

人机交互知识点整理

第一章 人机交互概述

背景知识

1、人机交互的概念：

定义 1：有关交互式计算机系统设计、评估、实现以及与之相关现象的学科[ACM]

定义 2：研究人、计算机以及他们之间相互作用方式的学科，学习人机交互的目的是使计算机技术更好地为人类服务[Alan Dix]

定义 3：有关可用性的学习和实践，是关于理解和构建用户乐于使用的软件和技术，并能在使用时发现产品有效性的学科[Carroll]

无论使用哪一种术语或定义方式，人机交互学科所关注的首要问题都是人和计算机之间的关系问题

2、人机交互的研究内容

- (1) 界面设计的方法和过程
- (2) 界面实现的方法
- (3) 界面分析和评估技术
- (4) 开发新型界面和交互技术
- (5) 构建交互相关的描述模型和预测模型

3、学习人机交互的原因

- (1) 从市场的角度，用户开始期望系统能够简单易用，同时对那些设计低劣的系统的容忍度越来越差
- (2) 从企业的角度，改善人机交互能够提高员工的生产效率；学习人机交互能够降低产品的后续支持成本
- (3) 从个人的角度，可以帮助用户有效地降低错误发生的概率，避免由于错误引发的时间、金钱以及生命等难以估量的损失

4、人机交互的相关领域

基础学科：心理学、认知科学、人机工程学、社会学、计算机科学、工程学、商务知识、图形设计、科技写作、产品设计、工业设计等

人机交互与人性因素的区别：人机交互更关注使用计算机的用户，而任性因素没有这一限制

人机交互与人机工程学的区别：人机交互对面向重复劳动的人物和过程关注较少，且对用户界面的物理形式和工业设计不够重视

人机交互与交互设计的区别：交互设计不仅仅关注人与计算机之间的关系，同时还包括人与人以及人与其他非计算机系统之间的相互作用。交互设计人员致力于改善人与产品或人与服务之间的关系

发展历史（新的界面变革都包含了上一代界面，有的交互方式仍然尤其存在的必要性）

1、重要的学术事件

- (1) 1945 年, Vannevar Bush, “As we may think” (诚如所思)
- (2) 1959 年, HCI 领域第一篇论文（从减轻操作疲劳的角度讨论计算机控制台设计）
- (3) 1960 年, Liklider JCK 提出 “Human-Computer Symbiosis” (人机共生), HCI 的启蒙观点
- (4) 1969 年, 第一次人机系统国际大会召开, 第一份专业杂志创刊 (HCI 的里程碑)
- (5) 1970 年, 英国拉夫堡(Loughbocough)大学的 HUSAT 研究中心和 Xerox 公司的 Palo Alto 研究中心成立 (PARC)

- (6) 1980s, 出版学术专著, Interface->Interaction
- (7) 1990s, 智能化交互、多通道交互、虚拟现实、脑机交互……

2、主要的发展阶段

(1) 批处理阶段:

特点: 不符合人的习惯; 耗费时间, 又容易出错; 只有少数专业人士才能够运用自如

应用: 文献服务器

(2) 联机终端时代: 真正意义上的人机交互开始于联机终端出现之后, 计算机和用户能够借助一种双方都能理解的语言进行交互式对话 (CLI)

特点: 一维界面; 回车后不能再对命令内容进行修改; 灵活、高效

主要研究问题: 大部分命令语言对用户输入的要求非常严格; 命令名称的缩写在一定程度上减轻了用户的使用负担

(3) 图形用户界面时期 (GUI):

主要特征: 直接操纵

WIMP 模型: 用户可在窗口内选取任意交互位置, 且不同窗口之间能够叠加; 二维半界面不同的交互方式本身在可用性方面并没有根本性的不同, 更重要的是认真对待界面设计的态度。

3、著名人物与事件 (参考 PPT 和教材 P11~12 的表格)

4、人机交互的发展

传统的用户界面包含两种静态的媒体类型: 文本和图形 (图像)

多媒体界面: 引入动画、音视频等动态媒体; 二维半->三维 或更高

多通道交互技术: 综合利用语音、手势、视线等交互方式;

具有并行性, 可同时接收来自多个通道的信息

允许使用模糊的表达手段传递信息 (语日常生活吻合)

避免了生硬、频繁、耗时的通道切换, 并具有交互的双向性

虚拟现实

语音交互

脑机交互

人机交互与软件工程的关系

1、二者相互独立的证据:

- (1) 软件工程师与人机交互设计人员关注的重点有很大不同 (软件工程: 以系统功能为中心; 交互设计: 以用户为中心)
- (2) 交互设计的评估方式也与一般软件工程方法存在不同 (交互评估通常基于真实用户, 评价机制往往来自于用户使用的直接感觉)
- (3) 历史传统使然



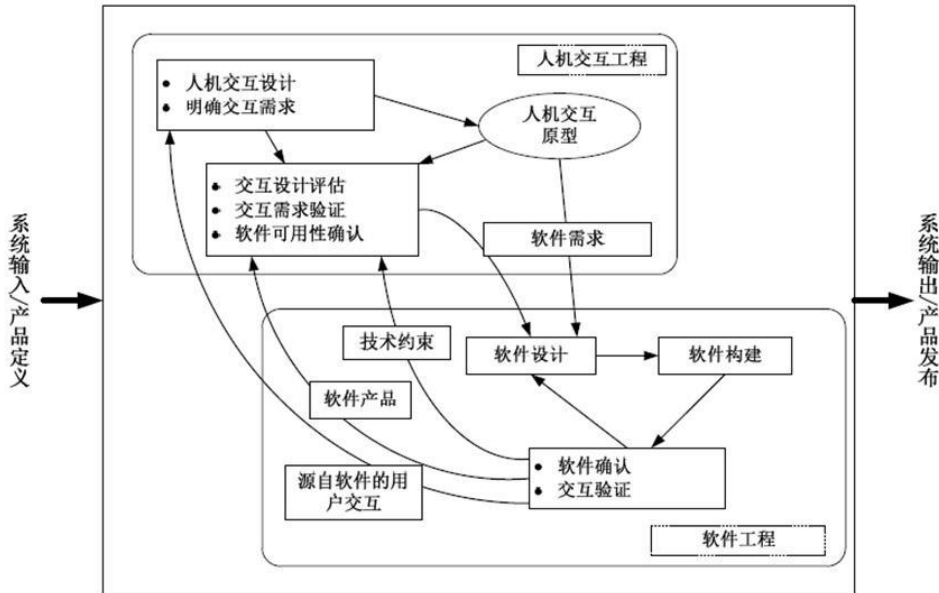
2、二者的关系:

- (1) 相互独立
- (2) 人机交互对软件工程技术的发展有非常大的促进作用

- (3) 将二者结合起来存在很大的困难

3、传统 SE 方法在实现交互式系统方面的缺陷：

- (1) 没有提出明确地对用户界面及可用性需求进行描述的方法
- (2) 不能够在系统开发过程进行中对用户界面进行终端测试
- (3) 具有完善的系统功能（产品的可用性、有效性以及满意度并不高）



4、将二者结合的困难：

- (1) 价值观不同：SE 的实施策略和方法选择上常有一定的倾向性；HCI 包含较多的主观性和灵活性
- (2) 方法论存在差异：SE 采用形式化分析方法；HCI 采用非形式化方法

第二章 人机交互的基础知识

交互框架

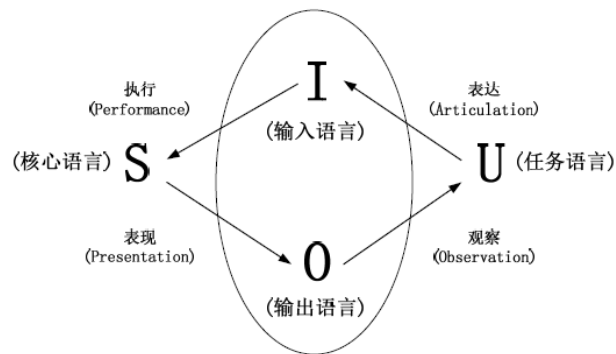
- 1、**框架定义**：框架是提供理解或定义的一种结构，他能够帮助人们结构化设计过程，认识设计过程中的主要问题，还有助于定义问题所涉及的领域
- 2、**执行/评估活动周期 EEC**：
 - (1) **活动的四个基本组成部分**：目标（想做什么）、执行（实现目标必须进行的操作）、客观因素（执行活动时必须考虑的客观条件）、评估（用于衡量活动执行的结果与目标之间的差距）
 - (2) **七个阶段**：建立目标、形成操作意向、明确动作序列、执行动作、感知系统状态、解释系统状态、对照目标评估系统状态
 - (3) Norman 模型是从用户角度探讨人机界面问题的
目标驱动循环：当某个目标确定后循环才开始
数据驱动循环：当某个事件环境确定后开始循环
 - (4) Norman 模型可以用于解释为什么有些界面会给用户使用带来问题
执行隔阂：用户为达到目标而制定的动作与系统允许的动作之间的差别
评估隔阂：系统状态的实际表现与用户预期之间的差别
- 3、**扩展 EEC 框架**
 - (1) 交互式系统的四个组成部分：系统（S）、用户（U）、输入（I）、输出（O）
 - (2) 其中，输入和输出构成了界面
 - (3) 执行阶段包括三个翻译过程：
定义：用户阐述某个目标，然后通过输入语言进行协调和链接

执行：输入语言被转换成内核语言，表示系统要执行的操作

表现：系统使用输出语言吧内核语言的执行结果表示出来

(4) 评估阶段的过程

观察：用户将输出与原有的目标进行比较从而评估交互的结果



交互形式

1、命令行交互：用户通过在屏幕某个位置上键入特定命令的方式来执行任务；基于字符的界面

优点：专家用户能够快速完成任务；较 GUI 节约系统资源；可动态配置可操作选项；键盘操作较鼠标操作更加精确；支持用户自定义命令。

缺点：命令语言的掌握对用户的记忆能力提出较高要求；基于回忆的方式 (recall memory)，没有 GUI 基于识别的方式 (recognition memory) 容易使用；键盘操作，出错频率较高；要求用户记忆指令的表示方式（与可用性理论所强调的“不应要求用户了解计算机底层的实现细节”相违背）。

2、菜单驱动界面：以一组层次化菜单的方式提供用户可用的功能选项，一个或多个选项的选择可以改变界面的状态（通过鼠标、数字键、字母键或者方向键进行选择）

优点：基于识别机制，对记忆的需求较低；具有自解释性；容易纠错；适合新手用户，若提供了较好的快捷键功能，则对于专家用户同样适用。

缺点：导航方式不够灵活；当菜单规模较大时，导航效率不高；占用屏幕空间，不适合小型显示设备，（为节省空间，通常组织为下拉菜单或弹出式菜单）；对专家用户而言使用效率不高。

3、基于表格的界面：显示给用户的是一个表格，里面有一些需要用户填写的空格

优点：简化数据输入；只需识别无需学习；特别适合于日常文书处理等需要键入大量数据的工作。

缺点：占用大量屏幕空间；导致业务流程较形式。

4、直接操纵：

简单介绍：用户通过在可视化对象上面进行某些操作来达到执行任务的目的；展现了真实世界的一种扩展；对象和操作一直可见；迅速且伴有直观的显示结果的增量操作；增量操作可以方便地逆转

三个阶段：自由阶段（指用户执行操作前的屏幕视图）、捕获阶段（在用户动作（点击、点击拖拽等）执行过程中屏幕的显示情况）、终止阶段（用户动作执行后屏幕的显示情况）

优点：将任务概念可视化，用户可以非常方便地辨别他们；容易学习，适合新手用户；基于识别，对记忆的要求不高，可减少错误发生；支持空间线索，鼓励用户对界面进行探索；可实现对用户操作的快速反馈，具有较高的用户主观满意度。

缺点：实现起来比较困难；对专家用户而言效率不高；不适合小屏幕显示设备；对图形显示性能的需求较高；不具备自解释性，可能误导用户。

5、问答界面：通过询问用户一系列问题实现人与计算机的交互（Web 问卷是典型的采用问答方式进行组织的应用，应允许用户方便地取消其中一个界面的选项）

优点：对记忆的要求较低；每个界面具有自解释性；将任务流程以简单的线性表示；适合新手用户。

缺点：要求从用户端获得有效输入；要求用户熟悉界面控制；纠错过程可能比较乏味。

6、隐喻界面：

本质：在用户已有知识的基础上建立一组新的知识，实现界面视觉提示和系统功能之间的知觉联系，进而帮助用户从新手用户转变为专家用户

优点：直观生动；无需学习。

缺点：不具有可扩展性；不同用户对同一事物可能产生不同的联想；紧紧地将我们的理念和物理世界束缚在一起；寻找恰当的隐喻可能存在困难。

7、自然语言交互：

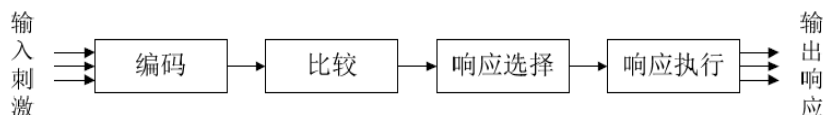
自然语言的模糊性；受限于理解技术，当前只能使用受限的语言与计算机进行交流。

交互形式	应用场合	主要特点
命令行	命令行界面，WIMP	容易编辑和重用历史命令，功能强大的语言可支持非常复杂的操作
菜单	WIMP，基于电话系统的界面	用户不必记忆选项，但有可能付出操作速度减慢或混淆多级菜单的代价
填表	WIMP	一次可以看见和编辑多个区域
直接操纵	WIMP，虚拟现实	在用户控制下。可以运用隐喻，有利于图形技术
问答	命令行界面，WIMP	计算机控制用户，适合偶尔使用的情况
隐喻	WIMP	用户无需学习，但可表达的概念非常有限
自然语言	未来交互系统	理想情况下允许用户自由输入

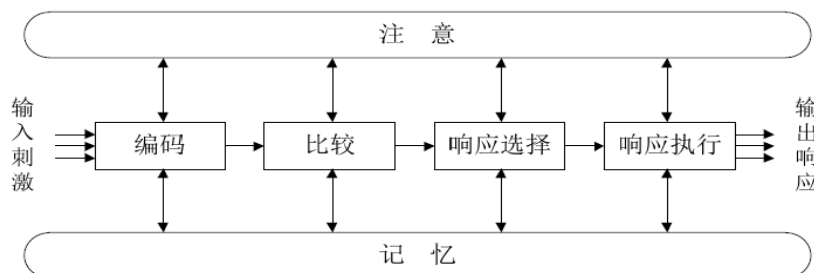
理解用户

1、信息处理模型

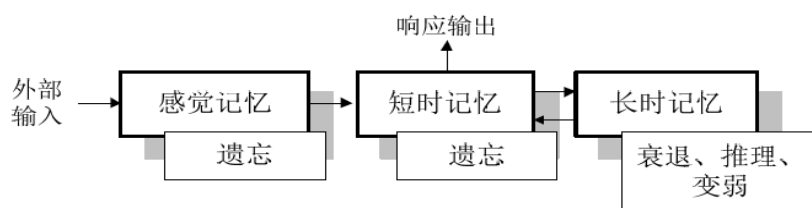
- (1) **信息处理机模型[Lindsay 和 Norman]：**没有考虑到注意和记忆的重要性



- (2) **扩展的信息处理机模型[Barber]：**注意和记忆功能与信息处理过程的各个阶段存在交互

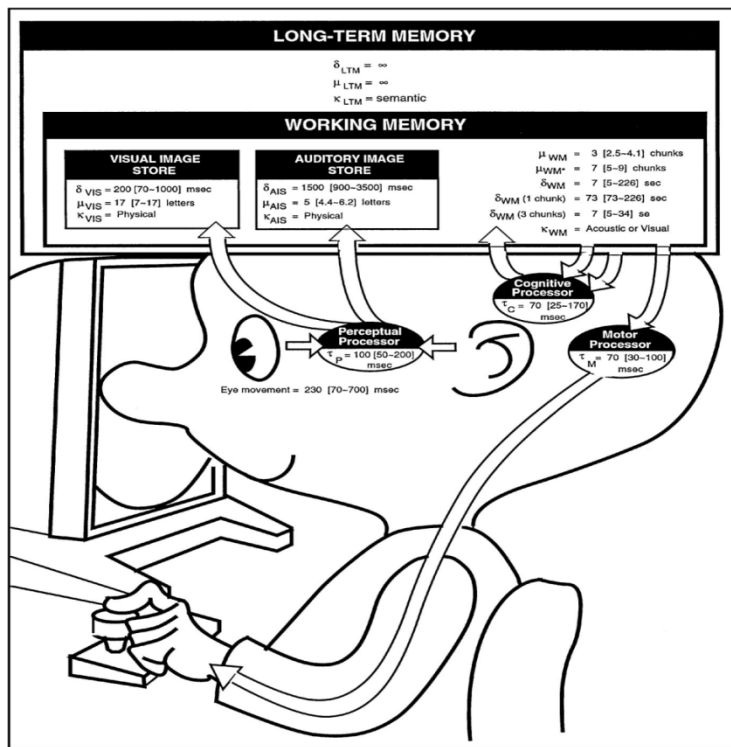


- (3) **记忆模型[Atkinson 和 Shiffrin]：**人脑的记忆分三个阶段：感觉记忆、短时记忆、长时记忆，三个阶段之间可以进行交换



- (4) **人类处理机模型：**

三个交互式组件：感知处理器（期信息将被输出到声音存储盒视觉存储区域）；认知处理器（输入将被输出到工作记忆，同时它能够访问工作记忆和长时记忆中的信息）；动作处理器（用于执行动作）。
存在的问题：把认知过程描述为一系列处理步骤；仅关注单个人和单个任务的执行过程；忽视了复杂操作执行人与人之间及任务与任务之间的互动；忽视了环境和其他人可能带来的影响



(5) 其他模型：外部认知模型、分布式认知模型

2、认知心理学

介绍：兴起于 20 世纪 50 年代中期；关注人的高级心理过程，如记忆、思维、语言、感知和问题解决能力等。

对 HCI 贡献：有助于理解人与计算机的交互过程，同时也可对用户行为进行预测；人对于外界的感知有 80% 来自于视觉获取的信息

格式塔心理学：研究人是如何感知一个良好组织的模式的，而不是将其视为一系列相互独立的部分，格式塔心理学又称完形心理学。表明用户在感知事物的时候总是尽可能将其视为一个“好”的型式

格式塔心理学的主要原则：

- (1) 相近性原则：空间上比较靠近的物体容易被视为整体
- (2) 相似性原则：人们习惯将看上去相似的物体看成一个整体
- (3) 连续性原则：人们会将共线或者具有相同方向的物体组合在一起
- (4) 完整和闭合性原则：感知过程中人们倾向于忽视轮廓的间隙而将其视作一个完整的整体
- (5) 对称性原则：人们习惯将相互对称且能够组合为有意义单元的对象组合在一起
- (6) 前景&背景：当关注一个物体时，物体本身作为前景对象，该物体周围的场景就变成的背景；前景和背景在某些情况下可以互换。

3、人的认知特性

感觉记忆：又称瞬时记忆；在人脑中持续约为 1 秒钟；帮助我们z把相继出现的一组图片组合成一个连续的图像序列，产生动态的影像信息。当引起感觉知觉的刺激物不再继续呈现是，其作用仍能保持一个非常短的时间，这种短暂的保持就是感觉记忆。

短时记忆：感觉记忆经编码后成为短时记忆；又称工作记忆，存储的是当前正在使用的信息，是信息加工系统的核心，约保持 30 秒；储存的是当前正在使用的信息，是信息加工系统的核心，可理解为计算机的内存；短时记忆的存储能力约为 7 ± 2 个信息单元。

