目录

[一、 第一章 概述 3](#_Toc26513)

[(1) 交互设计 3](#_Toc24241)

[(2) 流行术语 3](#_Toc11737)

[(3) HCI 3](#_Toc8381)

[(4) UI（人与计算机发生交互的一个空间） 3](#_Toc27949)

[(5) GUI 4](#_Toc1526)

[(6) HCI和SE 4](#_Toc14590)

[(7) Usability和UX 4](#_Toc178)

[(8) User-Centered Design 5](#_Toc15458)

[二、 第三章 用户分析和任务分析 5](#_Toc29016)

[(1) 研究方法 5](#_Toc30991)

[(2) 用户模型 9](#_Toc4627)

[(3) 场景和设计需求 12](#_Toc2753)

[(4) 以用户为中心的设计（user-centered design） 13](#_Toc6393)

[(5) Analysis（分析阶段） 13](#_Toc31548)

[三、 第四章 交互设计心理学 14](#_Toc3233)

[(1) Human Information Processing 人体信息处理 14](#_Toc21268)

[(2) Attention 注意 15](#_Toc26178)

[(3) Cognitive Processing 认知处理 16](#_Toc13779)

[(4) Motor Processing 动作处理 17](#_Toc13763)

[四、 第五章 可视化设计 18](#_Toc3130)

[(1) 关键词 18](#_Toc24431)

[(2) Visual Interface Design Factors 视觉界面设计因素 18](#_Toc5556)

[(3) Visual Interface Design Principles 视觉界面设计原则 19](#_Toc2089)

[五、 第六章 交互设计 26](#_Toc16946)

[(1) Design Framework 设计框架 26](#_Toc11342)

[(2) Design Techniques and Rules 设计技术与规则 34](#_Toc5)

[(3) Design Details 设计细节 43](#_Toc32431)

[六、 第七章 评价技术 44](#_Toc17781)

[(1) 关键词 44](#_Toc22595)

[(2) Heuristic 启发式 44](#_Toc7206)

[(1) Heuristic Evaluation 启发式评估 48](#_Toc1668)

[(2) How to Write a Usability Aspect Report (UAR) 如何编写可用性方面报告（UAR） 50](#_Toc25737)

[(3) Basic Issues in Think-Aloud Testing 有声思维测试的基本问题 50](#_Toc30096)

[(4) How to Conduct a Think-Aloud Usability Test 如何进行自言自语可用性测试 53](#_Toc31207)

[(5) Think-Aloud Testing vs. Heuristic Evaluation 有声思维测试与启发式评价 55](#_Toc23691)

# 第一章 概述

## 交互设计

数字产品、环境、系统和服务的交互设计实践。

是关于外形和形式的设计，然而最重要的是，它是“行为”的设计。

## 流行术语

**IA**（信息架构），主要是关于网站。

**UX**（User Experience），用户体验。包括**Form**（形式）、**Content**（内容）、**Behavior** （行为）

**IxD**（交互设计）

**HCI**（Human-Computer Interaction）人机交互

**GUI**（图形用户界面）

**Usability**（可用性）

**User Interface**（人与计算机发生交互的空间）

## HCI

**HCI的目标**：

解决技术设计和使用中的实际问题，使基于计算机的系统更容易使用，对人员和组织更有效。

## UI（人与计算机发生交互的一个空间）

**为什么需要UI：**

1. 方便计算机的操作与控制；
2. 人们在操作决策的时候，需要得到计算机的反馈。

**UI分类：**

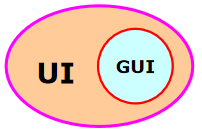
1.Command line interfaces

2.Graphical user interfaces

3.Touch user interface

4.Voice user interfaces

## GUI

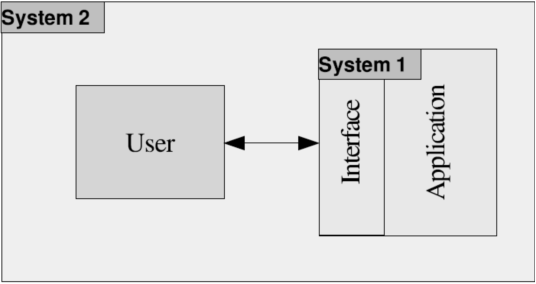
* **定义：**GUI是一个图形的而非纯文本的用户界面。
* **GUI与UI：**。
* **Metaphors（隐喻）：**依赖于用户在界面的视觉提示与软件功能之间建立的直觉 联系，从而不必了解软件的运行机制。

