

<<面向使用的软件设计>>

图书基本信息

书名：<<面向使用的软件设计>>

13位ISBN编号：9787111345756

10位ISBN编号：7111345754

出版时间：2011-6

出版时间：机械工业出版社

作者：（澳）康斯坦丁 等著，刘正捷 等译

页数：423

译者：刘正捷

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<面向使用的软件设计>>

内容概要

《面向使用的软件设计》是一本经典著作，曾荣获1999年美国软件开发杂志的jolt大奖。书中主要介绍一种以使用为中心的软件开发方法论的模型和方法，用以帮助程序设计人员开发出更可用的软件，使用户能够更容易、更高效地完成他们的任务。

本书的两位作者有着丰富的软件工程实践和管理经验，被公认为软件开发人类因素方面的权威，是国际著名的咨询专家、培训教师和作者。

他们在书中并不只是给出好的设计准则，而是循序渐进地指导读者去进行一个系统化的软件开发过程。

通过大量的软件应用实例和案例分析，作者描述了在实践中怎样做是可行的，哪些东西是有实用价值的，因此具有很高的实用性。

本书文字简洁、论述精辟、层次清晰，适合计算机软件和应用程序的各类设计开发人员阅读，对于高等院校计算机及相关专业的学生以及软件学院的学生， also 具有很高的参考价值。

<<面向使用的软件设计>>

作者简介

Larry L. Constantine, 悉尼理工大学计算机科学系教授, Constantine & Lockwood 有限公司研究与开发主任, 他是最早从事结构化设计以及现代软件工程理论与实践的许多核心概念和模型研究的开拓者, 在他本人近四十年的职业生涯中, 共出版了10部著作, 发表了130多篇论文与文章, 对计算机科学和人类科学做出了重要的贡献, 他作为一位国际公认的软件人类因素方面的权威, 曾以咨询专家和培训教师的身份在全球19个国家传授知识, 并在许多国际会议上做大会主题演讲, 他曾担任软件开发管理会议的主席, 负责编辑Software Development 期刊的管理论坛, 并在IEEE Software、Communications of the ACM、ITC Journal 等刊物担任编辑委员会委员, 他毕业于麻省理工学院SLOA管理学院, 曾在6所主要大学任教,

Lucy A. D. Lockwood, Constantine & Lockwood 有限公司总裁, 是以使用为中心的设计方法的合作开发者, 也是操作概况概念的提出者, 作为一位咨询专家、培训教师和撰稿人, 她在程序设计和项目管理方面有着15年以上的经历, 她侧重于软件可用性以及在软件工程过程中把人与技术进行整合的实用方法的研究, 她曾是许多国际会议上最受欢迎的演讲者, 在软件开发会议和Web设计与开发会议上担任可用性与用户界面设计分组的主席, 还在企业应用开发会议上担任大会主席, 她发表的论文涉及用户界面设计、跨功能的团队工作和项目管理等领域, 她毕业于塔夫茨大学, 曾在悉尼理工大学做访问学者,

Constantine & Lockwood 有限公司 (<http://www.foruse.com>) 是一个咨询与培训机构, 其客户包括世界范围内的一些知名大企业, 该公司专门从事针对软件可用性、高绩效团队工作、组织机构文化和变化管理等领域的有关技术与过程的咨询。