长时记忆：短时记忆中的信息经进一步加工后会变为长时记忆；只有与长时记忆区的信息具有某种联系的新信息才能够进入长时记忆；长时记忆的信息容量几乎是无限的

长时记忆中的问题：

- (1) 遗忘：长时记忆中的信息有时是无法提取；不代表长时记忆区的信息丢失了
- (2) 易出错：“人为错误”被定义为“人未发挥自身所具备的功能而产生的失误，它可能降低交互系统的功能”；从表面上看是由于用户的误解、误操作或一时大意；大部分交互问题都源于系统设计本身

长时记忆的启发：注意使用线索来引导用户完成特定任务；在追求独特的创新设计时应注重结合优秀的交互范型

7±2 理论:

- (1) 影响: 菜单中最多只能有 7 个选项; 工具栏上只能显示 7 个图标
- (2) 事实: 浏览菜单和工具栏基于人的识别功能 (人们识别事物的能力要远胜于回忆事物的能力); 界面设计时要尽可能减小对用户的记忆需求, 同时可考虑通过将信息放置于一定的上下文中, 来减少信息单元的数目

错觉: 即知觉感受的扭曲; 有别于幻觉, 是不可避免的 (即告知实际上应该如何, 大家仍然会被错误的判断主导); 启示: 对于物体的视觉感知与物体所处的上下文密切相关

交互设备基础

1、文本输入设备

- (1) **键盘: 一般键盘** (一般一次只能响应一个按键; 新用户的击键速率大约为每秒钟 1 次, 而熟练用户则能达到每秒 15 次的敲击频率; 按键较多的硬盘可营造一种专业化的印象, 新手用户望而生畏; 小键盘特别适合移动设备应用, 功能也可能因此受到限制; QWERTY 键盘时为了降低输入的速率); 和**弦键盘** (能够同时响应多个按键, 可实现快速数据输入; 学习使用时间也较长); **投影键盘** (内置的红色激光发射器可以在任何物体表面投影出标准键盘的轮廓; 红外线技术跟踪手指的动作; 可减小键盘所占的物理空间)
- (2) **手写输入:** 比较自然; 输入速度慢
- (3) **语音输入:** 易受场景噪音影响, 受识别效果影响, 输入效率仅为键盘输入的一半
- (4) **光学字符识别:** 简称 OCR; 让计算机直接 “阅读”; 实现了大批量历史数据的信息化

2、定位设备

- (1) **鼠标:** 最常用的指点和定位设备; 允许在较小的距离控制较大的距离。
- (2) **触控板:** 最广泛的膝上型电脑鼠标; 通过电容感应来获知手指移动情况, 对手指的热量并不敏感; 手指在板上移动的距离与光标在屏幕上移动的距离之间的比率随手指移动的速度而变化
优点: 反应灵敏、移动速度快
缺点: 定位精度较低; 手指出汗时会出现打滑, 不适合在潮湿、多灰的环境
- (3) **指点杆 (小红帽):** 操纵杆的变形; 初学者较难上手; 特点是定位准确 (可通过手指的力度控制鼠标光标移动的速度); 在火车等移动场景下也能够进行准确定位
- (4) **触摸屏:** 通过手指或铁笔中断一个光线的阵列; 较鼠标直观, 定位的速度很快 (适合于在屏幕上选择菜单条目); 定位精度较差, 在小范围选择比较困难; 制造成本很高, 容易污损
- (5) **尖笔/光笔:** 较高的定位精度; 在个人数字助理 (PDA) 中得到了普及; 用户的手需要在设备间不断切换 (交互效率下降); 遮盖部分屏幕显示 (设计界面时, 需要考虑用户手的摆放位置)

3、图像输入设备

- (1) **扫描仪:** 利用光电扫描将图像转换成像素数据
平板式扫描仪: 使用最广泛; 较好的扫描速度、精度和图像质量, 使用简单
手持式扫描仪: 扫描得到的图像是一个个长条; 应用于 CAD 及大幅面工程图纸
滚筒式扫描仪: 最精密的扫描仪器 (图像以 C、M、Y、K 或 R、G、B 的形式记录, 又被称为 “电子分色机”); 价格昂贵, 用于专业印刷排版领域
- (2) **数码相机:** 不需要胶卷和暗房; CCD (电荷耦合元件, Charge Coupled Device) 芯片 (色彩饱和度好, 图像较为锐利, 质感逼真); CMOS (互补金属氧化物导体, Complementary Metal-Oxide Semiconductor) 芯片 (高感光度下的表现要好于 CCD, 读取速度快, 成本低廉; 特别适合高性能的单反相机)
- (3) **传真机:** 兼具图像输入和输出功能; 热敏纸传真机 (又称卷筒纸传真机)、热转印式普通纸传真机、激光式普通纸传真机 (又称激光一体机) 和喷墨式普通纸传真机

4、显示设备

- (1) **光栅扫描阴极射线管 (简称 CRT)**
优点: 图像色彩丰富; 高清晰度、低成本; 丰富的几何失真调整能力
缺点: 辐射; 高电压具有潜在的爆炸危险; 占据较大空间
- (2) **液晶显示器**

原理：电压的变化会影响微小液晶泡的偏振，从而阻碍部分光线射出，进而使得屏幕显示出现明暗差异

优点：低辐射、低耗能；体积小质量轻；可视面积大，不存在几何和线性失真

缺点：价格昂贵；侧面显示会失真；响应时间慢；使用寿命有限

(3) 等离子体显示器

原理：通过细小封闭的玻璃泡将水平方向与垂直方向隔离，当水平或垂直方向中任何一个方向输入高电压时，玻璃泡中的惰性气体就会发光

优点：可视角度较液晶显示器大；更好的颜色质量和对比度

缺点：使用寿命也较短；易产生残留影像

(4) 发光二极管

原理：正向加压时，发光二极管能发出单色且不连续的光

优点：耗电量小；反应时间快；使用寿命长；体积小

缺点：价格昂贵；使用寿命与散热性能的好坏密切相关

(5) 电子墨水

原理：电子墨水液体中悬浮着几百万个细小的微胶囊，每个胶囊内部都是染料和颜色芯片的混合物。通电时，颜色芯片发生作用而改变颜色，并且可以显示变化的图像

优点：易读性好、低功耗、成本低、质量小；可以应用到纸上或布上，产生和报纸一样的显示效果；具有较好的柔性，可折叠

(6) 点字显示器

为盲人用户阅读文字提供了新的途径（字显示格+功能键；可更新的图像显示器正在开发中）

5、虚拟环境下的交互设备

原理：计算机模拟产生三维空间的虚拟世界；提供用户视觉、听觉、触觉等感知器官的模拟体验；让用户如同身临其境一般，实时地观察和操作三维空间内的事物

(1) **三维鼠标：**6个自由度可以选择（倾斜度、偏航角、翻滚度）

(2) **数字手套：**增强了使用的互动性和沉浸感

(3) **虚拟现实头盔：**可模拟实现视觉、听觉、视觉、触觉甚至味觉功能

第三章 交互设计目标与原则

背景

1、以往：热爱技术的专业人员；现在：缺乏耐心的消费者

2、各种术语区分：

(1) **“用户友好”：**用户希望在完成任务的时候，机器不要碍手碍脚；不同用户的需求各异，不能从系统单方面友好

(2) **“可用”：**使产品易学易用等

(3) **“用户体验”：**使用户喜欢产品

交互设计目标

1、**可用性：**可用性目标不仅涉及人与之正在发生交互作用的系统，还包括系统对使用它的人所产生的作用，是交互式系统质量的一种重要度量

2、可用性的五个方面：

(1) **易学性：**指使用系统的难易，即系统应当容易学习，从而用户可以在较短时间内应用系统来完成某些任务；最基本的可用性属性

评价系统是否容易学习的标准：“十分钟法则”

(2) **易记性：**用户在学会使用软件后应当容易记忆；学会某个系统后，应能够迅速回想起它的使用方法
影响因素：位置（将特定对象放在固定位置）；分组（对事物按照逻辑进行恰当的分组）；惯例（尽可能使用通用的对象或符号）；冗余（使用多个感知通道对信息进行编码）

启发：良好组织，使用用户已有的经验帮助提高易记性

(3) **有效率**：当用户学会使用产品之后，用户应该具有更高的生产力水平（效率）；效率指熟练用户到达学习曲线上平坦阶段时的稳定绩效水平

(4) **低出错率**：人是会犯错误的（有些错误会被用户发现并纠正，有些错误会带来灾难性后果）
措施：保证导致灾难性后果错误的发生频率降到最低；保证错误发生后迅速恢复到正常状态

(5) **主观满意度**：用户对系统的主观喜爱程度，有时候与前面几个方面是相矛盾的

观念的转变：

传统软件质量观：侧重内部效率和可靠性，如程序代码运行时的效率以及灵活性、可维护性

人机交互软件质量观：转向用户视角

3、用户体验目标：

(1) 问题：

随着新技术渗透到人们的日常生活中，人们对产品有了更多的要求；

让用户感到满意并留下愉快主观感受的产品更可能被多次使用才是用户愿意使用和购买的。

(2) 用户体验的概念：用户在与系统交互时的感觉

较可用性目标主观，有时却需要牺牲可用性来达到好的用户体验

可用性可能对用户体验带来阻碍

简易可用性工程

1、**特点**：以提高产品的可用性为目标的先进的产品开发方法论；借鉴了许多不同领域的方法和技术；强调以人
为中心来进行交互式产品的设计研发

2、**历史**：上世纪 80 年代获得工业应用；90 年代得到迅速普及

3、**实例**：IBM 公司的“可用性方面的投入是一本万利的”；MS 已有 14 个可用性实验室近 200 名可用性专业员工

4、可用性度量方法：

(1) **常用方法**：选择一些能够代表目标用户群体的测试用户；让这些用户使用系统执行一组预定的任务；
比较任务的执行情况；针对多维属性（取每个可用性属性的平均值；查看整体分布情况）

(2) **主观满意度**：在 1~5 分的 5 分制情况下平均值至少为 4；或至少 50%的用户给系统打 5 分（或给系统打 1 分的用户不超过 5%）

5、注意事项：

(1) **度量一定要针对特定的用户和特定的任务进行**：用户对不同任务的可用性结果预期可能不同；

(2) **测试前要明确一组具有代表性的测试任务**：任务描述应使用用户的任务语言，清晰明确

6、可用性各个方面的度量：

(1) 易学性度量：可用性属性中最容易度量的属性

步骤：1、找一些从未使用过系统的用户（能够代表系统的目标用户；区分没有任何计算机使用经验的新手用户和具有一般计算机使用经验的用户）2、统计他们学习使用系统直至达到某种熟练程度的时间

特定熟练程度：用户能够完成某个特定的任务；或用户能够在特定的时间内完成一组特定任务（原因：学习曲线没有明确区分“学会和未学会”）

(2) 使用效率度量：

并不是所有用户都能够迅速达到最终的绩效水平（用户自身的原因；少量系统的操作十分复杂）

同样要区分不同的用户群体（这项度量针对有经验的用户）：“有经验”较为正规的衡量方式是通过使用系统的小时数来定义的；先使用，然后度量其绩效水平；或为用户绘制学习曲线（当发现用户的绩效水平在一段时间内不再提高时，就认为已经达到了该用户的稳定绩效水平）

(3) 易记性度量：

用户分类：新手用户，熟练用户，非频繁使用用户

针对用户群：对非频繁使用用户进行测试最能体现系统的易记性

度量方法：1、对在特定长时间内没有使用系统的用户进行标准用户测试（记录下这些用户执行特定任务所用的时间）；2、对用户进行记忆测试（如在用户完成一个应用系统的特定任务后，让用户解

释各种命令的作用)

(4) **错误率度量:**

错误: 通常指不能实现预定目标的操作

度量: 1、在用户执行特定任务时通过统计这种操作的次数; 2、可以在度量其他可用性属性的同时来度量

错误分类: 1、错误发生后能够被用户立刻纠正, 不会对系统带来灾难性的影响 (往往会被包含在使用效率的统计当中); 2、不易于被用户发现, 从而可能造成最终结果存在问题 (设计人员在设计时也应该将其发生的频率降到最低)

(5) **满意度度量:**

满意度度量评价都是主观的 (以询问用户的方式进行度量更合适; 为减少单个用户评价的主观性, 把多个用户的结果综合起来取其平均值)

度量通常在用户测试完成后进行 (要求用户通过简单的调查问卷对系统打分: 1、可以 1-5 或 1-7 的 Likert 度量尺度或语义差异尺度作为打分标准; 2、一定要在用户使用系统执行真实的任务之后再询问他们的看法)

7、**四种主要技术:**

完整的可用性工程过程: 了解用户、竞争性分析、设定可用性目标、用户参与的设计、迭代设计、产品发布后的工作

简化的过程: 用户和任务观察、场景 (scenario)、简化的边做边说 (thinking aloud)、启发式评估

(1) **用户和任务观察:** 了解产品的目标用户是可用性工程的第一个步骤

注意: 要直接与潜在用户进行接触; 不要满足于间接的接触和道听途说; “你” 不是用户

(2) **场景:** 简便易行的原型工具; 通过省略整个系统的若干部分来减少实现的复杂性; 可以是纸质模型, 也可以是简单的 RAD 原型

水平原型: 减少功能的深度并获得界面的表层

垂直原型: 减少功能的数量而对所选功能进行完整实现

(3) **简化的边做边说**

让真实用户在使用系统执行一组特定任务的时候, 讲出他们的所思所想;

最有价值的单个可用性工程方法;

可了解用户为什么这样做, 并确定其可能对系统产生的误解;

实验人员需要不断地提示用户, 或请他们事先观摩

(4) **启发式评估**

研究表明, 能够发现许多可用性问题 (剩下的可以通过简化的边做边说方法来发现);

为避免个人的偏见, 应当让多个不同的人来进行经验性评估

n 个测试专家能够发现的可用性问题数量: $N(1-(1-L)^n)$

N: 设计中存在的可用性问题的总数

L: 单个参与者所能够发现的可用性问题的比例 (经验取值约为 31%)

5 名专家能够发现约 80% 的可用性问题; 建议将测试分阶段进行

调查问卷设计

1、**设计原则:**

(1) **通常设计得较为简短:** 以保证最高的结果返回率

(2) **以 1-5 或 1-7 的 Likert 度量尺度或语义差异尺度作为打分标准:**

得分越高, 说明认可的程度越高;

研究发现 1-5 分的评价尺度的中值是 3.6 分 (1 分满意度最低, 5 分最高)

评估定义一个锚点或基准点是非常重要的

(3) **不论采用什么样的评价尺度, 都应当在大规模测试前进行小规模试点测试**

图标的可用性度量 (参考 PPT 22~23)

交互设计原则

1、概述：

这些规则大多来源于提出者的经验和总结

不是完美无缺的，甚至有些会相互矛盾

在具体使用时，必须根据实际情况进行调整和细化

2、基本规则[Alan Dix]：（参考教材 P53~54，3 个表格）

- (1) **可学习性**：新用户能用它开始有效的交互并能获得最大的性能
可预见性、同步性、熟悉性、普遍性、一致性
- (2) **灵活性**：用户和系统能以多种方式交换信息
能动性、多线程、任务可移植性、可替换性、可定制性
- (3) **健壮性**：在决定成就和目标评估方面对用户提供的支持程度
可观察性、可恢复性、反应性、任务一致性

3、黄金规则[Shneiderman]：

- (1) 尽可能保证一致
- (2) 符合普遍可用性
- (3) 提供信息丰富的反馈
- (4) 设计说明对话框以生成结束信息
- (5) 预防并处理错误
- (6) 让操作容易撤销
- (7) 支持内部控制点
- (8) 减轻短时记忆负担

4、七项原理[Norman]：

- (1) 应用现实世界和头脑中的知识
- (2) 简化任务结构
- (3) 使事情变得明显
- (4) 获得正确的映射
- (5) 利用自然和人为的限制力量
- (6) 容错设计
- (7) 当所有都不成功时进行标准化

5、十项启发式规则[Nielsen]：

- (1) 系统状态的可见度（如，进度条）
- (2) 系统和现实世界的吻合
- (3) 用户享有控制权和自主权
- (4) 一致性和标准化
- (5) 避免出错
- (6) 依赖识别而非记忆
- (7) 使用的灵活性和高效性
- (8) 审美感和最小化设计
- (9) 帮助用户识别、诊断和恢复错误
- (10) 帮助和文档

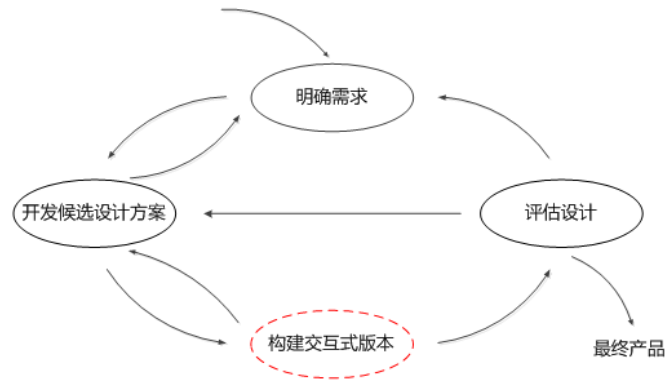
第四章 交互设计过程

交互式设计过程

1、设计过程的基本活动：不是所有的交互式产品都必须采取这个开发过程；代表了交互设计领域的实践经验

- (1) **标识用户需要并建立需求**：必须了解谁是目标用户；交互式产品应提供哪些支持；最基本的活动
- (2) **开发满足需求的候选设计方案**：设计的核心活动；概念设计和物理设计
可以划分为 2 个子活动：概念设计（制作产品的概念模型）、物理设计（产品的细节）

- (3) **构建设计的交互式版本**：评价设计的最佳方法就是让用户与产品交互；不一定是可运行的软件版本
- (4) **评估设计**：评估它的可用性和可接受性；制定各种评估标准



2、关键特征

- (1) **以用户为中心**：人机交互领域的一个核心观点
- (2) **稳定的可用性标准**：有助于设计人员选择不同的候选方案；并在产品开发过程中随时检查
- (3) **迭代**：设计人员不可能一次就找出正确的解决方案；利用反馈来改进设计

设计过程中的问题

1、如何选取用户

- (1) **用户的分类**：
 - 主要用户**：经常使用系统的用户
 - 二级用户**：偶尔使用系统，或者通过中间人使用系统的用户
 - 三级用户**：引入系统会影响到的人员以及影响系统购买的人员
- (2) **涉众**：被系统影响的、而且对系统需求有直接或间接影响的个人或机构
- (3) **不建议让所有的当事人都参与开发**：应该了解产品会造成多大范围的冲击；明确让什么人在何种程度上参与开发

2、如何明确需求

- (1) 必须理解用户的特征和能力，了解他们想要达到什么目标，以及可满足的支持；而不是简单地问：“你需要什么”
- (2) **原因**：用户的能力、特征将影响产品的设计；新产品的用户和有代表性的任务较难确定
- (3) **现状**：开发人员往往倾向于创建自己想要的产品或者是过去开发过的类似产品（当前或过去的行为是对未来行为的很好启示）

3、如何提出候选设计方案：有时候，好的设计来自于个别人的才干、创造力；但更多时候候选方案来自考虑其他相似的设计

候选方案的来源：1、竞争对手的产品；2、类似系统的早期版本；3、完全不同的东西；4、“所谓‘专家’，指的是那些能够从先前的经验中找到正确灵感并应用于当前工作的人”

注意：不应局限于已有系统的限制

4、最终方案的选择：从候选方案中作出选择成为“设计决策”