例如：工具栏中的小剪刀——剪切

文件夹——移动、打开。

* GUI中行为通常通过**直接操作（direct manipulation）**图形元素的方式来实 现。

## HCI和SE



**System1:HCI-UCD**

**System2:SE**

## Usability和UX（user experience）

* 界面设计的不好，通过**用户自定义不可行**的原因：

（1）大多数用户不会自定义界面。

（2）用户自定义只会使情况更糟。

* **Usability定义**：用来衡量用户怎样能很好的使用系统功能。

**usability只是系统的一个属性**

* **Usability五个方面：**

**Learnability：**可学习性

**Efficiency：**效率

**Memorability：**可记忆性

**Errors：**出错率

**Satisfaction：**主观满意程度

* **UX定义：**

它与人们使用产品、系统或者服务的感受密切相关。

它高度关注产品自身拥有的和人机交互的经验、情感、意义、和价值等方面。

它本身包含着人们对产品本身的实用性、可用性和效率方面的认知。

## User-Centered Design

* **两个核心：**

1、以用户为中心的设计——体现以人为本；

2、关注可用性——体现简单但是合理。

* 什么是**以用户为中心的设计**：

1、好的UI设计就是恰好能符合用户心里所想

2、检验是不是好的方法——评测

# 第二章 交互设计过程

## Basic Activities and Key Features 基础活动和关键特征

* 什么是设计？在一定限制中达成目标
* 关注三点：Goals、Constraints、Trade-off（权衡）

1. 目标：

我们设计的目标是什么？

为什么他们需要这个

1. 限制

我们必须使用什么元素？

我们必须适应什么标准？

开销？

我们需要花费多少时间？

他们是安全的问题吗？

1. 权衡

* 什么是交互设计

定义了交互系统的结构和行为

## 设计过程中的问题

* 如何选择用户

– 涉众（stakeholders）不等于用户

– 客户可能不是用户

– 用户应该分类（Primary Secondary Tertiary Facilitating）

* 如何识别需求

（1）.识别出用户

（2）.观察用户

（3）.配合想象力提出需求

a.谁实际使用这个系统

b.二谁不直接使用系统，而是接收系统的输出或向系统提供输入

c.谁不属于前两类，但谁直接受到系统成功或失败的影响促进谁负责系统的设计、开发和维护

* 如何提出替代设计方案

（1）.考虑其他类似的设计考虑

（2）.设计约束

* 4. 如何选择合适的设计方案

（1）.根据设计决策

（2）.根据外部特性

（3）.根据内部特性

（4）.考虑用户的体验、偏好和改进建议考虑“质量”

## 交互设计生命周期

* **传统软件生命周期模型**

传统的软件开发是一个瀑布型的文档驱动流程，各个阶段必须逐个进行，包括需求分析、分析设计、编码实现、集成测试、系统测试。而迭代设计能够很好的解决传统软件开发中出现的问题。迭代设计将整个软件开发分成多个阶段性，并且进行阶段性评估，更易于完成和达到目标。

1. 瀑布（Waterfall Process）

2. 螺旋（Spiral Process）

3. 快速应用程序开发Rapid application development（RAD）

4. 原型（Prototyping）

* **交互式设计生命周期模型Star Life Cycle Model**

交互设计的过程与软件设计相像，都需要获取需求，分析，设计，实现，不同的是，交互设计通过原型迭代，软件设计通过产品迭代。

例如：

**恒星生命周期模型**

* **交互设计过程中的一些专业人员**

1. Interaction Designers

2. Usability engineers

3. Web designers

4. Information architects

5. User experience designers

# 第三章 用户分析和任务分析

## 研究方法

* **分类：市场**（Marketing）、**用户**（User）、**可视化**（Visual）
* **定量**研究（Quantitative Research）与**定性**研究（Qualitative Research）