<<面向使用的软件设计>>

书籍目录

译者序

前言

作者介绍

第一部分 开发更可用的软件

第1章 面向使用的软件开发方法：使用、可用性及用户界面

1.1 提高可用性

1.1.1 人类的特性

1.1.2 使用质量

1.1.3 可用性的经济学

1.1.4 营销信息技术

1.2 如何改进可用性

1.2.1 测试、再测试

1.2.2 流行风格

1.2.3 反复建造

1.2.4 审视一下

1.2.5 专家的见地

1.3 事过境迁

第2章 培育可用性：以使用为中心的设计方法论

2.1 面对用户

2.2 以使用为中心方法论的组成元素

2.2.1 主要原理

2.2.2 模型与建模

2.2.3 开发过程

2.2.4 反复改进

2.2.5 质量度量

2.3 驱动模型

2.3.1 诱人的抽象方式

2.3.2 回归基本

2.3.3 界面体系结构

2.3.4 多视角

2.3.5 一般化和简单化

2.4 协同活动

2.4.1 灵活的阶段安排

2.4.2 秩序与混沌

第3章 总体原则：以使用为中心设计方法的准则和原理

3.1 对话式的设计过程

3.2 准则和原理

3.3 可用性准则

3.3.1 第一条准则：易掌握准则

3.3.2 第二条准则：功效准则

3.3.3 第三条准则：渐进准则

3.3.4 第四条准则：支持准则

3.3.5 第五条准则：环境准则

3.4 用户界面设计原理

3.4.1 结构原理

<<面向使用的软件设计>>

3.4.2 简单性原理

3.4.3 可见性原理

3.4.4 反馈原理

3.4.5 宽容原理

3.4.6 重用原理

3.5 其他准则

3.6 细节、细节、细节

第二部分 可用性基本模型

第4章 用户及有关的用户类型：对用户 和用户角色的理解

4.1 使用和用户

4.2 真实用户及其他有关方面

4.2.1 虚拟用户

4.2.2 用户代理人

4.2.3 信息提供者和解释者

4.2.4 间接信息来源

4.3 用户角色模型

4.3.1 角色建模

4.3.2 焦点角色

4.4 用户角色关系图

4.4.1 类似

4.4.2 分类

4.4.3 组成

4.5 用户角色的运用

4.6 结构化角色模型

4.6.1 角色内容

4.6.2 建立结构

4.6.3 与后续内容的关系

第5章 工作结构：基于基本用例的任务建模

5.1 工作、工作、工作

5.2 任务建模

5.2.1 从场景到用例

5.2.2 基本用例

5.2.3 抽象和一般化的程度

5.3 用例图

5.3.1 特例化

5.3.2 扩展

5.3.3 组成

5.3.4 类似

5.3.5 焦点用例

5.4 基本用例模型的建立

5.4.1 用例的识别

5.4.2 用户与用例

5.4.3 用例表述的编写

5.5 应用

第6章 界面体系结构：界面内容与导航

6.1 工作场所

6.2 界面内容

<<面向使用的软件设计>>

- 6.2.1 内容建模
- 6.2.2 内容列表
- 6.2.3 设置环境
- 6.2.4 工具和材料
- 6.2.5 内容建模过程
- 6.3 环境导航图
 - 6.3.1 行为视图、体系结构视图和顺序视图
 - 6.3.2 导航图的使用
- 6.4 应用

第三部分 创建可视化设计

第7章 对话设计：布局和交流

- 7.1 从抽象到具体
- 7.2 交流渠道
 - 7.2.1 词语
 - 7.2.2 彩色界面
 - 7.2.3 标记和符号
 - 7.2.4 声音
- 7.3 屏幕空间
 - 7.3.1 工作流
 - 7.3.2 发布信息
 - 7.3.3 视觉组织

第8章 实现窗口部件：可视组件的选择与设计

- 8.1 拿来还是自制
- 8.2 图标交流
 - 8.2.1 “视觉”设计
 - 8.2.2 符号学
 - 8.2.3 工具栏的组织
- 8.3 菜单
 - 8.3.1 菜单的组织
 - 8.3.2 键盘的访问
 - 8.3.3 右键快捷菜单
 - 8.3.4 快餐菜单
- 8.4 选择性部件的选择
 - 8.4.1 多项中选择一个
 - 8.4.2 多项中选择几个
 - 8.4.3 特例
 - 8.4.4 用菜单作为选择部件

第9章 创新界面：创新界面工程与定制组件

- 9.1 创新工程
 - 9.1.1 为什么创新
 - 9.1.2 创新什么
- 9.2 创新过程
 - 9.2.1 迭代创新
 - 9.2.2 两者兼备的设计
 - 9.2.3 想象它
 - 9.2.4 大幅度改进
- 9.3 指导性界面

<<面向使用的软件设计>>

- 9.3.1 提供与限制
- 9.3.2 可视对象持续性
- 9.3.3 习惯扩展
- 9.3.4 移动界面
- 9.4 创新的应用
- 9.4.1 滚卷工作
- 9.4.2 重载

第四部分 完成设计

第10章 解决方案表述：实现建模和原型

- 10.1 有趣的东西
- 10.2 原型和原型建造
- 10.2.1 原型种类
- 10.2.2 被动原型
- 10.2.3 主动原型
- 10.2.4 创建界面原型
- 10.3 模型间的映射
- 10.3.1 界面环境
- 10.3.2 界面内容
- 10.3.3 界面组合
- 10.4 建立实现模型的演示
- 10.4.1 扩展程序设计之一
- 10.4.2 扩展程序设计之二

