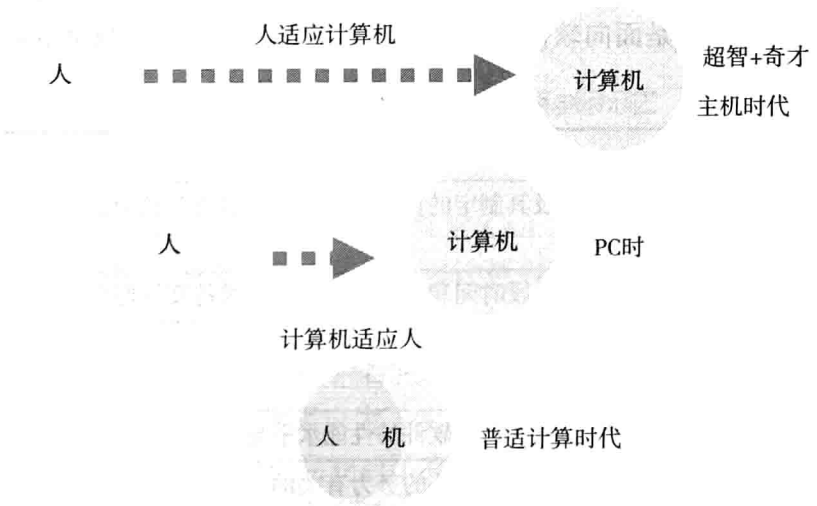


图 1-1 计算机系统的人机融合过程

1.2 以用户为中心的设计

以人为本的设计，也就是以用户为中心的设计（User-Centered Design, UCD），其最基本的思想就是将用户（user）时时刻刻摆在过程的首位。在产品的生存周期的最初阶段，产品的设计策略应当以满足用户的需求为基本动机和最终目的。在其后的产品设计和开发过程中，对用户的研究和理解应当被作为各种决策的依据，同时，在产品开发的各个阶段的评估信息也必须有来源于用户的反馈。所以，用户的概念是整个设计思想的核心，我们有必要在这里首先解释一下用户的概念。

简单地说，用户是指使用某产品（本书主要是指计算机软件产品）的人。这其中包含两方面的含义。

（1）用户是人类的一部分。用户具有人类的共同特性，包括视觉、听觉和嗅觉等客观的知觉特性，以及如记忆、思维、理解、学习和交流等主观认知特性，所有用户在使用任何产品时都会在各个方面反映出这些特性。人的行为不仅受到视觉和听觉等感知能力、分析和解决问题的能力、记忆力和对外界刺激的反应能力等人类本身具有的基本能力的影响，同时，人的行为还时刻受到个人心理和性格取向、物理和文化环境、教育程度以及以往经历等个体主观因素的制约。

（2）用户是产品的使用者。以用户为中心的设计研究的人是与产品使用相关的特殊群体，他们可能是产品的当前使用者，也可能是未来的、甚至是潜在的使用者。这些人在使用产品的过程中的行为也会与一些与产品有关的特征紧密相关。例如，对于目标产品的知识、期待利用目标产品所完成的功能、使用目标产品所需要的基本技能、未来使用目标产品的时间和频率等。

因此，以用户为中心，就应当从研究用户的人类一般属性和与产品相关的特殊属性着手。关于前者我们将在用户分析篇中论述，以下重点讨论产品（主要是计算机软件产品）的质量属性。

1.2.1 用户为中心的目标

关于软件质量的国际标准和国家标准规定了软件的如表 1-1 所示的 6 个质量特性。由 McCall 等人 1979 年提出 McCall 软件质量模型将软件质量的概念建立在 11 个质量特性之上，如表 1-2 所示。而这些质量特性分别是面向软件产品的运行、修正和转移的，具体如图 1-2 所示。

表 1-1 国际标准和国家标准规定的质量特性及其含义	
质量特性	含 义
功能性（Functionality）	与一组功能及其制定的性质有关的一组属性。这里的功能包括满足明确或隐含的要求
可靠性（Reliability）	与在规定的一段时间和条件下，软件能维持其性能水平的能力有关的一组属性
易用性（Usability）	与一组规定或潜在的用户为使用软件所需做的努力和对这样的使用所做的评价有关的一组属性
效率（Efficiency）	与在规定的条件下，软件的性能水平与所使用资源量之间关系有关的一组属性
可维护性（Maintainability）	与进行指定的修改所需的努力有关的一组属性
可移植性（Portability）	与软件可从某一环境移植到另一环境的能力有关的一组属性