**定量研究**只能回答沿着几个简化轴“how much”或“how many”的问题；

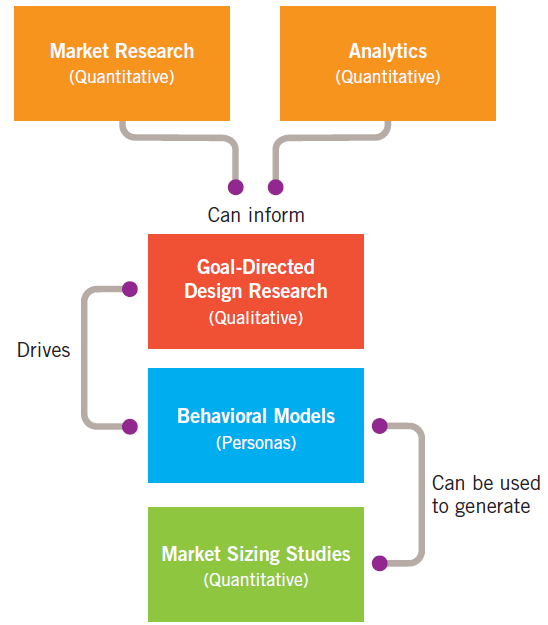
**定性研究**可以告诉你什么，怎么做，以及为什么要用丰富的细节来反映真实的人 类情况的复杂性。

* **定性的研究**对于帮助我们理解产品的领域、情景和受约束的方式非常有价值。
* **定性的研究**也有助于设计项目的进展：

1. 为设计团队提供**可信度**和**权威性**；
2. 对领域问题和用户关注点的共同理解来**团结团队**
3. **授权（Empowering）**管理层对产品设计问题做出更明智的**决策**，否则这些 决策将基于猜测或个人偏好。