第11章 尽可能提供帮助：帮助与帮助消息的设计

- 11.1 即使是专家也需要帮助
- 11.2 寻求帮助的用例
- 11.2.1 帮助用例
- 11.2.2 通过用例来组织帮助
- 11.3 访问和呈现技术
- 11.3.1 显示帮助
- 11.3.2 访问帮助
- 11.4 特殊技巧和方式
- 11.4.1 声音和闹剧
- 11.4.2 辅导
- 11.4.3 软件魔术
- 11.5 写出有帮助的东西
- 11.5.1 帮助风格要素
- 11.5.2 文档，还是帮助
- 11.5.3 有效的帮助
- 11.6 有帮助的信息
- 11.6.1 消息和媒介
- 11.6.2 消息模型
- 11.6.3 错误预防
- 11.6.4 抵制冲动
- 11.6.5 编写消息

第12章 初学只有一次：支持用法模式的演变

- 12.1 初学者阶段之后
- 12.2 滑雪场式的界面

<<面向使用的软件设计>>

- 12.3 渐进用法
 - 12.3.1 新手用法
 - 12.3.2 中级用法
 - 12.3.3 熟练用法
 - 12.3.4 用法概况
- 12.4 支持性界面
 - 12.4.1 获得设施
 - 12.4.2 转变设施
 - 12.4.3 生产设施
- 12.5 为渐进用法而设计
 - 12.5.1 映射快捷键
 - 12.5.2 转变帮助
 - 12.5.3 可修改界面和自适应界面
 - 12.5.4 连续不断的定制
 - 12.5.5 对特性的访问
- 12.6 渐进用法的应用
- 第13章 在适当的位置：符合操作环境
 - 13.1 不和谐的环境
 - 13.2 操作建模
 - 13.3 与环境相适应
 - 13.4 绑定环境
 - 13.4.1 担负职责概况
 - 13.4.2 交互概况
 - 13.4.3 信息概况
 - 13.5 环境概况
 - 13.5.1 设备约束概况
 - 13.5.2 操作风险概况
 - 13.6 把环境落到实处
 - 13.6.1 dimtel产品订单录入
 - 13.6.2 bankinvest投资分析
- 第14章 相同的游戏，不同的地方：特殊的应用，特殊的问题
 - 14.1 依然是主题和变化
 - 14.2 面向使用的web设计
 - 14.2.1 目的概况
 - 14.2.2 吸引点击
 - 14.2.3 相同却又不同
 - 14.2.4 web的悲哀
 - 14.3 应用web的智慧
 - 14.3.1 合理使用faqs
 - 14.3.2 qubit的改造
 - 14.3.3 修改过的建模
 - 14.4 嵌入式系统应用
 - 14.4.1 硬件和软件
 - 14.4.2 界面问题
 - 14.4.3 嵌入式应用程序建模
 - 14.4.4 嵌入式界面
 - 14.5 其他特殊界面