- (1) **决策的分类**：1、关于外部特征的（可见、可测量的）；2、关于内部特征的（除非是剖析系统，否则他们是不可见、不可测量的）
- (2) 在交互设计中，用户与产品的交互方式是设计的推动力，所以我们要把注意力集中在可见、可测量的外部行为上。
- (3) **以用户为中心**：让用户和当事人与各种方案相交互（听取他们的体验、偏好和改进建议等）；制定明确、无歧义的质量基准（不明确：系统应能够快速响应；明确：系统响应时间不应超过1秒）

交互设计生命周期

1、生命周期的概念：生命周期模型都是现实的简化表示，是现实的一种抽象

2、传统软件生命周期模型

- (1) **瀑布模型**：线性顺序的软件开发方法，包括需求分析、设计、编码、实现、测试和维护

贡献：1、把分析和编码作为了模型的两个主要组成部分；2、第一个得到广泛承认的模型

缺点：1、以文档为中心，用户难以理解；2、不适用于交互式软件产品的开发；3、假设需求是不变的

- (2) **螺旋模型：**其基本思想为使用原型及其他方法尽量降低风险，引入迭代思想是为了找出风险和控制风险

特点：1、以降低风险为中心；2、引入了“迭代”的思想

缺点：1、过于复杂，用户难以掌握；2、风险评估需大量信息

- (3) **快速应用开发：**采用以用户为中心的方法，目的是要把由于需求在开发过程中不断变化而导致的风险降至最低

两个关键特征：1、周期时限大致为6个月；2、用户和开发人员共同参与专题讨论，确定需求

动态系统开发方法 (DSDM)：可行性研究、商务研究、开发功能模型（迭代）、设计及构建（迭代）、实现

- (4) **原型法：**允许设计者从一个屏幕到另一个屏幕来讨论诸如外观和感觉、范围、信息流等设计问题；可以认为是对用户意见的一种尊重

原型的分类：1、抛弃原型模型（客户不清楚项目范围和客户不能精确描述项目需求时）；2、进化原型模型（随着项目的进展不断进化，直到实现最终产品）

优点：易于用户提供反馈；减少了开发时间和成本；用户参与到开发过程中

- (5) **总结：**

适应于原则性强的场合：如果从开始就知道想要生产什么，则为了达到目标，可以构建我们自己的设计方法

一个交互式软件系统的所有需求在开始时是无法确定的

如何使得系统更有用处？

建造系统

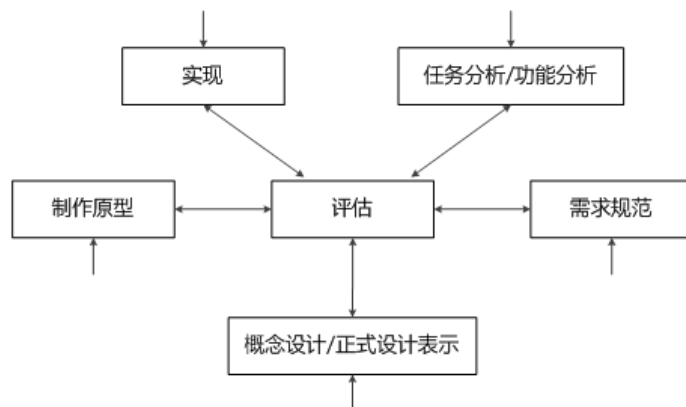
观察和评估与用户的交互

3、交互设计生命周期模型

- (1) **星型生命周期模型**

归纳的两种不同的活动模式：分析模式（自顶向下、组织化、判定和正式化）；合成模式（自顶向生、自由思考、创造性）

特点：1、没有指定任何活动次序；2、“评估”是这个模型的核心；3、源于开发人员的实际经验

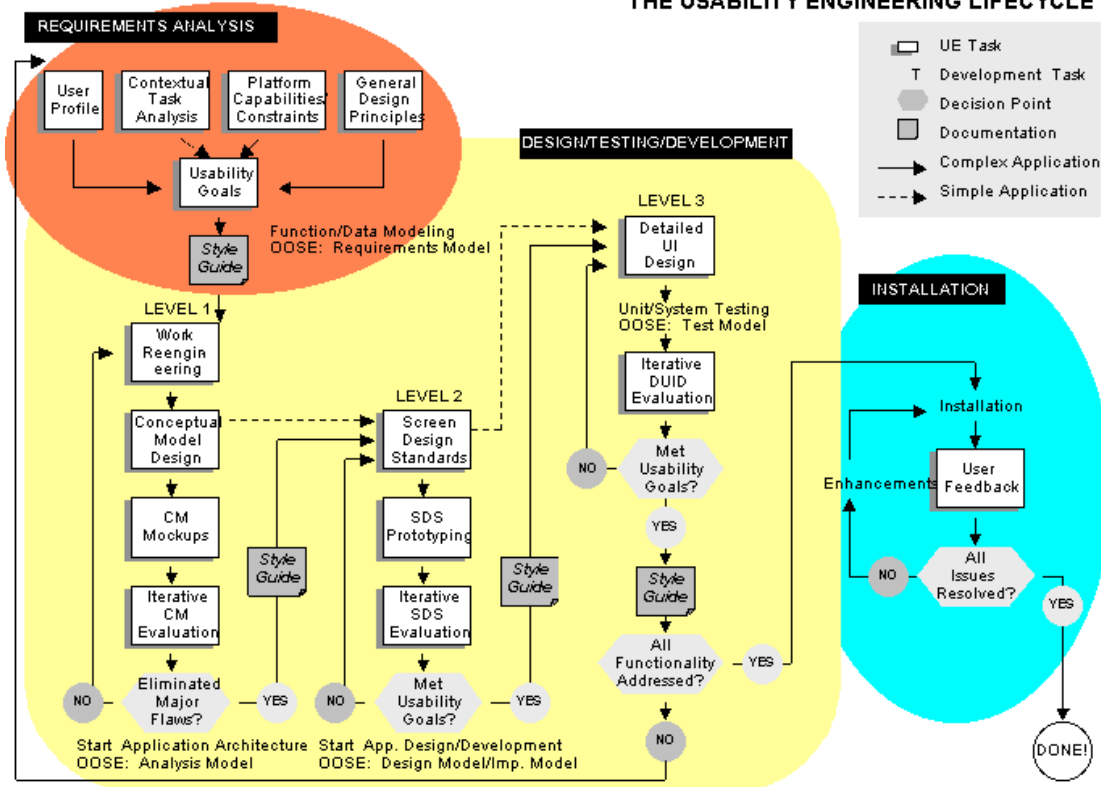


- (2) **可用性工程生命周期**

三个基本任务：需求分析、设计/测试/开发、安装（中间部分最复杂，设计最多子任务）

特点：1、体现了可用性工程的总体概念；2、详细描述了如何执行可用性任务；3、说明了如何把可用性任务集成到传统的软件开发生命周期中

THE USABILITY ENGINEERING LIFECYCLE



(3) 总结:

交互设计生命周期更为灵活

实践中可结合具体需要裁减恰当的生命周期模型（星型模型过于简单和随意；可用性工程生命周期略显呆板）

注意：两个模型都强调了评估的重要性

交互设计过程管理

1、现有问题：开发者和他们的商业委托人或者开发者和他们的客户之间缺乏沟通

2、传统软件工程方法：不足以提供清晰的进程来研究用户

3、以用户为中心的设计：

- (1) 在开发过程中产生尽可能少的错误
- (2) 开发费用少和维护成本低
- (3) 易于学习、执行速度更快
- (4) 鼓励用户探索
- (5) 应尽早使用

4、合理的以用户为中心的交互设计，LUCID

- (1) 预想：发展清晰、共享的产品场景，使概念草案具体化
- (2) 发现：研究用户以决定高端的用户需求、术语和智力模型
- (3) 设计基础：发展概念设计，创造关键屏幕画面原型
- (4) 设计细节：将高端设计加以充实，形成完备的详细说明书
- (5) 构建：通过回顾和后期改变管理方式来支持生产过程
- (6) 发布：通过有力的推广来支持用户向新产品的过渡；进入最后的可用性测试

5、特色：

- (1) 将注意力集中在关键屏幕画面原型：促进用户的早期参与，并为项目的进展创造动力
- (2) 快速原型法是安排进度和预算的关键：依赖于用户界面开发工具

6、目标：通过加强对开发、交付和评审等方面的重视，强调了可用性工程在软件开发过程中的地位，促进有序的开发过程

第五章 交互式系统的需求

1、交互式需求

- (1) **需求的定义：**关于目标产品的一种陈述，它指定了产品应做什么，或者应如何工作。需求应该是具体、明确和无歧义的。
- (2) **设计活动的两个目标：**1、“标识需要”（尽可能理解用户、用户的工作、工作的上下文）；2、“需求活动”（从用户需要中，提炼出一组稳定的需求，作为后续设计的坚实基础）
- (3) **需求活动的组成部分：**搜集数据、解释数据、需求提取
- (4) **数据搜集：**需求活动和评估活动的重要组成部分；数据搜集的技术并不多，但它们非常灵活，可以以多种方式组合、扩充。实际的数据搜集方式是多样化的，包括问卷调查、访谈、专门小组、专题讨论、自然观察和研究文档。

2、产品特征

- (1) **功能不同：**不同类型的产品往往在需求方面存在很大差异，不同的产品对功能的需求不同
- (2) **物理条件不同：**大小、颜色、重量、有效性等
- (3) **使用环境不同：**
 - 物理环境：**如操作环境中的采光、噪音和尘土状况，操作人员是否需要穿防护衣、戴手套、安全帽
 - 社会环境：**如是否要共享数据，同步还是异步
 - 组织环境：**用户支持的质量、响应速度如何，是否提供培训资源或设施
 - 技术环境：**产品应能运行于何种平台上，应与何种技术兼容

3、用户特性

心理学原理部分：假设每个人都有相似的能力和局限性（合理的，心理学原理可以适用于大多数人）。

交互产品设计人员应该意识到个性的差异：用户并不是完全相同的；在设计中尽可能地体现这些差异。

(1) 体验水平差异

背景：程序员只创造适合专家的界面；市场人员要求只适合新手的交互；数目最多、最稳定和最重要的用户群是中间用户；中间用户往往被忽略。

设计目标：让新手快速和无痛苦地成为中间用户；避免为想成为专家的用户设置障碍；让中间用户感到愉快（因为他们的技能将稳定地处于中间层）。

新手用户特点：敏感，且很容易在开始有挫折感。

新手用户设计要求：1、不能将新手状态视为目标；2、让学习过程快速且富有针对性；3、确保程序充分反映了用户关于任务的心智模型；4、无论什么样的帮助，都不应该在界面中固定；5、具有向导功能的对话框帮助较好（不要使用在线帮助作为学习指导）；6、菜单项应该是解释性的。

专家用户特点：对缺少经验的用户有着异乎寻常的影响；欣赏更新的且更强大功能；不会受到复杂性增加的干扰。

专家用户设计要求：对经常使用的工具集，要能快速访问。

中间用户特点：需要工具；知道如何使用参考资料；能够区分经常使用和很少使用的功能；高级功能的存在让永久的中间用户放心。

中间用户设计要求：工具提示（Tooltip）是适合中间用户最好的习惯用法；在线帮助是永久中间用户的极佳工具；常用功能中的工具放在用户界面的前端和中心位置；提供一些额外的高级特性。

(2) 年龄差异

老年人：残疾人的比例随着年龄的增加而增加（65岁以上的老年人中多数有某种残疾）；技术应能提供对残缺部位的支持（如听觉、言语和灵活性）；设计必须清楚、简单并且容许出错；利用冗余来支持信息访问。

儿童：在为儿童设计交互式系统时让他们参加很重要；允许多种输入模式（包括触觉或手写）的界面对于孩子们更适用；通过文本、图形和声音呈现信息的冗余显示也将增强他们的体验。

(3) 文化差异

在不同的文化中符号有不同的意思（不能假设每个人都以同样的方式解释符号）

姿势的理解存在差别

颜色的使用（通过冗余阐明特定颜色的指定意义）

(4) 健康差异

视觉损伤：GUI 应用的增加降低了视觉损伤用户应用的可能性；辅以声音的应用和触觉的应用

听觉损伤：较视觉残疾对与图形界面交互的影响要小（界面中多媒体的增加和声音的应用带来了交互困难）；给听觉内容加文字描述；姿势识别可作为信息输出方式

身体损伤：在控制和应用手的移动方面存在差别；语音输入和输出对那些没有言语障碍的人是一种选择；用姿势和眼球移动的跟踪进行控制

语音损伤：提供合成语音和基于文本的通信（语音合成必须快速地反映自然会话的步调）

诵读困难：提供拼写更正功能；一致性的导航结构和清晰的标识提示

4、用户建模

(1) 人物角色

定义：不是真实的人；是基于观察到的那些真实人的行为和动机，并且在整个设计过程中代表真实的人；是在人口统计学调查收集到的实际用户的行为数据的基础上形成的综合原型；概念简单，但使用起来相当复杂（拼凑出几个用户档案是不行的）

作用：解决产品开发过程中出现的 3 个设计问题

弹性用户（产品团队的每一个人都对用户机器需要有不同理解）：为弹性用户设计赋予了开发者根据自己的意愿编码，而仍然能够为“用户”服务的许可。

资参考设计：设计者或者程序员将其自己的目标、动机、技巧及心智模型投射到产品的设计中

边缘情况设计：必须考虑边缘情况，但它们又不应该成为设计的关注点

(2) 人物角色的构造

人物角色特点：与某个系统有关的用户假定的一组公共需要、兴趣、期望、行为模式和责任；这些属性可能是若干用户共有；同一个用户也可以扮演系统的任意个不同角色。

基于以下问题：

谁将使用系统？

这些用户属于哪些类型的人群？

是什么因素决定他们将怎样使用系统？

他们与软件的关系有什么特征？

他们通常需要软件提供什么支持？

他们对软件会有怎样的行为？他们对软件的行为有什么期望？

注意事项：1、要注意那些与软件用户界面设计有关的角色特征；2、要关注使角色之间彼此相区别的特征；3、要留心焦点角色（最常见、最典型的角色）

建模过程：

1、**拼凑：**采用头脑风暴方法，产生一些零碎概念或模型的片段，先不去考虑他们的细节

2、**组织：**将这些片段按照所构造模型的需要进行分组和分类，归并或删除那些冗余重叠的东西

3、**细节：**建立和完善相应描述，补充遗漏的数据

4、**求精：**对模型进行推敲，以便改进和完善

5、需求获取、分析和验证

(1) 需求的获取

需求获取的两个方法：观察和场景

观察：

①**观察的原因：**设计的最初，可能不知道问什么问题或由谁来回答这些问题

没有与参与者实际的讨论和观察是不可能得到完整的工作流程画面的

有用的信息可以通过观察人们在工作环境中完成他们的活动来获得

②观察的类型：

直接观察：陪同他们工作而直接获得信息；可能影响被观察者的日常活动

间接观察：用视频/录音获得信息；观察者更舒适

不论采用哪种观察方法，重要的是使潜在的用户接受这个过程，以便能在观察人员面前尽量表现得自然，而不受观察者的影响，以免测试出现偏差

场景：

①**场景的定义：**场景是表示任务和工作结构的“非正式的叙述性描述”。

以叙述的方式描述人的行为或任务，从中可以发掘出任务的上下文环境、用户的需要、需求形式可以类似于一篇故事、一个小品或者在给定环境下按照时间顺序的一段情节

②**使用场景的原因：**“讲故事”是人们解释自己做什么或者希望执行某个任务的最自然方式

故事的焦点就是用户希望达到的目标

若场景说明不断提到某个特定形式、行为或者地点，就表明它是这个活动的核心内容

③**场景的来源：**场景说明通常来自专题讨论或者访谈，目的是解释或讨论有关用户目标的一些问题

(2) 应用人物角色和场景剧本的需求定义（人物角色+场景剧本→需求）

需求定义的 5 个步骤：创建问题和前景综述、头脑风暴、确定人物角色的期望、构建情境场景剧本、确立需求（这几个步骤之间是迭代的关系，直到需求变得稳定）

①**创建问题和前景综述：**设计问题综述应该简明地反映需要改变的情况，来服务人物角色和提供产品给人物角色的商业组织；前景综述高层次的设计视图和需求是问题综述的倒置

②头脑风暴：

目的：“说反话”；尽可能地去掉成见，允许设计师以开放和灵活的方式想象来构建场景剧本，使用他们的思维从场景剧本中得到需求；将头脑置于“解决问题模式”中

特点：不受约束且不加以评判；不要花费太多时间，当想法重复或变慢时停止

③确定人物角色的期望：

原因：一个人的心理模型通常是根深蒂固的；界面表现模型与用户心理模型尽量匹配是非常重要的
对于每一个基本和次要人物角色，需确定：

A、影响人物角色愿望的态度、经历、渴望，以及其他社会、文化、环境和认知因素

B、人物角色在使用产品体验方面可能有的一般期待和愿望

C、人物角色对产品行为的期待和愿望

D、人物角色认为什么是数据的基本单元或者元素

理清如下问题：

主体首先提到的是什么？

他们使用哪些动作单词？

他们没有提及对象中的哪些中间步骤、任务或者对象？

④构建情境场景剧本：

特点：

关注人物角色的活动，及其心理模型和动机；将注意力集中在设计的产品中怎样能够最好地帮助你的人物角色达到目标；应该专注于高层次的从用户角度描述的行动，广而浅（不应描述产品或交互的细节）

解决的问题：

产品是否会被使用很长一段时间？

人物角色是否经常被打断？

和其一起使用的其他产品是什么？

人物角色需要做哪些基本的行动来实现目标？

使用产品预期的结果是什么？

注意事项：没有涉及太具体的界面和技术信息；活动是和人物角色的目标相关的（尽可能去掉多余的任务）

⑤**确立需求：**包括数据需求、功能需求、其他需求

数据需求：必须在系统中被描绘的对象和信息；可以被看作是与对象相关的宾语或形容词。如账号、人、文档、邮件、歌曲、图片以及它们的属性比如状态、日期等

功能需求：系统对象必须进行的操作；最终会转化为界面控件

其他需求：业务需求（开发时间表、规则、价格结构、商业模式）；品牌和体验需求（希望用户或顾客将产品、公司或组织联系起来的体验的特征）；技术需求（重量、大小、形式要素、显示、容量限制、软件平台选择）；顾客和合作伙伴需求（易于使用、维护和配置、承受成本和许可协议）

(3) **任务分析：**记录人们如何完成任务的一种方式

作用：可以用来了解通过观察和访谈目前参与工作流程的人收集到的数据；主要用于调查现有情形，而不是展望新系统或设备；分析基本原理，了解人们想要达到什么目标，如何达到这些目标，并由此建立需求

层次化任务分析 (HTA)：应用最广的任务分析技术

①**步骤描述：**把任务分解为若干子任务，再把子任务进一步分解为更细致的子任务。之后，把他们组织成一个“执行次序”，说明在实际情形下如何执行各项任务

②**终止规则（任务分析是一个迭代过程）**

终止点：任务包含了复杂机械响应的地方（如鼠标移动，此时分解没有价值）

涉及内部决断的地方（若决断和查找文档等外部动作相关，则分解；若为纯粹认知性，则终止）

③**方框—线条表示**

④**用途：**手册和教学、需求获取和系统设计、详细的接口设计

(4) **需求验证**

原型的重要性：用户往往不能准确描述自己的需要；用户在看到或尝试某些事物后，就能立即知道自己不需要什么

原型的定义：在某一方面和真正产品比较接近、以便人们能对这一方面的各种技术方案进行不断评估和改进的一种接近于实际产品的模型；借助于原型，当事人就能与未来的产品交互，从中获得一些实际的使用体验，并发掘新思路（实际上，原型可以是任何东西；是设计的一种受限表示）