表 1-2 McCall 提出的 11 个质量特性及其含义	
质量特性	含 义
正确性	在预定环境下，软件满足设计规格说明及用户预期目标的程度。它要求软件本身没有错误
可靠性	软件按照设计要求，在规定时间和条件下不出故障，持续运行的程度
效率	为了完成预定功能，软件系统所需的计算机资源的多少
完整性	为某一目的而保护数据，避免它受到偶然的或有意的破坏、改动或遗失的能力
易用性	对于一个软件系统，用户学习、使用软件及为程序准备输入和解释输出所需工作量的大小
可维护性	为满足用户新的要求，或当环境发生了变化，或运行中发现了新的错误时，对一个已投入运行的软件进行相应诊断和修改所需工作量的大小

续表

质量特性	含 义
可测试性	测试软件以确保其能够执行预定功能所需工作量的大小
灵活性	修改或改进一个已投入运行的软件所需工作量的大小
可移植性	将一个软件系统从一个计算机系统或环境移植到另一个计算机系统或环境中运行时所需工作量的大小
可复用性	一个软件（或软件的部件）能再次用于其他应用（该应用的功能与此软件或软件部件的所完成的功能有关）的程度
互连性	又称相互操作性。连接一个软件和其他系统所需工作量的大小。如果这个软件要联网或与其他系统通信或要把其他系统纳入到自己的控制之下，必须有系统间的接口，使之可以联结

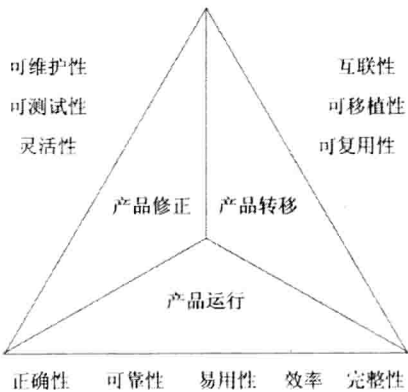


图 1-2 McCall 软件质量模型

从表 1-1 和表 1-2 可以看出，无论是国际标准和 McCall 软件质量模型，均把可用性作为软件产品的质量属性，且除了可使用性外，所有的质量属性更多地体现了软件自身的特征，而只有可用性是依赖与使用产品的用户。不难想象，无论一个软件产品功能多么完善，性能效率多么优越，可靠性多么高，可维护性和可移植性多么好，若它的可使用性或称为可用性不能让使用者满意，则不能说此软件产品是高质量产品。因为用户为了使用该软件，有可能花费很多时间学习其操作、准备数据，做许多步骤的操作，才用该产品完成了有关的工作。

从上述论述就不难得出用户为中心（为方便论述，以后简称 UCD）的目标是什么。让用户满意是 UCD 的主要目标。这其中最重要的一点就是可用性。

可用性通常的定义是：特定用户能够在特定环境下使用产品有效、迅速并满意地完成特定任务。产品的可用性由用户在特定环境下使用产品后决定。换句话说，一个产品的可用性特征由三部分决定，包括产品能否帮助用户完成特定的目标、产品对于执行任务的效率以及用户对使用产品的满意程度。

在用户看来，可用性仅仅是指他们和产品或系统进行交互的质量。事实上，产品的大部分属性都和可用性有关，比如功能、性能、可靠性、已安装性、可维护性、是否提供详细的文档以及是否提供相应的服务。也就是说，用户对产品可用性的满意程度在很大程度上支配着他们对所有这些属性的看法。从这个意义上讲，尽管产品的每个质量属性都很重要，而真正要做到高质量，必须首先可用性高，让用户满意，这样，产品其他高质量特性才能最终得到体现。这也就是为什么各行各业都十分重视可用性设计并应用 UCD 思想的根本原因。

Thomas K Landauer 在 1996 年撰文认为，计算机技术领域大量产品的应用并没有促使生产力的大幅提高，主要原因是因为计算机系统不容易使用。因此，我们必须解决可用性问题，而 UCD

的主要目的就是解决可用性问题。计算机领域分类广泛，UCD 与计算机硬件、软件及相关服务的联系较为密切。而且，UCD 不仅可用来开发销售量大的商业软件，还适用于开发面向某类用户的产品、定制产品或公司内使用的系统。

UCD 的使用范围很广，从世界级大公司使用的高端大型计算机到学校或家庭使用的多媒体电脑等均有涉及。大量可用于设计高档笔记本电脑、台式机、服务器，以及应用软件、数据库、网络和 Web 系统的开发。

尽管本书主要讨论计算机软件产品的 UCD 思想，其实 UCD 适用于一切项目的设计，从设计烤炉到核电站设备控制室，从家电顶盒到飞行器控制系统等，它无所不及，只不过它在计算机技术领域能更好地发挥其可用性优势。