<<面向使用的软件设计>>

14.5.1 声音界面

14.5.2 打电话给技术支持

第15章 以使用为中心设计方法的应用：teleguida

15.1 规模放大

15.2 电话标签

15.3 收集需求

15.4 teleguida用户和使用

15.4.1 teleguida的用户角色

15.4.2 teleguida的基本用例

15.4.3 teleguida的基本用例表述

15.5 建立teleguida原型

15.5.1 teleguida的内容模型

15.5.2 teleguida的导航图

15.5.3 teleguida可视化设计

第五部分 评估与改进

第16章 改进可用性：检查与评审

16.1 评估可用性

16.1.1 专家评估

16.1.2 同行评审

16.1.3 用户评审

16.1.4 测试与度量

16.1.5 检查

16.2 检查方法

16.2.1 经验式评估

16.2.2 认知走查

16.2.3 多元化可用性走查

16.3 协同可用性检查

16.3.1 目标

16.3.2 可用性缺陷

16.3.3 焦点

16.3.4 检查角色

16.3.5 检查过程

16.4 焦点式检查

16.4.1 一致性检查

16.4.2 符合性检查

第17章 量化：在实践中衡量可用性

17.1 量化式比较

17.2 可衡量的质量

17.2.1 度量指标的运用与滥用

17.2.2 偏好度量指标

17.2.3 绩效度量指标

17.2.4 预测度量指标

17.3 用户界面设计度量指标

17.3.1 度量什么

17.3.2 衡量标准

17.4 一组基本的可用性指标

17.4.1 基本效率

<<面向使用的软件设计>>

17.4.2 任务符合度

17.4.3 任务可见度

17.4.4 布局统一度

17.4.5 视觉相关度

17.5 度量指标运用

第18章 测试评分：实验室与现场可用性测试

18.1 测试的发展进程

18.2 两种可用性测试形式

18.2.1 实验室里的科学

18.2.2 现场测试

18.2.3 beta测试

18.3 测试协议

18.3.1 不适当的影响

18.3.2 说出自己的想法

18.3.3 测试前后

18.3.4 绩效度量指标

18.4 测试策略

18.4.1 规划目标

18.4.2 选择测试用户

18.4.3 测试用例

18.5 为什么要测试，为什么不要测试

第六部分 过程的组织和管理

第19章 编程实现：界面的实现

19.1 对象和界面

19.1.1 用于实现的对象

19.1.2 面向对象的用户界面

19.1.3 表面上的对象

19.1.4 对象体系结构

19.2 快速开发

19.2.1 集中建造

19.2.2 结构迭代

19.3 可视设计的可视开发

第20章 对用户的使用：开发过程中的用户

20.1 对用户的使用或滥用

20.1.1 开发周期中的用户

20.1.2 以使用为中心设计中的用户

20.2 需求对话，需求舞蹈

20.3 追根溯源

20.3.1 面对面

20.3.2 会议机制

20.3.3 观察工作

20.4 对用户的使用

20.4.1 对预期的管理

20.4.2 对付混乱

20.4.3 用户和原型

20.5 联合基本建模

20.5.1 联合运用

<<面向使用的软件设计>>

- 20.5.2 从jad到jem
- 20.5.3 参与者角色
- 20.5.4 活动概览
- 20.5.5 框架会议
- 20.5.6 建模会议
- 20.5.7 检查会议

第21章 如何组织：在更大应用环境中的可用性

- 21.1 组织部门
- 21.2 标准和风格指南
 - 21.2.1 标准的变更
 - 21.2.2 可用的可用性标准
 - 21.2.3 背离标准
- 21.3 竞争性群体
 - 21.3.1 市场，市场
 - 21.3.2 市场研究
 - 21.3.3 市场和使用
 - 21.3.4 当顾客不对时
- 21.4 专家和专长
- 21.5 符合文化
 - 21.5.1 集成可用性
 - 21.5.2 后期补救
 - 21.5.3 把真理分解开

附录a 推荐阅读书目

附录b 使软件更可用的11种途径：软件可用性基本原理

附录c 词汇表

附录d 以使用为中心设计方法所用的若干表格

附录e 软件可用性主观衡量（suss）

参考文献

<<面向使用的软件设计>>

章节摘录

版权页：插图：

<<面向使用的软件设计>>

编辑推荐

《面向使用的软件设计》：为了获得高质量的软件，开发者们长期以来一直不懈地改进其产品的内部体系结构。

现在最早创立结构化设计以改进内部体系结构的Larry L.Constantine与著名咨询专家Lucy A.D.Lockwood一起，把软件开发的焦点转向外部体系结构。

在《面向使用的软件设计》中，他们介绍了一种革命性的软件开发方法论的模型和方法，可以帮助程序设计人员开发出更可用的软件，使用户更容易、更高效地完成任务。

由于认识到可用性在成功的软件上所起的关键作用，《面向使用的软件设计》为程序设计人员提供了具体的工具和技术来实现这一目标。

《面向使用的软件设计》并不只是给出一些优秀用户界面设计的准则，还为读者提供了一个系统化的软件开发过程。

这个被称为以使用为中心设计方法的过程.将两种开发方法的元素有机组合起来：用例（可以与UML一起使用）和基本建模。

通过大量传统和特殊的软件应用的实例和案例分析，作者描述了真正可行和具有实用价值的实践。

《面向使用的软件设计》要点：提供一个开发高可用性软件的优秀过程。

介绍已在业界成功运用的实用方法和模型。

与统一过程和其他面向对象软件工程方法等现代软件开发实践相辅相成。

<<面向使用的软件设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>