原型的类别：低保真原型、高保真原型

低保真原型（多数项目的首选）：与最终产品不太相似的原型；使用与最终产品不同的材料，如纸张、纸板；优点是简单、便宜、易于制作和修改

高保真原型：与最终产品更为接近，使用相同的材料；有风险（用户会认为原型就是系统；开发人员可能认为已找到了一个用户满意的设计）

第六章 交互式系统设计

1、设计框架

原因：过早地把重点放在小细节、小部件和精细的交互上会妨碍产品的设计；先站在一个高层次上关注用户界面和相关行为的整体结构

设计框架：定义高层次上的屏幕布局；定义产品的工作流、行为和组织

设计框架的步骤：

(1) **定义外形因素、姿态和输入方法**

输入方式：产品与用户互动的形式；取决于产品的外形和人物角色的能力和喜好；哪种方式或者组合更适合设定的人物角色。（包括键盘、鼠标、触摸屏、声音等）

(2) **定义功能和数据元素**

数据元素：交互产品中的基本主体，如相片、电子邮件、订单

功能元素：对数据元素操作的工具以及输入或者放置数据元素的位置

(3) **决定功能组合层次**

原因：更好地在任务中和任务间来帮助促进任务角色的操作流程

需考虑的元素：哪些元素需要大片的视频区域；容器如何组织才能优化工作流；哪些元素是被一起

使用的等；产品平台、屏幕大小、外形尺寸和输入方法的影响

注意事项：容纳对象的容器之间有比较关系或者要放在一起使用，则其应该是相邻的；表达一个过程中多个步骤的对象通常也要放在一起，并且遵循一定的次序

(4) 勾画大致的交互框架

最初阶段：界面的视觉化工作应该非常简单

方块图阶段：用粗略的方块图来表达并区分每个视图；方块图对应窗格、控制部件（如工具栏）；为每个方块图添加上标签和注解

注意：不要被界面上某个特殊区域的细枝末节分散了精力

(5) 构建关键情景剧本线路

特点：描述了人物角色如何同产品交互；可使用低保真草图序列的故事板

注意：这些场景剧本描述了人物角色最频繁使用界面的主要路径；必须在细节上严谨地描述每个主要交互的精确行为，并提供每个主要线路的走查

(6) 通过验证性的场景剧本来检查设计

验证性的场景剧本：验证性的场景剧本不用具备很多细节，但包含一系列“如果怎样，将怎样”的问题

验证性的场景剧本的类型：关键线路的变种场景剧本（关键途径的替代）；必须使用的场景剧本（必须要被执行但又不是经常发生的情况）；边缘情形使用场景剧本（非典型产品具备，但不太常用的功能）

2、设计策略

设计策略的分类：删除、组织、隐藏、转移

传统与现实认知的差别：

传统：功能越多，软件越强大，因此就越能获得用户的青睐和喜爱

现实：功能越多，越难发现对用户而言真正有价值的功能，同时还可能使遗留代码变得越来越沉重，系统的维护成本越来越高

(1) 删除：最明显的简化设计方法

好处：可以让设计师专注于把有限的重要问题解决好；有助于用户心无旁骛地完成自己的目标

如何删除：1、**关注核心**（关系到用户日常使用体验的功能；能够消除用户挫折感的功能）2、**砍掉残缺的功能**（删除实现得不够理想的功能；“为什么要留着它？”而非“为什么应该去掉它”；把注意力集中到客户的目的上面）3、**删除视觉混乱**（减少用户必须处理的信息，集中注意力在真正重要的内容上；减少元素大小的变化；减少元素形状的变化）4、**删减文字**（几乎任何句子都能够精简，几乎任何文字都可以删除）

精简文字的方法（by Richard Lanham）：不使用介词（对于/根据/为了/基于/通过/关于）；不使用is的动词形式，尽可能使用其他表达方式；把被动句式转换为主动句式；删除索然无味的开头，开门见山；减少废话。

不要过度删减：人们希望自己能够掌控局面（让人们能够控制结果；足够多的控制可以让他们消除因基本需求得不到满足而引发的焦虑；但要避免控制太多导致他们因选择而浪费时间）

(2) 组织：最快捷的简化设计方式

组织的方法：1、**分块**（可对命令分块，如菜单；可以按字母表或时间顺序排列的清单）；2、**围绕行为进行组织**（人们希望按照某种特定的步骤做事）；3、**确定清晰的分类标准**（建议多找一些用户，询问他们的分类标准）；4、**字母表与格式、时间和空间**；5、**利用不可见的网格来对齐界面元素**；6、**大小和位置**（重要的元素要大一些，不太重要的界面元素应该小一些[规则：如果一个元素的重要性为1/2，那就把它的大小做成1/4]；把相似元素放在一起）7、**感知分层的实现方式**（眯起眼睛观察屏幕，看是否能区分不同的层；如地铁线路图）8、**期望路径**（在描述用户使用软件的路径时，千万不要被自己规划图中清晰的线条和整洁的布局所迷惑）

(3) 隐藏：

一种低成本的简化方案：用户不会因不常用的功能分散注意力；可作为删除不必要功能的开始；必须仔细权衡要隐藏哪些功能

需要隐藏的元素：主流用户很少使用，但自身需要更新的功能；事关细节（对服务器进行配置或设

计邮件签名)；选项和偏好（修改绘图应用的单位）；特定于地区的信息（如时间和日期需频繁自动更新的信息）

隐藏的方法：1、**自定义**（自定义可能是一件非常耗费时间的事，且要求对软件中各种各样的功能了如指掌；主流用户感兴趣的是展示自己的个性；如果自定义的工具很简单，还是有价值的；**一般来说，不应该让用户去自定义他们的软件**）2、**渐进展示**（隐藏精确的控制部件，比自定义的效果更好；对于用户期望的功能，要在正确的环境下给出明确的提示）3、**适时出现**（过分强调隐藏功能会导致混乱；成功的隐藏：尽可能彻底地隐藏所有需要隐藏的功能，在合适的时机、合适的位置上显示相应功能）

让功能易于发现：1、**怎样介绍被隐藏在幕后的附件项？**（为隐藏功能打上标签：更多，高级；把标签放在哪里比把标签做多重要得多）2、**用户关注点**（用户在遇到问题的时候，过于关注屏幕上问题区域就算标签再大，放在用户关注点之外，用户也看不到）

(4) **总结：**删除不必要的，组织要提供的，隐藏非核心的

3、设计中的折衷

(1) 个性化和配置

问题：是否应该让产品具有用户定制功能？

个性化：人们喜欢改变周围的事物，使之适合自己；必须简单易用；在用户确定选择之前给他们一个预览的机会；必须容易撤销

配置：移动、添加或者删除持久对象；富有经验的用户所期望的；包含多种配置形式

(2) 本地化与国际化

在信息领域，国际化与本地化是指调整软件，使之能适用于不同的语言及地区的过程

国际化：指在设计软件时，将软件与特定语言及地区脱钩的过程；当移植到不同的语言及地区时，软件本身不用做内部工程上的改变或修正；意味着产品有适用于任何地方的“潜力”；只需做一次

本地化：指当移植软件时，加上与特定区域设置有关的信息和翻译文件的过程；为了更适合于“特定”地方的使用，而另外增添的特色；针对不同的区域各做一次

(3) 审美学与实用性

审美与实用的关系：一个漂亮的界面不一定就是一个好的界面，相反，理想情况下，只要界面的每一项都是精心设计的，在美学上该界面就是令人满意的。

审美与实用的冲突：为确保文本的可读性，文本的背景采用较低的对比度；复杂而强烈的对比可能获奖，但不实用

交互设计角度：根据语义和任务因素来进行视觉组织是最重要的；视觉美学的重要性稍低；换句话说，先实现一个良好的基本布局，然后再在这个基础上进行改进来实现好的美学效果

注意事项：组件之间的空白非常重要；组件的对齐会影响界面的可理解性和易用性

4、软件设计的细节

(1) **设计体贴的软件：**1、体贴的软件应该具有常识；2、体贴的软件是尽责任的；3、体贴的软件是自信的；4、体贴的软件不问过多的问题 5、体贴的软件知道什么时候调整规则；6、体贴的软件承担责任

(2) 加快系统的响应时间：

软件的空闲时间被浪费了，如何利用程序的空闲时间：对用户的可能操作作出几个假设；需要以全新并更主动的方式来思考软件能够怎样帮助人们实现其目标和任务

(3) 减轻用户的记忆负担

为了能够使用软件来完成某些任务，必须记住两类信息或知识：和软件如何操作相关（应当选择哪个命令或操作、文件存在哪个目录中等）；和该任务所需的领域知识相关（哪些系统函数可以使用，这些函数的参数及返回值是什么）

好的软件通过回忆用户上次的行为预测用户可能的操作：程序可以使用用户以前的设置作为默认值（如文档存放目录、窗口位置等）

(4) 减少用户的等待感

以某种形式的反馈让用户了解操作进行的进度和状态（如进度对话框）

以渐进方式向用户呈现处理结果（分成多个连续的部分来顺序地把结果提供给用户；先传输全局概括，再传输细节）

给用户分配任务，分散用户的注意力（如新功能介绍）

减低用户的期望值（告诉用户可能会需要几分钟）

(5) 设计好的出错信息

原则：1、使用清晰地表达语言，而非难懂的代码；2、语言应当精炼准确，而非空泛而模糊 3、对用户解决问题提供建设性的帮助；4、出错信息应当友好，不要威胁或责备用户

5、交互设计模式

(1) **模式的定义**：模式就是某个情形下某个问题的解决方案。描述了问题和解决方案，并说明了它成功应用于何处。

(2) **交互设计模式**：模式在 HCI 中的应用还处于起步阶段；模式捕捉的只是良好设计中不变的特性（具体实现，将取决于环境和设计者的创造性）；模式不是拿来即用的商品，每一次模式的运用都有所不同

(3) **具体的交互模式**：

双面板选择器：

定义：两个相邻面板，第一个面板显示一组对象，用户可从中选择；第二个面板显示选中对象的内容

应用场景：展现对象列表、类别列表、甚至动作列表的时候

要求：显示设备足够大

画布加调色板工具条

定义：带图标调色板工具条放在空白的画布旁边，用户单击调色板工具条上的按钮，在画布上创建对象

应用场景：任何图形编辑器

注意：使用该模式时，不必用上以文档为中心的桌面应用的所有招数

向导

定义：在界面上一步步引导用户按预定的顺序完成任务

应用场景：任务既长又复杂，且对用户而言是陌生的；适合的任务或者有分支，或者冗长且单调无趣，包括一系列需要用户决策的步骤，然后走向不同后续步骤

需要时显示

定义：把最重要的内容显示出来，并把其他内容隐藏起来。让用户通过一个简单的操作访问隐藏内容

应用场景：一个页面要显示很多内容，但是其中一部分内容不太重要。希望把界面变得简单一些，但是不得不把所有内容放到某个地方

基于 UML 序列图的导航

定义：根据跳转情况绘出 UML 图，然后在 UML 图上优化，将优化后的方案实现出来

应用场景：复杂繁多的跳转操作

金字塔

定义：使用后退/下一个链接来关联一系列页面，并将其和主页面结合，使由主页面能够连接到序列中的所有页面

应用场景：网站或应用有一系列页面时；一个页面链接到许多类似但语义无关页面时

序列地图

定义：每页显示一幅地图，把所有页面按顺序排序在地图上，包括“你在这里”的位置指示器

应用场景：导航的拓扑结构非常庞大，且拥有层级关系；用户的访问路径是线性的

面包屑层级结构

定义：在层级结构的每个页面，显示所有父级页面的链接，向上追溯到主页面位置；可以和序列地图互换

应用场景：应用和网站按树型结构组织；用户通过直接导航或搜索在树中上下移动

交互设计模式的三种类型：

①定位模式：应用于概念层面，帮助界定产品对用户的整体定位。定位模式的实例之一就是“暂态”，即使用很短的时间服务于一个在别处实现的高级目标

②结构模式：解答如何在屏幕上安排信息和功能元素之类的问题，包括视图、窗格，以及元素的组合等

③行为模式：旨在解决功能或数据元素的具体交互问题，大多数人所说的组件行为即属于此

第七章 可视化设计

现代 GUI 有时也被称为 WIMP 接口，这是因为他们是有窗口、图标、菜单和指点设备组成的

1、窗口

- (1) **窗口的状态：**最大化、最小化、还原（平铺[Tile]窗口、重叠[Overlapping]窗口、层叠[cascade]窗口）



- (2) **窗口界面类型：**多文档界面、单文档界面、标签文档界面

- (3) **单文档界面：**

优点：节省系统资源；最小的可视集；协同工作区；多文档同时可视化

缺点：菜单随活动文档窗口状态变化，导致不一致性；文档窗口必须在主窗口内部，减弱多文档显示优势；屏幕显示复杂：子窗口可能在父窗口中被最小化

- (4) **多文档界面：**

优点：从用户角度出发，以文档为中心；界面的视觉复杂性小

缺点：不能管理分散但相关的文档窗口；相关文档不能从相同类型的其他文档中分离；文档打开过多时，任务栏可能被占满

- (5) **标签文档界面（窗口菜单：包含了当前打开窗口的列表）：**

优点：让用户看到哪些窗口是打开的

缺点：不允许用户看到两个及以上的窗口内容

2、菜单

- (1) **访问系统功能的工具，已经成为窗口环境的标准特征：**某些功能可以立即执行，或由选择的菜单命令激活一个包含相关功能的对话框；菜单适合初学者，包含完整工具集合（不仅仅适用于初学者）

- (2) **必不可少的组成部分：**菜单标题、菜单选项

- (3) **最重要的特性：**描述性、一致性

- (4) **菜单栏：**所有窗口必备的基本组件；菜单栏是代表下拉式菜单的菜单；菜单选项的标签、位置、归类等均已标准化

- (5) 用于现代 GUI 的标准菜单包括文件、编辑、窗口和帮助；其他可选菜单包括试图、插入、格式、工具等，他们通常出现在编辑和窗口菜单之间。

- (6) **菜单设计原则：**

1、菜单应该按语义及任务结构来组织；

2、合理组织菜单接口的结构与层次；

3、菜单及菜单项的名字应符合日常命名习惯；

4、菜单选项列表即可以是有序的也可以是无序的（频繁使用的菜单项应当置于顶部）

5、为菜单项提供多种的选择途径（如热键、快捷键，要尽可能使用工业标准）；

6、菜单项的表示应该符合一些惯例（菜单系统的可浏览性和可预期性）；

7、对菜单选择和点取设定反馈标记（如灰色屏蔽，为选中的菜单项加边框在菜单项前面加✓符号等）

- (7) **菜单快捷键设计的建议：**

1、对所有的菜单选项都要提供一个辅助内存

2、使用菜单选项描述的首个字母作为快捷键。在出现重复的情况下，使用首个后续的辅音

3、在菜单中，对首个字符加下划线

4、尽可能使用工业标准

3、对话框：典型的辅助性窗口

- (1) **用途：**常用于将某些破坏性的、令人混淆的、不太常用的操作与主要工作中使用的工具分开考虑；通知用户系统的一个错误或潜在的问题；没有标题栏图标、状态栏和调整窗口大小的按钮
- (2) **分类：**1、按照与应用窗口的关系：模态对话框、非模态对话框；2、按照用途：属性对话框、功能对话框、进度对话框、公告对话框
- (3) **模态对话框：**冻结了它属于的应用，禁止用户做其他操作，直到处理了对话框中出现的问题；可以切换到其他程序进行操作；用户最容易理解，操作非常清晰
“应用模态”：只停止其所属的应用程序
“系统模态”：使系统中的所有程序都停止；大多数情况下，应用程序不应该有系统模态对话框
- (4) **非模态对话框：**打开后无须停止进度，应用程序也不会冻结；由于其操作范围不确定而难以使用和理解。

存在的问题：缺乏一致的终止命令，如取消、应用、关闭等

- (5) **属性对话框：**呈现所选对象的属性或者设置，并允许用户改变；模态、非模态均可；控制选择遵循“对象-动词”形式（选择对象，通过属性对话框为所选对象选择新的设置）
- (6) **功能对话框：**控制单个功能，如打印、拼写检查等；允许用户开始一个动作，并允许设置动作的细节（打印多少页、多少份、哪一台打印机）
注意：一个功能可以配置，不意味着用户每次调用都想配置；配置和实际功能最好隔离开
- (7) **进度对话框：**由程序启动，而不是根据用户请求启动（向用户清楚地表明正在运行一个耗时的进程；向用户清楚地表明一切正常；向用户清楚地表明进程还需多长时间；向用户提供一种取消操作和恢复程序控制的方式）；进度表应表明相对整个过程所耗费时间的进度，而不是相对整个过程规模的进度

对话框是一个单独的房间（Mozilla Firefox 和微软 IE 的网页浏览器，加载网页是内在操作，进度条被安放在其下方的状态栏中）

- (8) **公告对话框：**GUI 中滥用最多的元素；无须请求，由程序直接启动；阻塞型公告 VS. 临时公告（错误、警告和确认消息都是阻塞型公告对话框；为程序服务，牺牲用户利益，应尽量避免；绝不要用临时对话框作为错误信息框或确认信息框！）

特殊的对话框：错误、警告和确认

错误对话框：用户希望避免错误造成的结果，而不是错误消息

错误消息存在的问题：“用户经常因为产品设计中的错误而自责”；设计者和程序员角度（当人犯错时喜欢别人告诉他们；错误消息框在提醒用户一些严重的问题）；用户角度（愚蠢地停止了程序进度；责备自己之前，先责备发出消息的一方）；错误消息不起作用！

消除错误消息：用更健壮的软件取代错误消息（消除用户犯错误的可能性；为所有的数据输入使用有界控件）；为用户提供正面反馈（软件的负面反馈对用户而言是一种侮辱）；最后一招：改进错误消息框（保证始终有礼貌、具有启发性、还要助人为乐；澄清问题的范围；可选择的方法；默认情况程序会做什么；丢失了哪些信息）。

警告对话框：通知用户程序的行为，“确认”操作赋予用户忽略该行为的权利（停止了进度）；**建议：**软件不必阿谀奉承

确认对话框：程序对自己的行为不自信，用对话框来征求许可；总是来自程序，而不是用户（把责任推卸给用户）；对话框在喊“狼来了”（常规地方提供确认对话框，可使用户忽略真正的意外）

消除确认对话框：1、做，不要问（设计软件时，勇往直前给它确信的力量）；2、让所有操作都可以撤销（删除或覆盖文件时可先将文件移动到暂时目录，待一段时间或物理删除后再删除）；3、提供非模态反馈帮助用户避免犯错误（如文档页面应显示真实可打印区域的向导，以避免打印文档超出可用的打印区域）

- (9) **对话框的设计原则：**

- 1、把主要的交互操作放在主窗口中（对话框适合主交互流之外的功能）
- 2、视觉上区分模态与非模态对话框（为非模态对话框提供一致的终止命令）
- 3、不要用临时对话框作为错误信息框或确认信息框（保证用户能够阅读）

4、不要堆叠标签

(10) 管理对话框内容：

标签对话框：取代充斥大量控件的大对话框；拥有更多控件不意味着用户会觉得界面易于使用或功能强大（不同窗格内容必须有放在一起的道理；窗格组织为某个专题上的深度或广度增加）；成功的原因是其遵循了用户有关“事物存储”的心理模型，即单层分组（不同控件组成多个平行窗格，只有一个层次深度）**标签对话框建议：**不要堆叠标签

扩展对话框：新手用户不必面对复杂的工具，熟练用户也不必为寻找这些工具烦恼。

扩展对话框注意：必须小心设计；否则对初学者傲慢无礼，专家使用也有麻烦；最好记住上次被调用时所处的使用状态

级联对话框：糟糕的习惯用法；一个对话框的控件在一个层次关系的嵌套中调用另一个对话框；适合处理深度问题，但层次太深导致界面复杂

4、控件

(1) **控件性质：**用户和数字产品进行交流的屏幕对象（具有可操作性和自包含性）；控件的使用必须恰当且合理（大多数布满控件的对话框都不是好的用户界面设计）；

(2) **根据用户目标，控件分为四种类型：**命令控件、选择控件、显示控件、文本输入控件

(3) **命令控件：**接收操作并立即执行（如按钮：单击释放后立即执行；“可按压特性”，点击时视觉发生改变；不改变会使用户不安，“到底被按下了吗？”；图标按钮（butcons）：工具栏中的按钮）