* **定性**研究的**价值：实现更快、成本更低**
* **定量**方法的**优点**和**局限性：定量研究**可以帮助指导设计研究

**用户研究**可以为市场研究提供信息



* **Goal-Directed Design Research（目标导向设计研究）**

1. **Kickoff meeting（启动会议）**

提出最初的关键问题

1. **Literature review（文献综述）**

内部文件

行业报告

网络搜索

1. **Product/prototype and competitive audits（产品原型和竞争者审核）**

在访谈利益相关者和主题专家之前或者同时，检查产品的现有版本或者 原型，以及主要竞争对手，对产品设计团队大有帮助。

1. **Stakeholder interviews（利益相关者访谈）**

从利益相关这那里搜集如下信息尤为重要：

产品初期设想

预算和日程计划

技术限制和机遇

商业驱动

利益相关者对用户的看法

1. **Subject matter expert (SME) interviews（主题专家访谈）**

在设计项目初期，找机会并会见极为主题专家（所在领域的权威人士）。

借助主题专家应该考虑下面几点：

主题专家通常是专业用户。

主题专家知识渊博，但不是设计师。

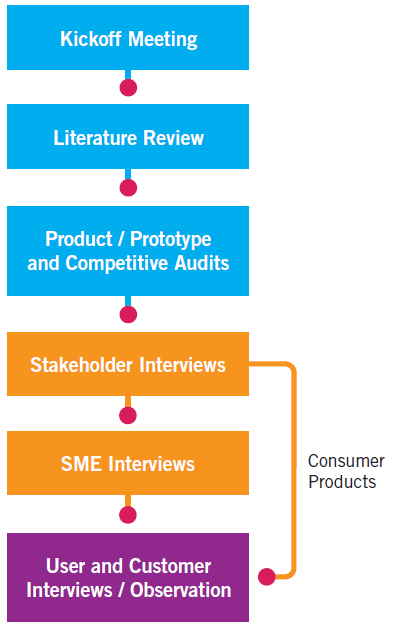
主题专家在复杂或者专业领域必不可少。

确保设计过程中能够得到主题专家的帮助。

1. **User and customer interviews（用户和客户访谈）**

用户和客户这两个概念容易混淆

**（7）User observation/ethnographic field studies（用户观察/人种学实地研究）**



* **Interviewing and Observing Users（访谈并观察用户）**

1. **Contextual Inquiry（情境调查）**

四个基本原则：

1.Context上下文

2.Partnership伙伴关系

3.Interpretation解释

4.Focus聚焦

1. **Improving on Contextual Inquiry（改进情境调查**）

1.Shorten the interview process（缩短访谈过程）

2.Use smaller design teams（适用小规模设计团队）

3.Identify goals first（首先找出用户目标）

4.Look beyond business contexts（超越商业情境）

1. **Preparing for Ethnographic Interviews（为人种学访谈做准备）**

1.理解与个体产品相关的交互行为和习惯。

2.Identifying candidates（确定候选人）

3.The persona hypothesis（人物模型假设）

4.Roles in business and consumer domains（商业和消费领域的角色）

5.Behavioral and demographic variables（行为和人口统计学变量）

6.Domain expertise versus technical expertise（技术专业知识和行业专业 知识）

7.Environmental considerations（环境因素）

8.Putting together a plan（做好计划）

1. **Conducting Ethnographic Interviews（进行人种学访谈）**
2. 构造了**人物模型假设**，并拿出**访谈计划**后就可以进行**人种学访谈**
3. 访谈分为**早期、中期**和**晚期**三个阶段
4. 访谈**基本方法**：

* Interview where the interaction happens（在交互发生的地方进行访谈）
* Avoid a fixed set of questions（避免一系列固定的问题）
* Assume the role of an apprentice, not an expert（假装成门外汉而非专家）
* Use open-ended and closed-ended questions to direct the discussion（采取开放式和封闭式相结合引导提问）
* Focus on goals first and tasks second（首先关注目标，任务其次）
* Avoid making the user a designer（避免把用户当成设计师）
* Avoid discussing technology（避免讨论技术问题）
* Encourage storytelling（鼓励讲故事）
* Ask for a show-and-tell（请求演示和讲解）
* Avoid leading questions（避免诱导性问题）

**Other Types of Qualitative Research（定性研究的其他模型）**

Focus groups（焦点小组）

Usability testing（可用性测试）

Card sorting（卡片分类）

Task analysis（任务分析）

## 用户模型

1. **创建关于用户的描述性模型**，它是交互设计中特有的强有力的工具-**人物模型persona。**
2. **为用户而进行设计的（UCD）**，因此重要的一点就是，要了解下面这些方面并将其视觉化：
3. 用户之间的关系
4. 用户的期望
5. 用户与社会及物理环境之间的关系
6. 用户与我们所设计的产品之间的关系
7. The **Power** of Personas
8. 为具有特定需要的特定个体类型设计
9. 选择正确的设计对象，即代表最广大关键人群需要的用户
10. 任务模型为不同类型用户及其不同的需求提供强有力的交流手段
11. Strengths of **personas** as a design tool（设计工具persona的优势）
12. 确定产品的功能和行为
13. 同利益相关者、开发人员和其他设计师交流
14. 就设计意见达成共识
15. 衡量设计效率：用人物模型对设计方案进行测试
16. 助力市场营销、销售规划和产品相关的其他工作
17. 避免下面三个**设计陷阱**：
18. The elastic user 弹性用户
19. Self-referential design 自我参考设计
20. Edge cases 边缘功能设计
21. 人物模型很**有效**：
22. 首先人物模型是用户模型，能够代表具体个人。
23. 人物模型把设计和开发团队的同理心聚集在用户目标周围。
24. 同理心对设计师来说很关键，人物模型不仅使得设计方案能够更好地服务于真实用户，而且是设计方案更吸引利益相关者。
25. Personas are **based on research**（人物模型以研究为基础）
26. Personas **represent** types of users of a **specific product**（人物模型代表特定交互产品的某一类用户群体）

有时候也称为**合成用户原型（composite user archetypes）**

1. Personas used **across multiple products**（跨产品的人物模型）
2. **Archetypes** vs. **Stereotypes**（模型化形象）

模型化形象往往是人物模型的反面，它通常是设计者或者产品团队偏见和意向的产物。

1. Personas **explore ranges of behavior**（人物模型拓展了用户行为的范围）
2. Personas have **motivations**（人物模型有动机）
3. Personas can **represent relevant nonusers**（人物模型可以代表用户之外的相关人士）
4. Personas are **more appropriate** design tools than other user models（人物模型是比其他用户模型更加合适的设计工具）
5. User roles 用户角色