1.2.2 用户为中心的设计流程

任何产品从前期调研、构想、研究、总体设计直到具体设计、制造和实施，都要经历一个复杂的过程。对于计算机软件产品，无论采取何种开发模型都必须经历可行性分析、需求分析、概要设计、详细设计、编码、测试、部署运行和维护服务等软件生存周期的各个阶段。高可使用性软件产品的质量是在其生存周期的各个阶段经过不断设计和完善逐渐形成的。

虽然，可行性分析以及产品部署和维护服务环节十分重要，但一般来说，软件产品质量属性的形成过程主要集中在需求分析到测试的五个阶段。抛开每一个产品的设计开发过程中各自的特殊性，典型的用户为中心的产品设计流程可描述为图 1-3 所示过程，它分为 3 个主要阶段。

- (1) 策略和用户分析。
- (2) 设计和评估。
- (3) 实施和评估。

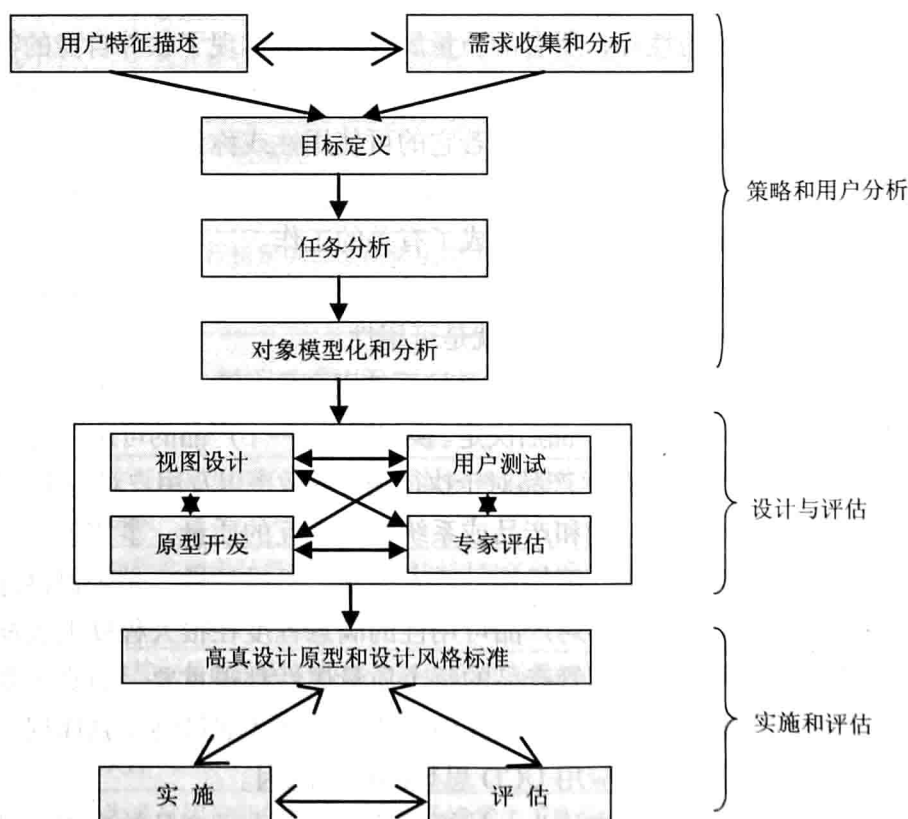


图 1-3 用户为中心的产品设计流程

本书后续章节将系统讨论 UCD 设计思想在以上 3 个主要阶段的具体应用,包括用户分析篇、设计工具篇和分析评估篇。用户分析篇分用户客观知觉特性和用户主观认知特性等分析内容,简要论述对用户共性特性的分析、具体项目用户的分析,为制定具体设计策略奠定基础;设计工具篇着重介绍支持 UCD 设计的几个开发工具,包括快速原型方法、Visual Basic 和 Qt 环境工具,它们为依据需求和用户分析设定的产品目标和任务实现提供了强大的快速实现工具,以便支持设计原型的形成,为设计评估提供仿真或保真原型;分析评估篇除对具体设计过程提供方法论外,重点论述设计过程中有效的启发式设计和评估方法,以及在产品交付给用户前使用的出声思维(Thinking-aloud)评估方法。本章以下小节将对上述 3 个阶段的主要特征做宏观描述。