要避免和超链接的混用：链接应该被用来浏览内容的，而其他操作应该采用按钮和图标按钮更为恰当

(4) **选择控件：**允许用户从一组有效的选项中选择一个操作数，还能用来设定操作；也可用来定义形容词或副词；部分情况下，选择控件也可以触发操作（如使用下拉列表调整字体大小等，较为自然；网页中下拉列表触发操作有可能把用户搞糊涂）

常见选择控件：复选框（单选框）、列表框、下拉列表

复选框：简单、可见、优雅；确切的文本使复选框清楚明确；但用户不得不放慢阅读速度；占据了数量可观的屏幕空间；方形复选框已经成为一个重要的标准

复选框变体：单选按钮（Radio Button）

列表框：允许用户从有限的文本字符串中选择，每个文本字符串代表一个命令、对象或属性；又称“列表框”或“列表视图”；标准的选择模式是相互排斥的（选择一个事物时，先前选定的事物就会取消）

列表框变体：下拉列表、列表中的图标、预览视图

(5) **显示控件：**显示和管理屏幕上信息的视觉显示方式（如管理对象在屏幕上的视觉显示方式的控件及静态显示只读信息的控件等）；包括屏幕分割线、标尺、导航栏、网格等

滚动条：适合用于窗口内容和文本导航器

滚动条包含的信息：1、当前可见文档的百分比；2、总共有多少页；3、拖动滑块时，显示页数；4、拖动滑块时，显示每一页的第一个句子；5、跳到文档开始和末尾的按钮

(6) **文本输入控件：**让用户在程序中输入新的信息，而不仅仅是从列表中选择；包括文本编辑字段、微调控件、标尺、滑动块、旋钮等；无界输入可能导致十分严重的问题

有界输入控件：用于在任何需要有数值界限的地方（杜绝“实际上不能却说可能”的粗鲁方式），如可以使用滑动条、微调控制项、标尺等

无界输入控件：允许用户输入任意文本值，包括文字和数字；如文本编辑控件；某些情况下，可接收值的集合是有限的，但是对列表控件来说数量太大；输入数据后加以验证（验证控件；为用户提供充分反馈）；保留输入的有效信息

5、工具栏（微软第一个将其引入用户界面，它以水平的方式置于菜单栏的下方，或者以垂直板的形式紧贴在主窗口的一边）

(1) **工具栏 vs 菜单：**都提供对程序功能的访问；菜单提供完整的工具集，主要用于教学；工具栏是为经常使用的命令设置的，对新手用户帮助不大（工具提示可以在一定程度上缓解这个问题）；“工具栏是单行(或单列)排列且始终可见的图形化立即菜单项”

(2) **工具栏上的图标和文字：**文本标签精确，但阅读与识别速度较慢；工具栏和菜单的用途不同（工具

栏主要为常用功能提供快速访问；图形的表意特征较文本更适合担当这种角色)；图标图像（找代表事物的图像比寻找代表动作或关系的图像容易得多）

(3) **为什么不能同时使用文本和图像？**

屏幕空间非常宝贵；

解决：工具提示

(4) **气球帮助 vs 工具提示**

气球帮助	工具提示
Apple 公司	Microsoft 公司
鼠标经过与气球弹出之间没有延迟	延时出现的时机非常好，大约 1 秒后显示帮助
气球会遮挡一大块应用的区域	只包含一个单词或非常短的词组
模态帮助系统，无法同时学习和使用	非模态帮助系统

(5) **工具栏的演化：没有理由将工具栏控件限制为图标按钮**

- (1) 状态指示工具栏控件
- (2) 工具栏上的菜单
- (3) 可移动工具栏
- (4) 可定制工具栏
- (5) 带条
- (6) 上下文工具栏

(6) **图标：小图像，用来表示系统对象、应用程序、实用工具、命令**

假设：图标能降低系统复杂度，使之更加容易学习和使用

问题：大量图标之间的区分；**解决方案**：通过颜色、形状或大小来区分同一类图标

图标设计原则：适合使用情境、适合当前任务

图标的四种表示方法：相似（通过类似的形象描绘的基本概念，如岩石坠落）；示例（一个典型的例子，如银器 -> 餐厅）；象征（表达更抽象图像，如破碎酒杯 -> 脆弱）；任意（承担没有关系的基本概念）

图标的注意事项：1、允许出现不同的图标状态；2、避免锯齿线；3、小心背景图案；4、坚持平台的图形风格；5、为最低的画面质量设计；6、颜色（微妙的，小的调色板，冗余）

图标的评估：1、图形的清晰度（图标显示了什么对象，图形分辨率如何？）；2、释义的准确性（图标是否显示了正确的概念？）；3、释义强度（图标率先表明了正确的概念？）；4、对比组的区别（每个图标是否能从组合中脱颖而出？）；5、对比组的包含（同一组的图标是否有某种关联？）；6、图形美观吗？

6、屏幕复杂性度量

(1) **布局复杂度**：根据可视对象的大小和位置来衡量

如果可视对象在高度和宽度上经常改变,以及对象与可视交互环境边界之间的距离比较大的话，就可以说这个布局较复杂

$$C = -N \sum_{n=1}^m p_n \log_2 p_n$$

C：以比特表示的系统复杂性；

N：所有组件的数量

m：组件分类的数目；

p_n：第 n 类组件出现的概率（以该类组件出现的频率为基础）

计算步骤：

- 1、使用矩形包围屏幕中的每个元素
- 2、计算屏幕中元素的数目及列的数目（垂直方向上的对齐点）

3、计算元素的数目及行的数目（水平方向上的对齐点）

改进算法：只需把可视组件总数加上顶端未对齐组件的个数和左侧未对其组件的个数，就可以得到相同的最终结果（屏幕中元素的个数；水平对齐点的个数；垂直对齐点的个数）

过分简单的接口会降低其实用性，具有中等布局复杂度的设计更好一些

(2) 布局统一度

布局复杂度 vs 布局统一度

布局复杂度	布局统一度
是严格的结构度量指标	只对界面组成部分的空间排列进行衡量
不考虑可视组件位于何处，只考虑组件在大小上的变化及其与边界之间的距离	不考虑这些组成部分是什么以及如何使用
在规划和改进真实设计的指南的使用范围相对有限	以“视觉上无序的排列有碍于可用性”这一原理为基础

规则：当可视组件整齐排列或者组件尺寸相差不大时，布局统一度就会提高

①左：各组件在大小或位置上没有一致性，布局统一度是 0%

②右：各组件的布局 and 大小完全一致，该接口布局统一度是 100%

③中庸的设计，布局统一度 82.5%

LU 介于 50%和 85%之间时是比较合理的

LU 计算公式：

$$LU = 100 \cdot \left(1 - \frac{(N_h + N_w + N_t + N_l + N_b + N_r) - M}{6 \cdot N_c - M}\right)$$

Nc：屏幕、对话框或其他界面组成部分上所有可视组件的总数

Nh、Nw、Nt、Nl、Nb、Nr：分别为可视组件的高度、宽度、顶端边距、做编辑、底边距和右边距

M：M = 2 + 2 · [2√N]；N 表示组件的数量

(3) 用户界面设计原理

(1) **结构原理：**根据清楚而一致的模型，以一种有意义和有用的方式对用户接口进行组织，把相关的东西放在一起，把不相关的东西分开放，区别不同的东西，使类似的东西看起来相似。

1、好的用户界面设计对用户界面的组织方式应符合所支持工作的结构，符合用户对工作的认识

2、过滥地运用隐喻会让用户界面难以理解

3、正文左对齐、右对齐和居中（使用垂直安排是一种不合适的可视隐喻）

4、缩进和边界对齐都与正文的边界有关（但这些在视觉上和逻辑上相关的功能却在空间上被分开安排）

(2) **简单性原理：**使简单、常用的功能简便易行，用用户自己的语言进行简明易懂的交流，对冗长的操作过程提供与其语义相关的快捷方式。

1、应从用户的角度来看什么任务更常用、什么任务更简单

2、并不一定非要进行调查获广泛研究

(3) **可见性原理：**让完成给定用户所需的所有选项和材料对用户可见，不要让额外或冗余的信息干扰用户。

1、在“所见即所得”的基础上，进一步实现“所见即所需”（设计目标是让所有需要和相关的选项可见和明确；不应让用户被不必要的信息所迷惑）

(4) **反馈原理：**通过用户所熟悉的清楚、简洁和无歧义的语言，让用户时刻了解系统对用户操作的反应和解释，了解与用户有关且被他们所关心的系统状态变化、出错、异常等所有情况。

1、成功的反馈就是以能被对方注意到、读到和正确理解的方式来提供信息（屏幕中央和顶端出现的反馈容易被注意；屏幕底端是最不容易被注意到的）

2、好的出错信息（以一个能让用户马上知道问题何在的题头开始；有针对性和简洁地解释问题究竟是什么；建议解决问题的方法或操作步骤）

(5) **宽容原理：**保持灵活和宽容，提供通过撤销和重做功能来减少用户出错和不当操作所带来的

开销，同时保持允许各种不同的输入形式和顺序以及通过合理地解释用户的所有合理操作来尽可能防止出错。

- 1、高可用性的系统帮助用户少犯错误
- 2、宽容性如何取决于将什么数据检测为有效以及何时进行这种合法性检查（检查后，在返回给用户的屏幕上将所有不合法的数据域置为高亮度，把光标放在第一个出错的数据域，并且在状态栏中给出解释的话；过分使用合法性检查会降低可用性）

(6) **重用原理**：对内部和外部的组件和行为加以重用，有目的而不是无目的地维持一致性，从而减少用户重新思考和记忆的需要。

- 1、用户界面内在外观、位置以及行为上的一致性使得软件容易学习和记忆如何使用（界面更具有可预测性和更容易理解）
- 2、重用程度越高，一致性就越好（不一致的用户界面不仅会降低软件的可用性，而且会增加程序设计的工作量）

第八章 交互设计模型与理论

1、预测模型：

- (1) **预测模型含义**：能够预测用户的执行情况，但不需要对用户做实际测试；特别适合于无法进行用户测试的情形；不同模型关注用户执行的不同方面
- (2) **主要的预测模型**：GOMS 模型（实际上是一个模型体系）；击键层次模型；Fitts 定律
- (3) **GOMS 模型**：最著名的预测模型

综述：是关于人类如何执行认知—动作型任务以及如何与系统交互的理论模型

采用“分而治之”的思想，将一个任务进行多层次的细化

把每个操作的时间相加就可以得到一项任务的时间

GOMS 的全称解释/组成元素：

- 1、Goal-目标：用户要达到什么目的（可分不同层次）；通常是层次化的（高层次目标可分解为若干个低层次目标）；通常表示为动作-对象序列（如复制文件，创建目录等）
- 2、Operator-操作：任务执行的底层行为，不能分解（为达到目标而使用的认知过程和物理行为）；用户为了完成任务必须执行的基本动作；操作种类分为外部操作（用户与系统之间可观测的物理操作）和心理操作（用户内部行为，不可观测，如假想、猜测等）；操作时间是上下文无关的（花费的时间与用户正在完成什么样的任务或当前的操作环境没有关系）
- 3、Method-方法：如何完成目标的过程，即对应目标的子目标序列和所需操作（外部 + 心理）；若用户能够执行一个任务，则表示其拥有该任务的一种方法（目标对应的方法不唯一）；
- 4、Selection-选择规则：选择规则是用户要遵守的判定规则，以确定在特定环境下所使用的方法；GOMS 中并不认为这是一个随机的选择，而是根据特定用户、系统的状态、目标的细节来预测要选择哪种方法

GOMS 方法步骤：

- 1、选出最高层的用户目标；2、写出具体的完成目标的方法（即激活子目标）；
- 3、写出子目标的方法（递归过程，一直分解到最底层操作时停止）；4、子目标的关系有顺序关系和选择关系（以 select：引导）

GOMS 分析：

- 1、优点：能够容易地对不同的界面或系统进行比较分析；GOMS 有助于确定新产品的有效性
- 2、局限性：假设用户完全按一种正确的方式进行人机交互，没有清楚地描述错误处理的过程；只针对那些不犯任何错误的专家用户，很难预测普通用户执行方式的有效性；任务之间的关系描述过于简单，只有顺序和选择关系；把所有任务都看作是面向目标的，忽略了用户间的个体差异

四种 GOMS 模型：击键层次模型（简化的假设）；CMN GOMS (Card Moran Newell, 伪码描述，结构严格)；NGOMSL (Natural GOMS Language, 程序形式，结构泛化；仅能够预测性能和学习次数)；CPM GOMS (Cognitive Perceptual Motor GOMS, 基于 Model-Human Processor；允

许操作符的并行操作)

- (4) **击键层次模型**：对用户执行情况进行量化预测（仅涉及任务性能的一个方面：时间）

用途：预测无错误情况下专家用户在下列输入前提下完成任务的时间；便于比较不同系统；确定何种方案能最有效地支持特定任务

操作符：

操作符名称	描述	时间（秒）
K	按下一个单独按键或按钮	0.35（平均值）
	熟练打字员（每分钟键入 55 个单词）	0.22
	一般打字员（每分钟键入 40 个单词）	0.28
	对键盘不熟悉的人	1.20
	按下 shift 键或 ctrl 键	0.08
P	使用鼠标或其他设备指向屏幕上某一位置	1.10
P ₁	按下鼠标或其他相似设备的按键	0.20
H	把手放回键盘或其他设备	0.40
D	用鼠标画线	取决于画线的长度
M	做某件事的心理准备（例如做决定）	1.35
R(t)	系统响应时间——仅当用户执行任务过程中需要等待时才被计算	t

使用：执行时间预测方法（列出操作次序，累加每一项操作的预计时间）

如何确定是否需要在具体操作之前引入一个思维过程：

以编码所有的物理操作和响应操作为开端。接着使用规则 0 放置所有的候选 M 操作符，然后循环执行规则 1 到 4，并对每一个 M 操作判断是否应该删除	
规则 0	在所有 K 操作符前插入 M 操作符，要求 K 操作的值不能是参数字符串（如数字或文本）的一部分。在所有的对应于选择命令（非参数）的 P 操作符前放置 M 操作
规则 1	如果某个操作符前的 M 操作完全可以由 M 之前的操作符预测，则删除 M
规则 2	如果一串 MK 组成的字符串是一个认知单元（如一个命令的名字），则删除除第一个 M 以外的所有 M
规则 3	如果 K 是一个冗余的终结符（如紧跟在命令参数终结符后面的命令终结符），则删除 K 之前的。
规则 4	如果 K 是常量字符串（如一个命令名）的终结符，则删除 K 之前的 M。如果 K 是变量字符串（如参数字符串）的终结符，则保留 K 之前的 M。

KLM 分析：建模可以给出执行标准任务的时间，但没有考虑下面的问题：

错误；学习性；功能性；回忆；专注程度；疲劳；可接受性

KLM 的应用：在交互设计早期阶段为用户性能提供有效、准确的模型

1、鼠标驱动的文本编辑（Xerox Star 研发过程）

2、查号工作站（贝尔实验室，探讨操作员流程的效率）

- (5) **Fitts 定律**：能够预测使用某种定位设备指向某个目标的时间（用户访问屏幕组件的时间对于系统的使用效率是至关重要的）；人机交互中，根据目标大小及至目标的距离，计算指向该目标的时间（可指导设计人员设计按钮的位置、大小和密集程度）；对图形用户界面设计有明显的意义；“最健壮并被广泛采用的人类运动模型之一”

“轮流轻拍”实验：尽可能准确而不是快速的轮流轻拍两个薄板；记录拍中和失误的情况；以实验数据为依据，得到困难指数如下： $ID = \log_2(2A/W)$

其中，A 为振幅（与目标的距离）；W 为目标宽度

改写之后的形式： $ID = \log_2(A/W+1)$ ；运动时间（MT）计算： $MT=a+b*\log_2(A/W+1)$

Fitts 定律的三个部分：

- 1、困难指数 ID (Index of Difficulty) = $\log_2(A/W+1)$ (bits): 对任务困难程度的量化; 与宽度和距离有关
- 2、运动时间 MT (Movement Time) = $a + b \cdot ID$ (secs): 在 ID 基础上将完成任务的时间量化
- 3、性能指数 IP (Index of Performance) = ID/MT (bits/sec): 基于 MT 和 ID 的关系; 也称吞吐量
Mackenzie 改写为: $ID = \log_2(A/W+1)$

1、更好地符合观察数据; 2、精确地模拟了支撑 Fitts 定律的信息论; 3、计算出的任务困难指数总是整数

a,b 的确定:

- 1、设计一系列任务, A 和 W 分别取不同的值
- 2、对每一种条件下的任务: 尝试多次; 记录每次执行时间; 进行统计分析
- 3、记录准确性: 记录选择的 x,y 坐标, 或错误率, 即鼠标落在目标区域外的百分比

说明: 1、如果 MT 的计算单位是秒, 则 a 的测量单位是秒, b 的测量单位是秒/比特(ID 的测量单位是比特); 2、系数 a(截距)和 b(斜率)由经验数据确定, 且与设备相关; 3、对于一般性计算, 可使用 $a=50, b=150$ (单位是毫秒); 4、A 和 W 在距离测量单位上必须一致, 但是不需要说明使用的具体单位

Fitts 定律建议:

- 1、大目标、小距离具有优势 (对选择任务而言, 其移动时间随到目标距离的增加而增加, 随目标的大小减小而增加);
- 2、屏幕元素应该尽可能多的占据屏幕空间
- 3、最好的像素是光标所处的像素
- 4、屏幕元素应尽可能利用屏幕边缘的优势
- 5、大菜单, 如饼型菜单, 比其他类型的菜单使用简单

Fitts 定律应用:

- 1、首先被 Card 等人应用在 HCI 领域: 鼠标的定位时间和错误率都优于其他设备; 鼠标速率接近最快速率; 使用鼠标完成运动任务比使用其他设备更加协调, 这在交互设计中非常重要
- 2、策略一: 缩短当前位置到目标区域的距离 (如右键菜单技术)
- 3、策略二: 增大目标大小以缩短定位时间 (Windows 操作系统和 Macintosh 操作系统中的应用程序菜单区域位置的设计)

Mac OS 的菜单是沿着屏幕边缘排列的 (用户往往在距离屏幕边缘 50 毫米处停下来, 50 毫米作为 Mac OS 的菜单宽度)

Windows OS 的菜单位于标题栏下面

Mac OS “dock”:

工具栏组件大小可以动态改变: 1、为用户提供了一个放大的目标区域; 2、可显示更多图标; 3、新版 Mac 操作系统中都实现了扩展工具栏

2、动态特性建模:

(1) 状态转移网络:

用途: 用于描述用户和系统之间的对话; 可被用于探讨菜单、图标和工具条等屏幕元素, 还可以展示对外围设备的操作; 适合表达顺序操作和循环操作

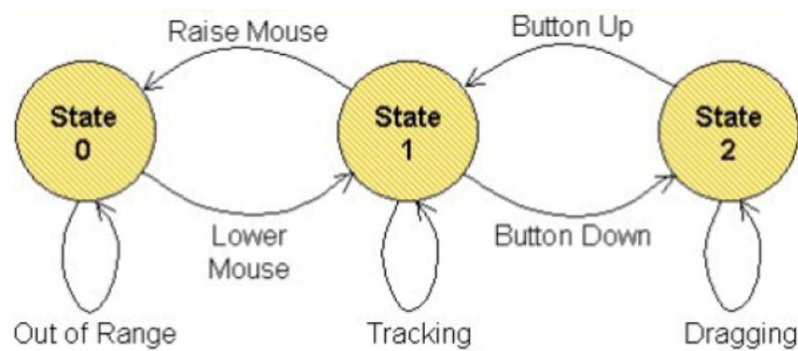
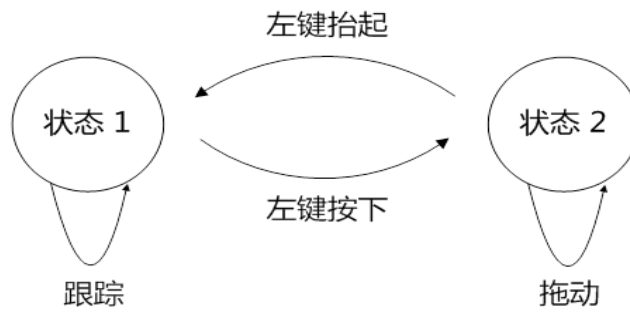
性质: 最常用的状态转移网的形式; 有向图; 图中的结点表示系统的各种状态; 图中的边表示状态之间可能的转移; 状态之间通过转移 (用带方向箭头的线段表示) 互相连接; 转移被事件 (转移线段上的标记) 触发; 伪状态——初始状态和终止状态 (是 STNs 的起始和终止; 可以与系统的其他部分相连接)

(2) 三态模型:

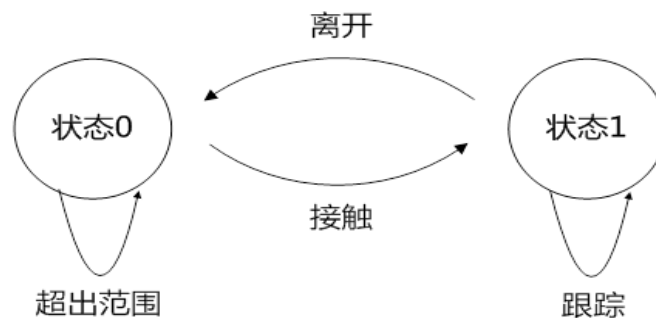
用途: 帮助设计者为特定交互设计选择合适的 I/O 设备 (三态模型能够体现设备间的关键差别); 用于对指点设备建模; 将指点设备的操作使用状态转移来表示

指点设备的状态: 无反馈运动 (状态 0); 跟踪运动 (状态 1); 拖动运动 (状态 2)

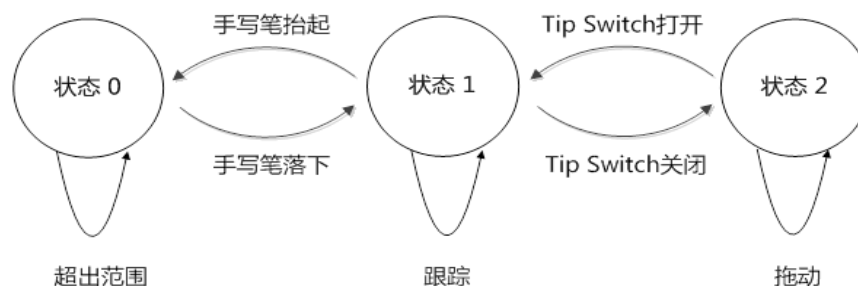
鼠标的三态模型:



触摸板的三态模型：



手写笔的三态模型：



比较：鼠标和手写笔等在 s0 状态下的行为是存在差异

鼠标或触摸板：系统保存鼠标被拿起前的位置和手指离开触摸板前的位置；位置独立 s0

手写笔或光笔：重新参与到系统后指示设备当时的位置依赖于设备的位置；位置依赖 s0

3、语言模型：BNF, Backus-Naur Form

用途：用户和计算机的交互通常是通过一种语言进行考察的；BNF 语法常用于说明对话；目的在于理解用户的行为和分析认知界面的难度

名称类型：1、非终止型：小写字母；2、终止型：大写字母

符号：“::=” 读作“定义为”

操作符：“+”（序列）和“|”（选择）

界面分析：

方法一：计算规则的数目

规则越多，界面就越复杂

缺点：对于描述界面的确切方式是相当敏感的

方法二：计算“+”和“|”操作符的数目

更健壮

缺点：使较复杂的单个规则处于不利地位，可能存在错误判断

4、系统模型：

出发点：为评估系统的可用性，需要知道系统做的是什

分类：1、基于模型的标记法，描述系统的状态和操作（Z 标记法）；2、代数标记法，描述动作序列的作用；3、时序和义务逻辑，描述事情什么时候发生，以及由谁负责

面向模型的标记法：

- 1、源于 20 世纪 70 年代后期
- 2、反映了用在真实编程语言中的多种结构
- 3、提供一种描述软件系统行为的方法
- 4、用一种和系统怎样编程密切相关的方式和更抽象的语言描述软件系统行为（允许设计人员在设计的早期不考虑机器或者实现；使设计或者规约的推理更为严密）
- 5、主要方法有：Z 标记法和 VDM 标记法

Z (Zed) 标记法：

1、**基于集合和函数：**最简单的集合对应于编程语言中的标准类型；如实数 R 、整数 Z 和自然数 N 等

2、**非标准类型定义方式 1：**显式列出集合中有限的可能值就可以定义新集合

`Shape_type ::= Line | Ellipse | Rectangle`

`Keystroke ::= a | b | ... | z | A | ... | 9 | Cursor_left`

非标准类型定义方式 2：用方括号括起集合符号，如 `[Keystroke]`；可表示集合的存在而不提供集合内容的定义

基本集合可构建复杂集合：如已命名无序元组、序列和函数等，如 `Point == R × R`

3、**函数：**具有程序语言中标准计算的功能；表达用户在某个图形编辑器的绘画中建立的形状集合时，`Shape` 模式不能挑出单个或多个形状；通过命名一个把形状和标识符映射起来的函数，可以表达一组相同形状
`Shape_dict == Id → Shape`

4、**状态和不变式：**模式 `State` 包括两个部分，即中间线的上部和下部。中间线的上部定义了系统的状态，中间线的下部是状态的不变式。在没有特殊说明的情况下，上下部中间的分割线可以理解为逻辑符号“与”

5、**操作符定义：**

- (1) 为定义一个操作符，必须描述绘图系统在操作完成前后的状态：一个命名为 `State`，表示操作完成前的状态；一个命名为 `State'`，表示操作完成后的状态。对需要用户输入的操作，相应操作符后面需要跟随一个问号（?）；如果操作会提供输出，那么操作符后面跟随一个感叹号（!）来表示。

第九章 以用户为中心的设计

以用户为中心的含义：把人置于系统设计过程的中心；由纯粹的技术问题演变成为关于用户自身的问题

以用户为中心：

- 1) 以真实用户和用户目标作为产品开发的驱动力，而不仅仅是以技术为驱动力
- 2) 应能充分利用人们的技能和判断力，应支持用户，而不是限制用户
- 3) 需要透彻了解用户及用户的任务，并使用这些信息指导设计
- 4) 这是一种设计思想，而不是纯粹的技术

1、**以用户为中心的设计思想：**美国加州大学圣地亚哥分校的唐纳德诺曼（Don Norman）研究实验室首创；侧重以人为本；重点是获得对使用拟设计系统的人的全面了解

- (1) **以产品为中心：**传统“以产品为中心”的设计过程主要关注的是机器及其效率，同时期望人能够适应机器

(2) 以用户为中心的三个方面的假设:

- 1、好的设计结果使客户感到满意
- 2、设计过程是设计人员与客户之间的协作过程。设计要进化并适应客户不断变化的考虑,作为一种副产品,这个过程要产生规格说明
- 3、在整个过程中,客户和设计人员要不断沟通

(3) 以用户为中心的四个重要原则:

- 1、及早以用户为中心:设计人员应当在设计过程的早期就致力于了解用户的需要
- 2、综合设计:设计的所有方面应当齐头并进地发展,而不是顺次发展,产品的内部设计与用户界面的需要应始终保持一致
- 3、及早并持续性地进行测试:当前对软件测试的唯一可行方法是总结经验总结出的方法,即若实际用户认为设计时可行的,它就是可行的。
- 4、迭代开发:大问题往往会掩盖小问题的存在,设计人员和开发人员应当在整个测试过程中反复对设计进行修改

(4) UCD 项目通常包含的方法:

- 1、**用户参与**:在有些项目中,用户会成为设计团队的一部分,并在设计过程中不断被请教、被资讯。(然而,如果开发时间较长,用户可能逐渐变得像开发人员,进而不能很有效地表达用户的需要,所以有的时候会采用轮流机制)
 - 2、**焦点小组**:焦点小组通常用于产品功能界定、工作流程模拟、用户需求发现、用户结构设计和交互设计、产品的原型测试、用户模型的建立等。设计人员与不同用户进行交流,观察他们之间的如何相互联系,进而获得用户对产品的意见和态度;需要一个有经验的主持人来维持主题
 - 3、**问卷调查**:问卷调查可以获得地理位置上分散的大量用户群体的信息,但是相比焦点小组问卷有更多的限制;无法直接接触被调查的人,所以问卷无比清楚且使用直观的度量方法
 - 4、**民族志观察**:民族志观察发生在完成工作的地方;包括受访问者的日常事务,对受访问者进行追踪。
 - 5、**走查**:执行一次走查指确定用户的目标或者任务,然后浏览一个设计提议来查看任务是否能够被完成;意味着发现所有在实际使用中可能出现的问题,可以专注于设计的某一个具体的方面或者整个设计
 - 6、**专家评估**:专家评估师可用性专业人员通过使用来自于可用性研究和著作的启发是或者指导性准则进行实施的。它从设计团队提供的用户观察点对产品设计进行评估。
 - 7、**可用性测试**:可用性测试可以采用多种不同的形式。它们既可以在昂贵的可用性实验室中,通过使用特殊的镜子、摄像机、跟踪软件及其他的观察设备来完成,也可以不使用任何技术而只是简单地使用一个纸上原型、一个观察者和一个受访问者完成
- (5) **软件工程**:源于 20 世纪 60 年代中期;把设计看作是定义规格说明并通过规格说明导出系统的一种形式化过程;产生拥有客户所描述的形式和功能的系统的过程中
- (6) **工程设计过程的三个假设**:
- 1、设计的结果是一种产品(成品、机器或系统);
 - 2、产品通过由客户给出的规格说明导出。原则上,有了足够的知识和计算能力,这种导出就可以被机械化;
 - 3、一旦客户和设计人员就规格说明达成一致,直到交付之前,客户和设计人员之间不大可能再需要进行接触。

2、用户参与设计

(1) 用户参与的重要性

原因:项目初期调研对用户的了解程度仍不足以解决设计过程中出现的所有问题

选择用户的标准:将来真正使用系统的人

用户参与的原因:

- 1、期望管理:保证产品不会出乎用户的意料,避免失望;对用户进行培训也有助于改进期望管理
- 2、拥有权:人们总渴望自己的意见得到重视;更容易接受有“拥有权”的最终产品

(2) 用户参与的形式

- 1、作为设计组成员，全职或兼职参与整个项目或部分项目开发

优点：用户能透彻理解系统及其原理

缺点：用户会逐渐变得不具有普通用户的代表性

注意：对大型项目应注意用户更新

- 2、以邮件、专题讨论或类似会议的形式参与，定期接收有关项目进展的信息

- 3、存在大量的用户时的折衷方案

各个用户组派代表以全职的形式加入设计组；其他人员通过专题讨论等形式参与

注意事项：要想实现一个成功的交互式系统，企业政策和个人爱好比技术上的问题更重要；关于参与式设计的许多建议都是从不同的用户哪里提炼出来的，这些用户既包括儿童，也包括老人，有些用户没有人知能力或者是件很宝贵；任何形式的用户都涉及成本问题

用户过于投入开发的负面影响：随着项目的进行，用户会提出更复杂的想法并希望实现它们；用户害怕失去工作或担心增加工作负担；用户是不可预测的，而且对软件开发不够“宽容”

- (3) **参与式设计：**让用于参与开发的一种方法；用户成为设计小组成员，与设计人员合作设计产品；制作模型能够有效地利用用户的经验和知识

两种基于纸张的原型制作技术：PICTIVE、CARD

- 1、**PICTIVE（协作式产品的界面造型技术）：**使用低保真的办公室用品模型来研究系统的特定屏幕和窗口布局

目的：使得用户能够参与设计过程；改进设计过程的知识获取方法

四个阶段：当事人做自我介绍；简短讲解说明不同应用域；围绕设计的集策讨论；设计走查和决策讨论

会议准备：

- 1) 明确并要求有代表性的“当事人”：包括用户、设计人员、实现人员等
- 2) 出席者的家庭作业：用户要写出他们工作中最重要或最常执行任务的分布场景描述；设计人员需要准备一份调查发现的有关需求和问题的简短描述；设计人员要求准备系统实现中可能使用到的组件以及技术方法的基本资料

会议开始：

- 1) 自我介绍
- 2) 作业展示：需求为后续讨论打下基础；用户使用场景能够帮助设计人员和实现人员了解系统应如何支持用户工作；实现人员有关技术细节的解释帮助用户建立哪些是能够做到的头脑模型（展示是相互学习的过程）

参与设计：

- 1) 设计组提出问题，并要求就问题给出解决方案：与会者使用纸、笔、剪刀以及菜单和对话框等的模拟工具
- 2) 会议应该基于所有参与者的头脑风暴过程展开，而非由特定人员引导的讨论：每个参与者都有机会展示和解释各自的设计思想
- 3) 录像过程：记录完整的决策过程以及作出的设计决策，用于后续分析；关注的焦点在于为什么特定设计方法提出的动机

- 2、**CARD（需求和设计的协作分析）：**使用画有计算机和屏幕图像的卡片发掘各种工作流；是“情节串联图”的一种形式

- 3、**CARD & PICTIVE：**

侧重点不同：PICTIVE 关注的是系统细节，而 CARD 注重的是宏观的任务流
可互为补充

3、理解用户工作

理解用户的思考角度：

- (1) 他们是谁？（年龄、计算机水平、文化等）
- (2) 可能不像你（设计人员不能依赖自己对用户界面问题的直觉）
- (3) 找他们谈话（鼓励用户告诉你真实的情况）
- (4) 观察他们（并非所有事情都可以说清楚）

(5) 应用你的想象力（十分危险！角色是良好的帮助信息）

上下文询问法：又称情景调查；观察并与用户交流会比仅仅观察的效果要好；强调到用户工作的地方，在用户工作时观察，并和用户讨论他的工作；基于“学徒模型”（用户是师傅，研究人员是学徒）

上下文询问法与观察法的区别：用户知道研究人员的存在；也知道他们是研究的一部分

上下文询问法的四个原则：

- (1) 上下文环境：应深入工作空间，以了解其中发生的事情；可以要求用户边做边说，也可以只在必要时发问
- (2) 伙伴关系：开发人员和用户应相互合作；提醒用户是专家，将研究人员作为新手
- (3) 解释：解释过程必须由用户和开发人员合作完成；杜绝设计人员片面地对事实作出解释或假设
- (4) 焦点：把问题集中在所定的研究题目上；准备一个观察方向的列表

上下文询问法 vs 民族志观察：

- (1) 过程更简短（2 至 3 个小时 vs. 数周或数月）
- (2) 重点更为明确、集中（民族志观察的角度更广）
- (3) 设计人员只询问，不参与
- (4) 目的明确—设计新系统
- (5) 完成访谈之后，应尽快回顾这些数据，并建立正式文档（如 workflow 模型、顺序图、文化图和物理模型等）

4、以用户为中心浅析

好的以用户为中心的设计能保证产品可以工作，人们能够使用，进而避免失败

Larry Kelley 的三品质概念模型：

- (1) 可能性：由技术专家决定；什么是可以建造的，什么是不可以建造的
- (2) 可行性：由商务人员决定；销售什么才能获得利润
- (3) 期望值：由设计师决定；什么能给人们带来高兴和满意

以活动为中心：

“这个世界上的大多数东西都是在没有得益于用户研究和以人为中心的设计方法的情况下被设计出来的”

- (1) **以用户为中心方法的缺陷：**影响产品的创新性；可操作性受到时间、预算和任务规模的限制；忽视了人的主观能动性和对技术的适应能力
- (2) **以活动为中心的设计思想：**把用户要做的“事”（活动）作为重点关注的对象；更适合于复杂的设计项目

ACD 思想是对 UCD 思想的一种反思：早期的设计是以技术为中心；直到出现以人为中心，这是一次飞跃；ACD 把人与技术综合起来进行考虑，不单纯考虑人或者技术，而是关注事情本身的活动目标；ACD 同样需要对用户进行研究或调研

倾听用户永远是明智的，但屈从于用户的要求是不明智的

第十章 评估的基础知识

1、评估基本概念

评估总是需要的

评估的定义：系统化的数据搜集过程；目的是了解用户或用户组在特定环境中，使用产品执行特定任务的情况；问题既可以涉及高层的目标，也可以十分具体

评估不是设计过程中一个单独的阶段：优秀的交互设计师应掌握如何在不同的开发阶段评估不同的系统

2、评估目标和原则

评估的五个好处：

- (1) 能够在产品交付之前修正错误
- (2) 设计小组能够专注于真实问题，而不是假想问题
- (3) 工程师们能专心于编程而不是争论
- (4) 能够大大缩短开发时间

(5) 在发布了第一版之后，销售部门即可获得稳定的设计，在销售时也就不必猜想后续版本将如何工作

评估的三个目标：

- (1) 评估系统功能的范围和可达性
- (2) 评估交互中用户的体验
- (3) 确定系统的某些特定问题

评估原则：

- (1) 评估应该依赖于产品的用户（与专业技术人员的水平和技术无关）
- (2) 评估与设计应结合进行（仅靠用户最后对产品的一两次评估，是不能全面反映出软件可用性的）
- (3) 评估应在用户的实际工作任务和操作环境下进行（根据用户完成任务的结果，进行客观的分析和评估）
- (4) 要选择有广泛代表性的用户（参加测试的人必须具有代表性）

3、评估范型和原则

基本概念：

- 1、**用户研究：**用户研究就是研究人们在实际工作环境中或者在实验室中，如何使用新、旧技术执行任务
- 2、**评估范型：**范型与具体学科相关，对如何评估有很大影响（可用性测试是一种评估范型）
- 3、**评估技术：**每种范型有特定的技术（可用性测试的技术有观察、问卷调查、访谈等）

四种评估范型类别：快速评估、可用性测试、实地研究、预测性评估

快速评估：设计人员非正式地向用户或顾问了解反馈信息，以证实设计构思是否符合用户需要。基本特征：快速

- 1、可在任何阶段进行
- 2、强调“快速了解”，而非仔细记录研究发现（如在设计初期了解用户对新产品的意见、在设计末期了解用户对图标设计的看法等）
- 3、得到的数据通常是非正式、叙述性的（可以口语、书面笔记、草图、场景的形式反馈到设计过程）
- 4、是设计网站时常用的方法

可用性测试：20 世纪 80 年代的主导方法

- 1、评测典型用户执行典型任务时的情况：包括用户出错次数、完成任务的时间等
- 2、基本特征：是在评估人员的密切控制之下实行的
- 3、主要任务：量化表示用户的执行情况
- 4、缺点：测试用户的数量通常较少；不适合进行细致的统计分析（不进一步处理变量值）