对一类用户及其问题之间关系的**定义**，包括**需求、兴趣、期望**和**行为模式**；

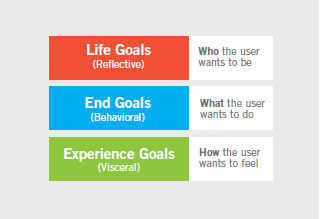
**局限性**体现在：

1.抽象地、与拥有这些行为和关系的人隔离开来，更难以清晰地传达人 类的行为和关系。

2.他们几乎完全专注于任务，而忽略了目标作为设计思维和综合的组织 原则的使用。

3.它们很难作为开发、交流和度量设计决策的连贯的工具。

1. Personas versus user profiles 用户信息
2. Personas versus market segments 市场划分
3. **Understanding Goals 理解目标**
4. **The three types of user goals 用户目标有三种类型：**

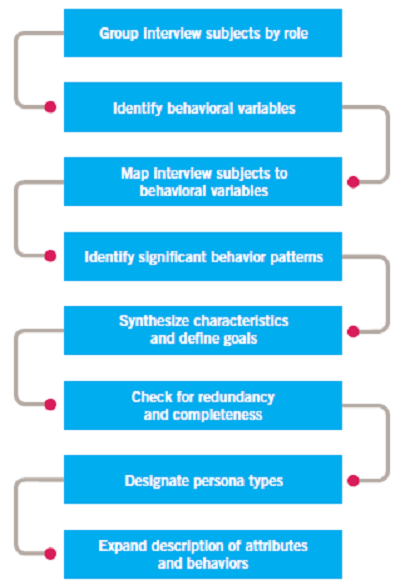


人生目标 反思

最终目标 行为

体验目标 本能

1. 用户目标是用户动机（motivations）
2. **Nonuser goals（非用户目标）**
3. Customer goals 客户目标
4. Business goals 商业目标
5. Nonprofit organization goals 技术目标
6. 如何**构建人物模型**



1根据角色对方谈对象分组 Group interview subjects by role

2找出行为变量 Identify behavioral variables

3将访谈主体和行为变量对应起来 Map interview subjects to behavioral variables

4找出重要的行为模型 Identify significant behavior patterns

5综合各种特征，阐明目标 Synthesize characteristics and define goals

6检查完整性和冗余 Check for completeness and redundancy

7指定人物模型类型 Designate persona types

人物模型的**类型**：主要 Primary

次要Secondary

补充Supplemental

客户人物模型Customer

接受服务的人物模型Served

负面人物模型Negative

8进一步描述特征和行为 Expand description of attributes and behaviors

1. **Other Design Models**

Work flow models 工作流

Artifact models 人工制品

Physical models 物理模型

## 场景和设计需求

1. **四个主要活动**
2. 利用故事情节或者场景剧本来**设想**理想用户的**交互过程**
3. 运用场景剧本来**提取设计需求**
4. 依次运用这些需求来**定义**产品的**基本交互需要**
5. 在这个框架中不断**增加设计细节**
6. **Scenarios: Narrative as a Design Tool （场景，以叙述为设计工具）**
7. **Scenarios vs. use cases & user stories（场景设计**和**使用案例**以及**用户故事）**