1. 策略和用户分析

策略分析 (Strategy Analysis) 和用户分析 (User Analysis) 阶段着重解决的问题是决定产品设计的方向和预期目标。以用户为中心的设计思想认为,产品的成败最终取决于用户的满意程度。要达到用户满意的目标,首先应当深入而明确地了解谁是产品的目标用户 (Target User)。产品的设计者主要关心的不是这些用户的姓名,而是目标用户群体区别于一般人群的具体特征,例如,特定年龄区间、特殊的文化背景等。这一过程就是用户特征描述 (User Profiling)。同时,产品设计者还应当明确地了解目标用户对被设计产品的各方面期望 (Expectation) 是什么,包括用户希望使用的功能或达到目标的指标等。这一过程就是需求收集 (Requirement Collection) 和需求分析 (Requirement Analysis)。用户需求的数据和信息可以来源于用户试验和市场分析资料等多种渠道。

用户特征描述与需求收集和需求分析可以同时交叉进行并且互相受益。在一个产品周期的最初阶段,产品开发者往往对于设计产品的基本情况有一个大致的轮廓。由产品的基本性质就可以大致辨别出目标用户的最明显的群体特征。在与用户进行进一步的交流后,用户需求的情况得以不断具体化,同时根据用户需求的分布情况,又可以进一步挖掘出更准确、具体的用户特征。用户特征描述和需求分析是以用户为中心的设计过程的基础。只有全面扎实地做好这两方面的分析才能使整个设计有的放矢。在产品设计和开发的全部过程中,用户参与活动都将以用户特征描述和用户需求作为依据。

由于人力、物力、时间等资源的限制,一个产品往往不可能同时满足所有用户的所有需求,并且不同用户需求之间往往还有互相矛盾、互相排斥的情况。所以,设计和开发人员在全面分析用户和需求后,需要根据自身条件将项目的应用范围加以限制,并且同时将项目目标正规化,这就是目标定义。一个项目目标的具体内容往往不可能用几句话就可以概括,不同的目标按照其层次和逻辑关系可以组织为一个金字塔结构 (Hierarchical Structure)。用户所提出的需求和期望大多可以纳入目标金字塔结构 (Goal Hierarchy)。

产品设计目的是帮助用户完成他们期望完成的任务。在确定了项目目标后,产品支持用户完成的任务也就随之相对确定。这时用户产品设计和开发人员就可以将注意力集中在用户完成任务的具体行为方式上。产品设计的逻辑应当与用户完成任务的习惯或自然理解相吻合,这样,用户才能以最快的速度、最轻松地掌握系统的使用。任务分析的目的就是采用系统的用户研究方法,深入理解用户最为习惯的完成任务的方式。任务分析的数据来源于用户试验。在试验中,用户研究人员用观察、讨论、提问等方式从用户代表处获得各种与完成任务有关的信息,然后将这些信息归纳整理后用图示、列表、叙述等各种方式直观、清晰地表达出来,作为系统设计的指导。

2. 设计和评估

全面的策略和用户分析为产品的设计提供了丰富的背景素材。这些素材必须通过系统的方法

进行分析，并且以精练的方式表达出来才能被有效运用。一种常用的分析方法是对象模型化（Object Modeling）。对象模型化将所有策略和用户分析的结果按讨论的对象进行分类整理，并且以各种图示的方法描述其属性、行为和关系。这种方法类似于面向对象的分析方法，但是侧重于归纳与系统设计有关的信息而不求对系统的描述面面俱到。

对象的抽象模型可以逐步转化为不同具体程度的用户界面视图。比较抽象的视图有利于逻辑分析，比较具体的视图更接近于系统人机界面的最终表达。根据视图表达方式的具体程度，比较抽象的视图又被称为低真视图（Low-fidelity Prototype），比较具体的视图又被称为高真视图（High-fidelity Prototype）。

在设计不同具体程度视图的过程中，设计人员应当经常吸收各种渠道的反馈信息，避免闭门造车。收集反馈信息最常用的方法是用户测试（User Testing）和专家评估（Expert Evaluation）。用户测试法是指将设计的视图展现在目标用户面前，通过让用户模拟使用或讨论等方法获得用户反馈的数据。专家评估法是指设计人员请人机界面设计和系统功能的专家，根据他们的经验审查设计的视图，提出设计可能存在的可用性问题。用户测试法能够直接发现用户使用的问题，但是往往成本相对较高，周期较长。专家评估法容易管理，用时较短，同时可能会发现一些比较深层次的问题。但是，由于专家的背景从根本上不同于用户，所以研究结果可能与用户的直接反馈意见有不同程度的偏差。所以，虽然设计人员往往根据当时资源等情况决定使用用户测试法或专家评估法得到反馈意见，但这两种方法从根本上是不能互相替代的。