现场研究：现场研究是在真实的工作环境中进行的，这是它的基本特征，目的是理解用户的实际工作情形以及技术对他们的影响。

基本特征：在自然工作环境中进行

目的：理解用户的实际工作情形以及技术对他们的影响

作用：探索新技术的应用契机；确定产品的需求；促进技术的引入；评估技术的应用

分类：评测人员作为“局外人”；评测人员也可作为“局内人”或测试用户

现场研究泛型包括：访谈、自然观察、用户观察、现场研究

预测性评估：

两种方法：1、专家们根据自己对典型用户的了解（通常使用启发式过程）预测可用性问题；2、基于已有的理论模型得出评估结论

基本特征：用户可以不在场；使得整个过程快速、成本较低

启发式评估是典型的预测性评估方法（注意：启发式原则应定制）：可能误导设计人员，且有些结果可能并不准确

评估范型比较：

评估范型	快速评估	可用性测试	实地研究	预测性评估
用户角色	自然行为	执行测试任务集	自然行为	用户通常不参与
控制权	评估人员实施最低限度控制	评估人员密切控制	评估人员与用户合作	评估人员为专家
评估地点	自然工作环境或实验室	实验室	自然工作环境	类似实验室的环境，通常在客户处进行
适用情形	快速了解设计反馈。可使用其他交互范型的技术，如专家评测	测试原型或产品	常用于设计初期，以检查设计是否满足用户需求，发现问题，发掘应用契机	专家（通常是开发顾问）检查原型，可在任何阶段进行。使用模型评测潜在设计的特定方面
数据类型	通常是定性的非正式描述	量化数据，有时是统计数据。可采用问卷调查或访谈搜集用户意见	应用草图、场景、例证等的定性描述	专家们列出问题清单，由模型导出量化数据，如两种设计的任务执行时间
反馈到设计	通过草图、例证、报告	通过性能评测、错误统计报告等为未来版本提供设计标准	通过描述性的例证、草图、场景和工作日志	专家列出一组问题，通常附带解决方案建议。为设计人员提供根据模型计算出的时间值
基本思想	以用户为中心，非常实用	基于试验的实用方法，即可用性工程	可以是客观观察或现场研究	专家检查以实用的启发式原则和实践经验为基础，采用基于理论的分析模型

评估技术类型：观察用户、询问用户意见、征求专家意见、用户测试、基于模型的分析评估

观察用户：有助于确定新产品的需求，也可用于评估原型

挑战：如何在干扰用户的前提下观察用户，以及如何分析大量数据

询问用户意见：简单，调查用户数量从几个到几百不等

询问专家意见：“角色扮演”方式评估；同时专家会提出解决方案

测试用户的执行情况：可比较不同设计方案优劣；通常在受控环境中进行

基于模型和理论，预测界面的有效性：常用技术如 GOMS 模型和 KLM 模型等

评估范型和评估技术的关系：

评估技术	评估范型			
	快速评估	可用性测试	实地研究	预测性评估
观察用户	观察用户实际行为的重要方法	使用摄像和交互日志的记录方式，可做进一步分析，以找出问题，了解操作步骤，计算执行时间	实地研究的核心方法。在现场研究中，评测人员与测试环境相融合；在其他类型的研究中，评测人员只做客观观察	——
询问用户意见	与用户和潜在用户讨论，可采用个别会谈、集体会谈或专门小组的形式	通过问卷调查了解用户满意度，也可通过访谈了解更多详情	评测人员可采用访谈的形式，与用户讨论观察到的问题。现场研究可采用现场访谈	——
询问专家意见	专家评估原型的可用性（提供“评估报告”）	——	——	在设计初期，专家使用启发式原则预测界面的有效性
用户测试	——	在受控环境中，测试典型用户执行典型任务的情况，是可用性测试的基本方法	——	——
用户执行情况 分析模型	——	——	——	使用分析模型预测界面的有效性，或比较用户使用不同设计方案的执行效率

4、评估方法的选择

(1) 区分评估技术的因素：

- 1、评估在周期中的位置： 评估在开发过程中所处的阶段是影响评估方法选择的首要因素；设计早期阶段的评估更快速、便宜。
- 2、评估的形式： 实验室环境 or 工作环境；实验室研究允许进行受控制的实验和观察，然而损失了一些自然的东西，现场研究保留了自然的情况，但是丧失了对环境和用户的控制能力
- 3、技术的主客观程度： 技术越主观，受评估人员知识的影响越大（如认知走查等）
- 4、测量的类型： 与技术的主客观性有关（主观技术：定性数据；客观技术：定量数据）
- 5、提供的信息（低层信息：这个图标是可理解的吗？高层信息：这个系统是可用的吗？）
- 6、响应的及时性：边做边说法可及时记录用户行为；任务后的走查取决于对事件的回忆
- 7、干扰程度：直接响应测量可能会影响用户表现
- 8、所需资源：设备、时间、资金、参与者、评估人员的专业技术及环境等

- (2) 评估技术分类：一方面，不同的评估方法应用于软件开发生命周期的不同阶段；另一方面，他们各有优缺点，能够在某些程度上相互弥补

评估方法中最简单的两种是观察和边做边说，因为它们把大部分“工作”都留给用户来做

方法	生命周期阶段	用户人数	主要优点	主要缺点
启发式评估	早期设计，反复设计过程的“内循环”	无	能发现单个可用性问题，能发现熟练用户碰到的问题	没有涉及真实的用户，故无法再用用户需求方面有“惊人发现”
绩效度量	竞争性分析，最终测试	至少 10 人	硬性数据，对结果容易进行比较	不能发现单个可用性问题
边做边说	反复设计，形成性评估	3~5 人	准确了解用户的错误想法，测试费用低	用户感到不自然，熟练用户感到很难用语言表达
观察	任务分析，后续研究	3 人或以上	生态有效性；发现用户的真实人物；建议系统功能与特征	很难约定安排，实验人员无法控制
问卷调查	任务分析，后续研究	至少 30 人	发现用户主观偏好，容易重复进行	需要进行问卷预答（避免出现误解）
访谈	任务分析，后续研究	5 人	灵活，可以深入了解用户观点和用户体验	耗时，难以进行分析、比较
焦点小组	任务分析，用于参与	每组 6~9 人		分析起来困难，有效性低
使用过程记录 用户反馈	最终测试，后续研究 后续研究	至少 20 人 上百人	跟踪用户需求和想法上的变化	需要专门部门来处理回复

- (3) 评估方法组合：评估方法的组合取决于项目待评估的具体特性

常用组合：

- 1、启发式评估+边做边说等用户测试技术
专家可通过启发性评估排除显而易见的可用性问题；重新设计后，经用户测试，反复检查设计的效果
- 2、访谈+问卷调查
先对小部分用户进行访谈，确定问卷中的具体问题
- 3、启发式评估 vs. 用户测试
前者不需要用户参与；二者发现的可用性问题不同，可以互补

5、评估步骤

DECIDE 评估框架：

- (1) 决定评估需要完成的总体目标
- (2) 发掘需要回答的具体问题
- (3) 选择用于回答具体问题的评估范型和技术
- (4) 标识必须解决的实际问题，如测试用户的选择
- (5) 决定如何处理有关道德的问题

(6) 评估解释并表示数据

确定目标：评估目标决定了评估过程，影响评估范型的选择

评估的原因：产品设计是否理解了用户需要？为概念设计选择最佳隐喻？界面是否满足一致性需要？探讨新产品应做的改进？

发掘问题：根据目标确定问题，问题可逐层分解

目标：找出为什么客户愿意通过柜台购买纸质机票，而非通过互联网购买电子机票

问题：用户对新票据的态度如何（是否担心电子机票不能登机）；用户是否能够通过互联网订票；是否担心交易的安全性；订票系统的界面是否友好（是否便于完成购票过程）

选择评估范型和技术：范型决定了技术类型

必须权衡实际问题 and 道德问题：

- (1) 最适合的技术可能成本过高
- (2) 最适合的技术所需时间过长
- (3) 最适合的技术不具备必要设备和技能

可结合使用多种技术：

- (1) 不同技术有助于了解设计的不同方面
- (2) 不同类型数据可从不同角度看待问题
- (3) 组合有助于全面了解设计的情况

明确实际问题：

- 1、用户：应选择恰当的用户参与评估（能代表产品的目标用户群体；可以先做测试，确定用户技能所属的用户群）；任务时间多长（20 分钟休息一次）；可在任务执行前，安排用户熟悉系统
- 2、设施及设备（如需多少台摄像机录像，具体摆放在何位置）
- 3、期限及预算是否允许
- 4、是否需要专门技能（没有可用性专家）

处理道德问题：

- 1、应保护个人隐私：
 - (1) 除非获得批准，否则书面报告不应提及个人姓名，或把姓名与搜集到的数据相联系
 - (2) 受保护的个人资料包括健康状况、雇佣情况、教育、居所和财务状况等
 - (3) 可在评估前签署一份协议书
- 2、指导原则：
 - (1) 说明研究的及要求参与者做的工作
 - (2) 说明保密事项，对用户&对项目
 - (3) 测试对象是软件，而非个人
 - (4) 对测试过程的特殊要求，是否边做边说等
 - (5) 用户可自由表达对产品的意见
 - (6) 说明是否对过程进行录像（不能拍摄用户的面部）
 - (7) 欢迎用户提问
 - (8) 用户有随时终止测试的权利
 - (9) 对用户话语的使用应征得同意，并选择匿名方式

评估、解释并表示数据：

- 1、搜集什么类型的数据，如何分析，如何表示：通常由评估技术决定
- 2、可靠性：给定相同时间，不同时间应用同一技术能否得到相同结果；非正式访谈的可靠性较低
- 3、有效性：能否得到想要的测量数据
- 4、偏见：评估人员可能有选择地搜集自己认为重要的数据
- 5、范围：研究发现是否具有普遍性
- 6、环境影响：霍桑效应

6、小规模试验：

作用：

- 1、确保评估计划的可行性

- 2、检查设备及使用说明
- 3、练习访谈技巧
- 4、检查问卷中的问题是否明确

小规模试验可进行多次：类似迭代设计；测试——反馈——修改——再测试；快速、成本低

7、可用性问题分级方法：评估结果总是可用性清单，以及改进建议

(1) 方法一 基于量化数据的分级（如多少人遇到该问题，耗费多少时间等）

(2) 方法二 问题严重性的主观打分，取平均值

0：不是一个可用性问题

1：一个表面的可用性问题（如果项目时间不允许，可不予纠正）

2：轻微的可用性问题（优先级较低）

3：重要可用性问题（需要重视，给以高优先级）

4：可用性灾难（产品发布之前必须纠正）

(3) 方法三 可用性分级的两个因素：多少用户会遇到这个问题；用户受该问题影响的程度

问题对用户的影响程度 \ 遇到问题的用户比例	少	多
	低严重性	中严重性
小	低严重性	中严重性
大	中严重性	高严重性

(4) 方法四：该问题只在第一次使用时出现，还是会永远出现

第十一章 评估之观察用户

1、观察综述：

(1) 观察用户怎样工作是极其重要的可用性方法：

用户并不总能客观和完整地描述产品的使用情况（访谈得到的信息可能不真实）

用户有可能忽略一些细节（忽视了细节的重要性）

(2) 观察法是所有可用性方法中最简单的方法：涉及看和听两个方面

(3) 观察适用于产品开发的任何阶段

初期：理解用户的需要

开发过程：检查原型

后期：对最终产品进行评价

2、观察方式：观察可以发现一些意想不到的用户操作方式

观察的两种方式：

(1) 真实环境中的观察：观察者既可作为旁观者，也可作为参与者；重点是应用的上下文

(2) 受控环境中的观察：观察者不能作为参与者，重点是研究用户执行任务的细节

(3) 二者差别不大：有时前者模仿后者的测试条件；实地观察也可作为实验室观察的补充

实验室观察：在专门为可用性测试而安装配置的固定设备的环境下进行的观察，在一些情况下，在实验室中观察是唯一的选择（如空间站等危险环境下的系统）

(1) 实验室布局：测试区、观察区（为避免干扰，二者分开）；用户也可以坐在家中测试（如使用 Windows Live Meeting 等软件，使用户处于更加自然和真实的状态）

(2) 优点：提供了可控且一致的评估环境（易于分析比较）

(3) 缺点：可控且一致的评估环境（人为环境、不自然；可能降低测试结论的普遍性和一般性）；不利于观察多人之间的协作

(4) 具备观察测试过程的能力对于开发人员很重要：令开发人员感到丢脸面；给他们一些启发和指导；使之能够设身处地地为最终用户着想

实验室观察的实际问题：

- (1) 测试地点选择，测试设备安装：摄录面部表情；摄录移动鼠标和击键过程；更广范围内的用户肢体语言
- (2) 测试设备检查：确保设置正确，并能正常工作
- (3) 文档准备：协议书（要求用户签署；说明测试的目的、测试时间，并解释他们的权利；让用户感觉轻松自在）；脚本（要求用户执行的任务）
- (4) 观察框架：观察过程发生的事件都非常复杂且变化迅速，观察框架用于组织观察活动和明确观察重点
 - 1、Goetz and Lecomfte 框架：关注事件的上下文、涉及的人员和技术
人员：有哪些人员在场？他们有何特征？承担什么角色？
行为：人们说了什么？做了什么？举止如何？是否存在规律性的行为？语调和肢体语言如何？
时间：行为何时发生？是否与其他行为相关联？
地点：行为发生于何处？是否受物理条件的影响？
原因：行为为何发生？事件或交互的促成因素是什么？不同的人是否有不同的看法？
方式：行为是如何组织的？受哪些规则或标准的影响？
 - 2、Robson 框架：有助于组织观察和数据搜集活动
空间：物理空间及其布局如何？
行为者：涉及哪些人员？人员详情？
活动：行为者的活动及其原因？
物体：存在哪些实际物体（如家具）？
举止：具体成员的举止如何？
事件：所观察的是不是特定事件的一部分？
目标：行为者希望达到什么目标？
感觉：用户组及个别成员的情绪如何？
- (5) 生理反应监控：决定用户对一个界面的感情反应，提供了客观地获得用户感情状态信息的一种方法
 - 1、心脏活动：是压力或愤怒的反应
 - 2、汗腺活动：表示激励和精神努力的程度
 - 3、大脑活动：与决策的制定、关注和动机有关难点：不清楚这些事件与测量之间的关系，如心率的增加是否意味着在应用界面时遭受到挫折或压力而不能完成任务
- (6) 观察中的问题：
问题：不知道用户在想什么，只能根据观察到的现象去揣测
解决方法 1：让用户“边做边说”，要求被测试人说出自己的想法以及想要做的事情，帮助观察人员了解他们的思考过程，当用户沉默时，观察人员可以提醒用户
解决方法 1 优点：简单、只需要很少的专业技术
解决方法 1 缺点：不自然，可能改变人们执行任务的方式
解决方法 2：合作评估，两位用户共同合作，以便他们互相讨论、相互帮助
解决方法 2 优点：限制少，评估者很容易学会应用；鼓励用户对系统提出批评；在出现不清楚的情况时，评估者能够澄清容易混淆的地方；能提示许多信息（尤其适合评估面向儿童的系统；也适用于多人共享系统的评估）
- (7) 实验室观察评价：
 - 1、能够使得研究人员更好地分离多个可能的影响因素，从而能够得出更准确的研究结果：对具有良好协调性的人机界面所起到的重要作用
 - 2、观察者对被观察者的影响取决于观察类型和观察技巧：可以作为旁观者（只对用户的某些行为感兴趣）或者参与者进行观察

现场观察：指在用户的实际环境中观察用户在使用软件时的情况，是发现同使用环境有关的问题的最佳手段（参考 PPT 中的超市收银系统案例）

- (1) 现场观察问题清单：
 - 1、明确初步的研究目标和问题

- 2、选择一个框架指导观察
 - 3、决定数据记录方式：笔记、录音、摄像，还是三者结合；确保设备到位并能正常工作
 - 4、评估后，尽快与观察者或被观察者共同检查所记录的笔记和其他数据：研究细节，找出含糊之处；人的记忆能力是有限的，最好 24 小时内回顾数据
 - 5、数据记录时，应区分个人意见和观察数据：明确标注需要进一步了解的事项
 - 6、培养良好的合作关系：取得被观察对象的认可和信任；避免只关注某些人，应注意小组的所有成员
 - 7、处理敏感问题：避免让观察对象产生被冒犯的感觉
 - 8、注重团队协作：通过比较不同评估人员的记录，得到更为可信的数据
 - 9、应从不同的角度进行观察，避免只专注于某些特定行为
- (2) 几个难题
- (3) 注意事项：
- 1、观察人员自始至终应尽量保持安静：让用户感觉不到观察人员的存在；保证用户操作和平时工作的状态一样
 - 2、当用户的操作令观察人员无法理解时：需要打断用户，请他对所做的某些操作进行解释；或把用户莫名其妙的操作行为记录下来
 - 3、观察初期，应该拒绝用户的任何帮助请求：希望观察用户在没有系统专家指点的情况下如何操作；待评估完成后为用户提供适当帮助

观察与访谈相结合：

- (1) 观察方法只能展示用户做了什么，而无法知道用户为什么这样做
- (2) 应该在记录数据之后再结合其他方法，如访谈
- (3) 访谈操作方法：请用户详细讲述记录里面任何可能引发可用性问题的地方；如对一个没用过系统某个功能的用户，询问为什么没有使用某项功能
- (4) 让用户面对记录数据时应非常小心：避免让用户产生被监视的想法

3、数据记录：可以根据研究人员的专业素质及环境、项目的特点来选取合适的方法

(1) 纸笔记录：最原始、最廉价

前提：对观察对象有一定的了解，从而有明确的观察侧重点

优点：事后对观察结果进行分析的工作量小

缺点：观察者容易疲劳，而且记录速度有限；建议将记录者和评估者分开

(2) 音视频记录：

适用场合：对于观察对象不太了解，或者是需要观察的内容较多；特别是在采用边做边说法时

音频记录：信息全面，没有任何遗漏，便于事后详细分析；没有可见记录，转录数据非常烦琐；常用于提示重要细节或作为情景说明的辅助材料

视频记录：能够看到参与者正在做的事情；但要始终让参与者停留在视觉的范围内很困难

缺点：所含的信息量很大，所以数据分析非常耗时

(3) 日志和交互记录：

间接观察的适用场合：直接观察可能影响用户；评估人员无法在现场进行研究；可根据搜集到的数据，推断实际情形，并找出可用性和用户体验方面的问题

优点：体现了用户是如何完成真实任务的；使得从工作在不同环境下的大量用户那里自动收集数据变得相当容易（适用于用户分散、无法当面测试的情形，如互联网应用和网站设计项目等）

包含信息：使用软件的频度、使用软件的时间、使用每个操作的频度、哪些操作经常被使用、使用何种交互方式启动各个功能（鼠标 or 键盘）

4、数据分析

(1) 观察数据：

观察得到的数据：1、草图、相片、访谈或事件的录音录像、日志和交互记录等；2、用户在做什么并统计用户花在任务各个部分上的时间；3、用户的情感反应（如叹气、皱眉、耸肩等）

数据类型：用于描述的定性数据（描述观察到的现象）；用于分类的定性数据（使用各种技术进行分类）；定量数据（用于统计目的）

- (2) 定性分析：
分析方法：详细分析通常不必要；粗略分析
常用的分析方法：
- 1、找出关键事件，如用户遇到困难的地方（用户发布评论、保持沉默或表露出迷惑的神情）
 - 2、内容分析（content analysis）：用于“详细分析”录像数据
操作：把数据内容划分为一些有意义、而且互斥的类别（不能以任何方式相互重叠、内容类别也必须可靠，即不同人的分类结果不能存在很大差异）
特点：费时、费力、不常使用
改进：层次化内容分类技术
 - 3、会话分析（conversation analysis）：用于仔细检查语义，重点是对话过程
操作：弄明白发话者想要表达什么，而受话者又是怎样理解它的意思并做出反应
可用于聊天室、虚拟社区等互联网应用，以增进对用户需要的理解
 - 4、话语分析（discourse analysis）：
关注话语的使用，而不是内容
把语言视为反映心理和社会因素的媒介（了解人们如何使用语言），措辞上的微小改动即可改变话语的意思
- (3) 定量分析：视频数据的标注和分析
统计分析：平均值、标准偏差、T 检验
分析工具：
- 1、文本记录分析工具：访谈稿、现场记录、开放性问卷调查等、NUDIST、Ethnograph
 - 2、视频分析工具：Observer VideoPro
 - 3、网络日志分析工具：Webtrends、WebLog