场景和使用案例都是用来描述用户和系统的交互方法

1. **Goal-Directed Scenarios**（**目标导向的场景**）

从具体用户（人物模型）的角度定义产品行为的迭代手段

1. **Use cases**（**使用案例**）

通常是一种技术手段，基于对系统功能需求的全面描述上，具有事务性，关注底层 用户行为和系统的反映。

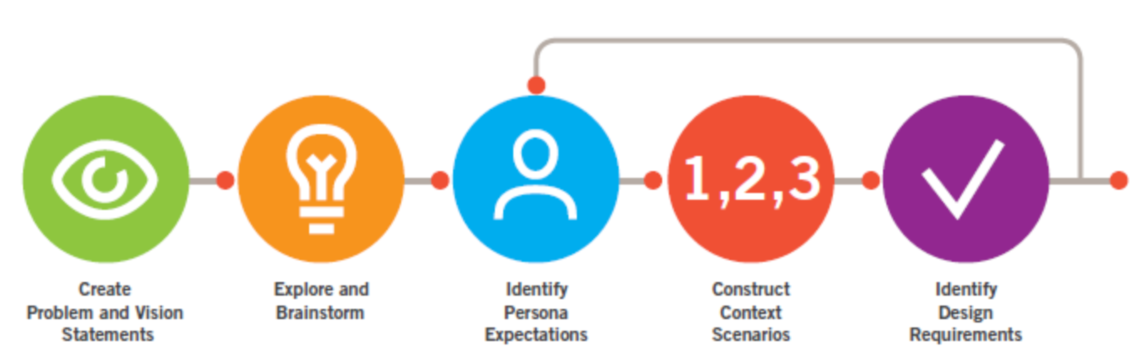
1. **User stories**（**用户故事**）

通常用简短的词句组成，简单描述完成这一交互所需要的界面，不是场景，没有整 个用户流程和用户的最终目标。

1. **Scenario-based design**（基于场景的设计）

缺少的一环就是人物模型

1. **Persona-based scenarios**（**基于人物模型的场景**）
2. **Three types of scenarios（三种场景）**
3. Context scenario 情境场景
4. Key path scenario 关键路径的情况
5. Validation scenarios 验证场景
6. **Design Requirements**: The “What”of Interaction（**设计需求**：交互的什么问 题）
7. 在设计产品的功能之前，先**定义**产品的**功能**。
8. 设计需求**不是特性**
9. 设计需求**不是规范**
10. 设计需求是**战略性**的
11. 设计需求来**自多个来源**
12. **The** **Requirements Definition Process**（**需求定义**的过程）



1. Create problem and vision statements **创建问题和愿景陈述**
2. 问题和远景陈述的内容应该直接来自研究和用户模型。
3. 用户目标和需求应该来自主要和次要角色。
4. 业务目标应该从涉众访谈中提取。
5. Explore and brainstorm **探索和头脑风暴**
6. Identify persona expectations **确定人物模型的期望**

对于每个主要和次要角色，我们确定如下:

1. 影响角色期望的态度、经历、愿望以及其他社会、文化、环境和认知 因素
2. 角色对使用产品的体验的一般期望和愿望
3. 角色将期望或希望从产品中获得的行为
4. 该角色如何考虑基本元素或数据单元
5. Construct context scenarios **构建情景场景**
6. Identify design requirements **明确设计需求**

separate requirements **独立需求：**

1. Data requirements 数据需求
2. Functional requirements 功能需求
3. Contextual requirements 上下文的需求
4. Other requirements 其他需求

## 以用户为中心的设计（user-centered design）

1. **定义**：以用户为中心的设计就是在设计阶段，最终用户在需要，渴望，以及限制等 方面给以优先权的过程。

## Analysis（分析阶段）

1. **分类：**
2. **User analysis 用户分析**

搜集与用户相关的信息，用户是谁，他们的目的是什么，他们需要完成什么任 务

1. **Task analysis 任务分析**

用户需要操作的每一个步骤的特征

创建实际使用的具体场景

决定支持的用户和任务

1. 为什么**进行用户分析是很明白**的：
2. 因为你不是用户，你需要确定谁才是真正的用户。
3. 不仅仅要知道他的个性，还要了解他的职业，他在什么环境下使用你的软件，有什么会分散他的注意力，他的社会与背景，使用环境（电影院、图书馆、车里、飞机上）：你所设计的用户界面要根据环境的不同有不同的约束。
4. **Multiple Classes of Users（多类用户）**
5. **How to Do User Analysis（怎样进行用户分析）**
6. Questionnaires 调查问卷
7. Interviews 直接面谈
8. Observation 观察
9. **Task Analysis （任务分析）**