3. 实施和评估

随着产品进入实施阶段，产品开发人员投入越来越多的时间和精力，对高真设计原型进行最后的调整，并且撰写产品的设计风格标准，产品各个部分风格的统一性就是由设计风格标准保证的。

产品实施完成并在投放市场之前，通过出声思维式的实验室可用性测试方法对产品进行评估，找出产品在可用性方面（也包括功能等其他方面）存在的不足，反复修改并改正错误和不足，以使产品更为完善。

产品投放市场后进入运行和维护阶段，设计人员往往仍会发现各种各样的新问题或用户的建议，收集和处理这些信息不仅有利于当前产品的销售或运作，也有利于下一代产品的研制和开发。所以，产品的实施或投放市场完全不是以用户中心的设计过程的终止。从某种意义上讲，这时候甚至仍然可以理解为设计的一个特殊的阶段，以上讨论的在设计过程中应用的评估方法依然适用。特别要提出的是，在这一阶段，实验室可用性测试及用户调查表的用户研究方法的使用尤其有效。这些评估的目的是保证产品实施的质量，跟踪用户使用情况和满意程度，收集用户在使用中遇到的问题和建议并且随时解决产品的问题。

1.3 用户为中心的设计原则

图 1-3 中所标示的流程是经过抽象后简化的典型步骤，在实际设计应用过程中，各个步骤之间根据需要经常进行合并和转化。此外，仅遵循这个流程还远远不够，要达到 UCD 方法的目标，从根本上实现从“以机器为中心”的传统设计观念到 UCD 设计观念的转变，还必须坚持 UCD 设计的六大原则。

1. 确定商业目标

确定市场目标、目标用户和主要竞争对手不仅是一切设计的关键，也是制定用户参与策略的关键。

本原则要求定位目标市场，清楚在这个市场中的所有用户及他们的特点，并且清楚大部分目标用户当前采用的解决方案，从而掌握产品竞争形势。UCD 必须符合公司的商业策略并能明显提升经济价值。必须在项目一开始就得到这些信息。如果没有这些信息或信息不完全，那么接下来的用户中心设计活动是否有效就值得怀疑了。

2. 了解用户

了解用户是所有设计的动力源泉。这个原则是 UCD 的基础。UCD 设计的基础在于对客户当前及以后所希望的工作方式的正确把握。

为了了解客户，设计小组就需要知道客户现在和以后需要完成的任务、客户使用何种工具完成任务、在使用工具时遇到了一些什么问题以及他们完成任务的工作环境有什么主要特点（例如，他们主要是在公司一起工作还是在途中工作）。因此，UCD 的一个关键步骤是了解所有目标用户群。让每位参与设计的人员都清楚目标用户的信息，这有助于确保最终的设计确实以用户为中心。通过公司内部交流实现所有相关人员对目标用户信息的不断了解。

3. 设计总体客户体验

设计总体客户体验原则要求 UCD 多学科小组在设计中考虑客户所能看到、听到和触及的一切，考虑客户体验中的方方面面，UCD 的关键要素是设计一个整体解决方案。它设计开发的产品应该具有以下属性。

- （1）易于购买。
- （2）易于安装。
- （3）易于学习。
- （4）易于使用。
- （5）直观。
- （6）吸引人。
- （7）有用。

读者可能要问，为什么 UCD 方法要考虑购买这个因素？原因是产品的广告和包装等会给客户留下第一印象。当用户在看到这些时，便会建立自己对产品设计的看法，例如，这个产品大致应该有什么特征，然后考虑是否购买。如果你的产品广告和包装做得不贴近实际，用户看了之后不知道产品是什么，有什么特性，甚至产生一种抵触情绪，那么即使产品设计做得再好也在用户实际接触之前大打折扣。

那么为什么还要有直观和吸引人这两个属性。人们一度认为可用性就是指用户使用产品时不存在大的问题。但是，现在市场竞争越来越激烈，可用性还要求产品能方便用户使用并提供一些吸引用户的、赏心悦目的特性。

为什么要设立多学科专家小组进行设计呢？理由十分简单，由于我们产品面向的用户形形色色，各不相同，一个人或同一学科的几个人组成的设计小组很难完全预期自己设计的可用性。毕竟设计者大多是技术专家，但并不是用户，一个人或某几个人并不完全知晓用户，很容易按自己的想法设计系统和产品。因此，必须由来自不同学科的专家共同参与设计，按照小组成员对目标用户的掌握情况，从不同的侧面依照对用户的了解进行产品设计，并采用快速原型方法展示设计模型，直接交给用户体验和评价，通过多次迭代获得更多的评价信息，以便完善产品设计。