第十二章 评估之询问用户和专家

1、了解用户的需要和对产品的意见和建议：

- (1) 观察用户
- (2) 询问用户
 - 1、适用于客观上较难度量的、与用户主观满意度和可能的忧虑心情相关的问题；
 - 2、访谈和问卷调查：在实地研究中同时进行；在研究用户如何使用系统，以及哪些系统功能是用用户非常喜欢或不喜欢的方面也非常有效
- (3) 不知道该怎么做或者对预期的结果没有把握
请专家帮忙：不能帮助大家成为可用性专家；但有助于更好地去评估自己和他人的工作

2、询问用户之访谈

- (1) 访谈的定义：有目的的对话过程。分为开放式（或非结构化）访谈，结构化访谈，半结构化访谈和集体访谈
- (2) 指导原则：
 - 1、避免过长的问题
 - 2、避免使用复合句，应把它们分解几个独立的问题
 - 3、避免使用可能让用户感觉尴尬的术语或他们无法理解的语言
 - 4、避免使用有诱导性的问题
 - 5、尽可能保证问题是中性的
- (3) 访谈步骤：
 - 1、“开始”阶段：访问人先介绍自己；解释访谈的原因，消除受访人对道德问题的疑虑，询问受访人是否介意被记录（录音或摄像）
 - 2、“热身”阶段：先提出简单的问题
 - 3、主要访谈阶段：按逻辑次序由易到难提问

- 4、“冷却”阶段：提出若干容易的问题，消除用户的紧张感觉
- 5、结束访谈：感谢受访者，关闭录音机，收好笔记本，表面访谈已经结束

(4) 补充建议：

- 1、衣着整洁、朴实。可能的话，与受访人穿着相似的服装
- 2、准备一份供受访者签署的协议书
- 3、若需要使用记录设备，应事先掌握设备的使用方法，确保设备能够工作
- 4、如实记录访谈过程，不要润色、更正或修改受访人的回答

(5) 访谈类型：

1、非结构化访谈：

问题是开放式的，不限定内容和格式；
受访人自行选择详细回答还是简要回答；
访问人应确保能够搜集到重要问题的回答。

2、结构化访谈：

根据预先确定的一组问题进行访谈；
问题通常是“封闭式”的，它要求准确的回答

3、半结构化访谈：

开放式问题+封闭式问题
注意不要暗示答案

4、集体访谈：

基本思想：个别成员的看法是在应用的上下文中通过与其他用户的交流而形成的
“焦点小组”是集体访谈的一种形式

(6) 焦点小组：存在风险，比较省钱而又类似于焦点小组的方法是使用各种形式的计算机回忆和各种电子网络系统

1、非正式的评估方法：

在界面设计之前和经过一段使用之后评估用户的需要和感受；
是市场、政治和社会科学研究经常使用的方法；
人数限制：由大约 6 到 9 个典型用户组成；
如在评估大学的网站时，可考虑由行政人员、教师和学生组成 3 个分别的焦点小组。

2、主持人工作：

事先列出一张讨论问题和数据收集目标的清单；
保持所谈论的内容不离题；
保证小组的每个成员都积极参与谈论；
讨论结果的分析报告

3、询问用户之问卷调查：问卷调查是用于搜集统计数据和用户意见的常用方法，可单独使用，也可与其他技术结合使用

问卷设计原则：

- (1) 应确保问题明确，具体
- (2) 在可能时，采用封闭式问题并提供充分的答案选项
- (3) 对于征求用户意见的问题，应提供一个“无看法”的答案选项
- (4) 注意提问次序，先提出一般化问题，再提出具体问题
- (5) 避免使用复杂的多重问题
- (6) 在使用等级标度时，应设定适当的等级范围，并确保它们不重叠（做到直观、一致）
- (7) 避免使用术语
- (8) 明确说明如何完成问卷（如说明应在选项前的方框内打“√”）
- (9) 在设计问卷时，既要做到紧凑，也应适当留空

问题类型：

- (1) 常规问题：年龄、性别、职业、居住地、应用计算机的经验等
- (2) 自由回答问题：能够提出设计人员没有考虑到的建议

(3) 量化分级问题：要求用户以数值尺度判断一个特定陈述，奇数刻度较偶数刻度更常用

(4) 多选题：对于收集用户以前的经验信息很有用

用户满意度调查表 (QUIS)：使用 9 级标度，已经过了多次实践检验

涉及方面：

- (1) 界面细节（如符号的易读性和屏幕显示的布局设计）
- (2) 界面对象（如具有象征意义的图标）
- (3) 界面行为（如为用户经常使用的操作设置的快捷方式）
- (4) 任务表达（如适当的术语和屏幕显示顺序）

问卷组织：

问卷调查中的两个关键问题：如何寻找有代表性的用户？如何达到合理的回复率？

有助于提高回复率的措施：

- (1) 精心设计问卷，避免用户因为厌烦而拒绝回复
- (2) 参照 QUIS，提供简要描述，说明用户若没有时间完成整份问卷，可以只完成简短的部分
- (3) 提供一个带有回复地址并粘好了邮票的信封
- (4) 解释为什么要进行这些问卷调查，并说明将为参与者保密
- (5) 发出问卷之后，通过后续邮件、电话或电子邮件联系参与者
- (6) 采取一些激励措施（如有偿调查等）
- (7) 进行小规模测验

在线问卷调查：

(1) 能有效而方便地搜集大量人员的意见：

- 1、能够快速搜集调查结果
- 2、与纸张式的问卷调查相比，成本更低，甚至为零
- 3、数据可以立即输入数据库进行分析
- 4、可缩短数据分析的时间
- 5、容易更正问卷中存在的问题
- 6、回复率可能低于纸质问卷

(2) 两种形式：

- 1、基于电子邮件：能够针对特定的用户，但邮件能够容纳的内容有限
- 2、基于网页的调查：形式灵活，并能验证数据的有效性，但调查对象是随机的

(3) 把纸质问卷转变为网页的形式需要经过的四个步骤：

- 1、根据原始的纸质问卷制作无错误的交互式电子问卷
- 2、应确保用户能够从网络的任何位置，使用任何通用的浏览器打开问卷，而且问卷的显示不受屏幕大小的限制
- 3、为避免同一位用户多次提交问卷结果，应记录用户的标识信息，同时也应该注意保密
- 4、在正式进行调查之前，先进行小规模的用户测试

问卷调查与访谈：

(1) 问卷调查或访谈都属于间接方法：

- 1、因为两者都不对用户界面本身进行研究，而只是研究用户对界面的看法
- 2、都不能完全听信和采纳用户的说法

(2) 访谈：形式更自由、难以获得确切数据、需要花费更多时间、可在访谈后立即得到结果、可能回避某些“敏感问题”的真实想法

4、询问专家之认知走查

评估应该贯穿于整个设计过程中，理想情况下，系统所有实现工作开始之前就应该评估，专家分析可应用于项目设计的任何阶段

认知走查：

- (1) 逐步检查使用系统执行任务的过程，从中找出可用性问题
- (2) 无需用户参与
- (3) 认知走查的主要目标是确定使一个系统如何易于学习

- (4) 试图想象出人们在第一次使用某个产品时的想法以及所采取的动作，它的大概流程是怎样的
- (5) 评估的具体过程就是把用户在完成这个功能时所做的所有动作讲述成一个令人可以信服的故事

走查的步骤：

- (1) 标识并记录典型用户的特性：有关用户自身心理、心理特点以及他们的知识和经验的描述；
- (2) 基于评估重点，设计样本任务：应该是大多数用户要做的典型任务；
- (3) 制作界面原型（或界面描述），明确用户执行任务的具体步骤；
- (4) 由设计人员和专家级评估人员（一位或多位）共同进行分析
- (5) 评估人员结合应用的上下文，逐步检查每项任务的操作步骤：检查每项任务的操作步骤时，了解以下问题。认知走查的记录工作非常重要！
 - 1、用户是否想执行产生某种效果的行为？
 - 2、正确的操作对于用户是否足够明显？即用户能否知道如何完成任务（可预见）
 - 3、用户能否注意到正确的操作？功能名称或图标设计是否容易理解（可理解）
 - 4、能否正确解释操作的响应？执行—评估交互周期的完成（可解释）
- (6) 在完成逐步检查之后，汇总关键信息
- (7) 修改设计，更正发现的问题

认知走查分析：

优点：不需要用户参与；不需要可运行的原型；能找出非常具体的用户问题

缺点：工作量大，非常费时；关注面有限（只适合于评估一个产品的易学习性；不太容易发现使用效率方面的可用性问题）

协作走查：由用户、开发人员和可用性专家合作，逐步检查任务场景，讨论与对话元素相关的可用性问题。在评估过程中，每一位专家都承担用户的角色

优点：专注于用户任务；能够产生定量数据；符合参与式设计原则

缺点：需要各方面的专家，速度慢；由于时间限制，通常只能评估有限的场景

5、询问专家之启发式评估：一种灵活而又相当廉价的评估方式

- (1) **Nielsen 的十条启发式规则：**系统状态的可视性；系统应与真实世界相符合；用户的控制权及自主权；一致性和标准化；帮助用户识别、诊断和修复错误；预防错误；依赖识别而非记忆；使用的灵活性及有效性；最小化设计；帮助及文档

- (2) **启发式评估分析：**

优点：

- 1、由于不涉及用户，所以面临的实际限制和道德问题较少
- 2、成本相对较低，不需要特殊设备，而且较为快捷
- 3、又被称为“经济评估法”

缺点：

- 1、评估人员需要经过长时间的训练才能成为专家：理想的专家应同时具备交互设计和产品应用域的知识
- 2、可能出现“虚假警报”：“专家每找到一个真实的可用性问题，将发出约一个假警报（1.2），忽略大约半个问题（0.6）”

参与评估的专家数目可以互不相同：5个评估人员可以发现75%的可用性问题

- (3) **评估步骤**

阶段	步骤
准备（项目指导）	a. 确定可用性准则； b. 确定由3-5个可用性专家组成的评估组； c. 计划地点、日期和每个可用性专家评估的时间； d. 准备或收集材料，让评估者熟悉系统的目标和用户。将用户分析、系统规格说明、用户任务和用例情景等材料分发给评估者； e. 设定评估和记录的策略。是基于个人，还是小组来评估系统？指派一个共同的记录员还是每个人自己记录？
评估（评估者活动）	a. 尝试并建立对系统概况的感知； b. 温习提供的材料以熟悉系统设计。按评估者认为完成用户任务时所需的操作进行实际操作； c. 发现并列出系统中违背可用性原则之处。列出评估注意到的所有问题，包括可能重复之处。确保清楚地描述发现了什么？在何处发现？
结果分析（组内活动）	a. 回顾每个评估者记录的每个问题。确保每个问题能让所有评估者理解； b. 建立一个亲和图（affinity diagram，又称KJ法、A型图解法），把相似的问题分组； c. 根据定义的准则评估并判定每个问题； d. 基于对用户的影响，判断每组问题的严重程度； e. 确定解决问题的建议，确保每个建议基于评估准则和设计原则。
报告汇总	a. 汇总评估组会议的结果。每个问题有一个严重性等级，可用性观点的解释和修改建议； b. 用一个容易阅读和理解的报告格式，列出所有出处、目标、技术、过程和发现。评估者可根据评估原则（heuristics）来组织发现的问题。一定要记录系统或界面的正面特性； c. 确保报告包括了向项目组指导反馈的机制，以了解开发团队是如何使用这些信息的； d. 让项目组的另一个成员审查报告，并由项目领导审定。

6、颜色的使用：

- (1) 使用颜色突出重要信息：
 - 1、在页面上使用一致的颜色；
 - 2、黑色文字在白色或浅灰色背景下，或者棕色文字在白色背景下具有较好的可读性
 - 3、不要为大段文字使用蓝色显示
 - 4、记住还有很多用户对颜色感知存在问题
- (2) 不要使用过多颜色
- (3) 协调配色：单色、互补色、相近色、三等分

第十三章 评估之用户测试

1、用户测试：

- (1) 在受控环境中（类似于实验室环境）测量典型用户执行典型任务的情况
- (2) 目的是获得客观的性能数据，从而评价产品或系统的可用性，如易用性、易学性等
- (3) 最适合对原型和能够运行的系统进行测试
- (4) 可对设计提供重要的反馈
- (5) 在可用性研究中，往往把用户测试和其他技术相结合

2、测试设计：

用户测试须考虑实际限制并做出适当的折衷：

应确保不同参与者的测试条件相同；

应确保评估目标特征具有代表性；

实验可重复，但通常不能得到完全相同的结果

(1) 定义目标和问题

- 1、目标描述了开展一个测试的原因，定义了测试在整个项目中的价值
- 2、目标是对关注点的说明和解答
- 3、举例，对菜单结构的关注：用户在第一次尝试使用时将能选择正确的菜单；用户在少于 5 秒的时间内，能够导航到正确的 3 级菜单

(2) 选择参与者

1、注意事项：

参与者的选择对于任何实验的成功至关重要；

了解用户的特性有助于选择典型用户，要尽可能接近实际用户；
通常也需要平衡性别比例；
至少 4~5 位，5~12 位用户就足够了

2、参与者安排：

各种实验情形的参与者不同；
各种情形的参与者相同；
参与者配对

3、参与者不同：随机指派某个参与者组执行某个实验情形

缺点：要求有足够多的参与者；实验结果可能会受到个别参与者的影响（解决：随机分配 or 预测试）

优点：不存在“顺序效应”，即参与者在执行前一组任务时获得的经验将影响后面的测试任务

4、参与者相同：相同的参与者执行所有实验情形，与前一种方法相比，它只需一半的参与者

缺点：可能产生“顺序效应”

解决方法：如果有两项任务 A 和 B，那么，应让一半的参与者先执行 A，再执行 B，另一半则先执行 B，再执行 A

优点：能够消除个别差异带来的影响；便于比较参与者执行不同实验情形的差异

5、参与者配对：根据用户特性（如技能和性别等），把两位参与者组成一组，再随机地安排他们执行某一种实验情形；适用于参与者无法执行两个实验情形

6、参与者配对：根据用户特性（如技能和性别等），把两位参与者组成一组，再随机地安排他们执行某一种实验情形，适用于参与者无法执行两个实验情形

缺点：实验结果可能会受一些未考虑到的重要变量的影响；如在评估网站的导航性能时，参与者使用互联网的经验将影响实验结果；因此，“使用互联网的经验”即可作为一个配对标准

7、几种安排方法的比较：

参与者安排	优点	缺点
不同参与者	无顺序效应	需要许多参与者；可能受个别参与者的影响（可通过随机编组等方法解决该问题）
相同参与者	能消除各种实验情形下的个体差异	需要均衡处理以避免顺序效应
配对参与者	无顺序效应；能消除个别差异的影响	可能忽略一些重要变量，造成配对不当

(3) 设计测试任务

- 1、测试任务应当与定义的目标相关；
- 2、测试任务通常是简单任务（如查找信息）；
- 3、有时采用较为复杂的任务（如加入在线社团等）；
- 4、任务不能仅限于所要测试的功能，应使用户全面的使用设计的各个区域（如关注搜索功能的可用性，可请求参与者搜索找出产品 X；更好的方法就是请求参与者找出产品 X 并同产品 Y 进行比较）；
- 5、每项任务的时间应介于 5~20 分钟；
- 6、应当以某些合乎逻辑的方法安排任务（开始时，先提出简单问题有助于增强用户的自信心）

(4) 明确测试步骤

- 1、在测试之前，准备好测试进度表和说明，设置好各种设备
- 2、正式测试前应进行小规模测试
- 3、在必要时，评估人员应询问参与者遇到了什么问题
- 4、若用户确实无法完成某些任务，应让他们继续下一项任务
- 5、测试过程应控制在 1 小时之内
- 6、必须分析所有搜集到的数据

(5) 数据搜集

数据搜集特点：

- 1、确定如何度量观测的结果

2、使用的度量类型依赖于所选择的任务

3、定量度量和定性度量

常用的定量度量：

1、完成任务的时间

2、停止使用产品一段时间后，完成任务的时间

3、执行每项任务时的出错次数和错误类型

4、单位时间内的出错次数

5、求助在线帮助或手册的次数

6、用户犯某个特定错误的次数

7、成功完成任务的用户数

3、测试准备：

(1) 建造一个测试计划时间表：

协调参与者的日程计划、小组成员的日程计划及实验室的可使用性；

(2) 在测试过程中编写对应的脚本：

脚本应当包括协调者和参与者交互的每一个方面，也应当包括一些意外事件（如参与者感觉有点灰心丧气，原型出现错误等）；

(3) 安排示范性测试：

1、测试可以在特定实验室里完成，也可以借助简陋的测试设备完成

2、应当使用一个客观的参与者

4、数据分析：

(1) 变量：实验的目的是回答某个问题或测试某个假设，从而揭示两个或更多事件之间的关系。这些“事件”称之为“变量”。

(2) 自变量：

为回答假设问题，需被操作的一个或多个变量；即开始实验之前，已经设置好的变量。

复杂的实验可能包含不止一个自变量；如假设用户的反应速度不仅取决于菜单选项的数目，也取决于菜单中应用的命令选择

(3) 因变量：

能在实验中测量的变量，其值依赖于自变量的变化

(4) 分析方法：

1、定量数据：最常用的描述性统计方法是次数统计

2、定性数据：通常按主题分类

预先定义类别 ^o	描述 ^o	用户 ^o	对应记录中的时间 ^o	备注 ^o
导航的清晰程度 ^o	不能找到信息 ^o	2# ^o	14:23 ^o	用户 2 用了 5 分钟来查找信息，最终放弃了 ^o
		4# ^o	10:58 ^o	用户 4 在正确的页面上，但没有注意到他要找的信息 ^o
	不知道点哪里 ^o	5# ^o	11:16 ^o	用户 5 注意到所有标签看上去都不对，所以他不停地点击所有按钮，直到找到所需信息 ^o
文字密度 ^o	文字太密集了，阻碍了阅读 ^o	4# ^o	6:57 ^o	当用户 4 发现文字篇幅很长时放弃了继续查找 ^o

(5) 总结报告：将测试的结果以书面形式反馈给产品的设计人员，以便于他们对设计进一步的分析和改进

1. 标题页↵
2. 测试环境描述↵
 - 硬件、软件版本、测试场地、测试时间↵
3. 执行概要↵
 - 简要概括测试发现（几页纸）↵
4. 测试描述↵
 - 最终版的测试计划、方法、培训和任务↵
5. 测试用户数据↵
 - 以表格形式描述用户的年龄、职业、经历↵
6. 结果↵
 - 以图表形式描述花费的时间、出错次数、问卷反馈等↵
 - 讨论和分析，适当引用用户言论↵
7. 正面反馈列表↵
8. 针对发现问题的建议列表，按照问题的严重等级和修复的难易程度降序排列。其中，每条建议内容包括：↵
 - 诊断问题出现的原因↵
 - 给出相应屏幕截图↵
 - 给出严重程度等级↵
 - 准确指出遇到该问题的用户数量↵
 - 给出相应视频记录的时间戳↵
 - 可能的话引用用户的原话↵
 - 给出改进建议↵
9. 附录（原始数据和表格）↵
 - 背景问卷、协议书、测试脚本、数据收集表格、音视频记录、手工笔记等↵