用户分析的下一步就是找出与任务相关的问题。

一个任务可以理解为一个目标：即需要做什么？怎么做？

开始任务分析的好方法就是**逐级分解任务**。

1. **Essential Parts of Task Analysis（任务分析的重要组成部分）**

应该做什么？应该先做什么？这个任务分几个步骤？

**目标**就是任务的名称。例如发送邮件信息。

**先决条件：**为了实现这个任务必须满足的条件。

另一个设计解决方案就是**为完成先决条件提供机会**。

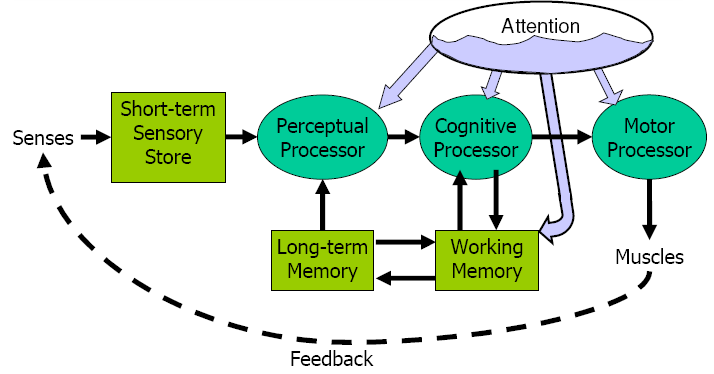
最后就是**将任务分解成子任务**。如果子任务仍然很大就需要进一步分解。

# 第四章 交互设计心理学

## Human Information Processing 人体信息处理

1. **How to process information?**

Human Information Processing Model：



1. **Memory properties 内存属性**

**Encoding （编码）**: type of things stored

**Size（大小）**: number of things stored

**Decay time（衰退时间）**: how long memory lasts

1. **Short-Term Sensory Store 短期感官储存**

Visual information store视觉信息存储

Auditory information store 听觉信息存储

1. **Working Memory (WM) 工作记忆**

Small capacity: 7±2 “chunks” **小容量**

Fast decay (7 [5-226] sec) **快速衰减**

Maintenance rehearsal fends off decay **维修排练，防止腐烂**

Interference causes faster decay **干扰会导致更快的衰减**

1. **Long-term Memory (LTM) 长期记忆**

Huge capacity **巨大的容量**

Little decay **很少腐烂**

Elaborative rehearsal moves chunks from WM to LTM by making connections with other chunks **精心排练通过与其他语块建立联系，将语块从WM移到LTM**

1. **Processors 处理器**
2. **Perceptual Fusion知觉融合**
3. **Bottom-up vs. Top-Down Perception自下而上与自上而下的感知**

**自下而上**使用刺激的特征

**自上而下**使用上下文

时间、空间

利用长期记忆

1. **Chunking 分块**

unit of perception or memory 感知或记忆单位

## Attention 注意

1. **Photoreceptor 光受体**
2. Rods（视杆）

Only one kind (peak response in green wavelengths) 只有一种（绿色波长的 峰值响应）

Sensitive to low light (“scotopic vision”) 对微光敏感（“暗视视觉”）

Saturated at moderate light intensity (“photopic vision”) 对微光敏感（“暗视 视觉”）

1. Cones（视锥）
2. **Signals from Photoreceptors 光感受器发出的信号**
3. Brightness 亮度

M + L + rods M+L+杆

1. Red-green difference 红绿差

L - M L-M

1. Blue-yellow difference 蓝黄色差

Weighted sum of S, M, L S，M，L的加权和

1. **Color Blindness 色盲**

**Red-green** color blindness **红绿**

**Blue-yellow** color blindness **蓝黄**

**Guideline**: 不要仅仅依赖于颜色的区别，使用冗余信号：亮度、位置、形状

1. **Chromatic Aberration（色差）**

不同的波长聚焦不同。

**Guideline ：**不要在蓝色文本上使用红色，它看起来很模糊，读起来很痛

1. **Blue Details Are Hard to Resolve 蓝色的细节很难解决**

**Guideline ：**不要在小细节很重要的深色背景下使用蓝色（文本！）

1. **Fovea Has No Rods 中央凹没有杆**

如果你不直视一颗暗淡的星星，就更容易看到它

1. **What is attention?**

注意力是一个过程，通过这个过程，大脑在任何给定的时刻从各种刺激中进行 选择。

1. **Spotlight metaphor聚光灯隐喻**

Visual dominance视觉优势：更容易关注视觉通道而不是听觉通道

## Cognitive Processing 认知处理

1. **Cognitive processor 认知处理器**

Compares stimuli 比较刺激

Selects a response 选择响应

1. **Types of decision making 决策类型**

Skill-based 基于技能

Rule-based 基于规则

Knowledge-based 基于知识

1. **Hick-Hyman Law of Choice Reaction Time 希克-海曼选择反应时定律**

Reaction time depends on information content of stimulus

（反应时间取决于刺激的信息含量）：

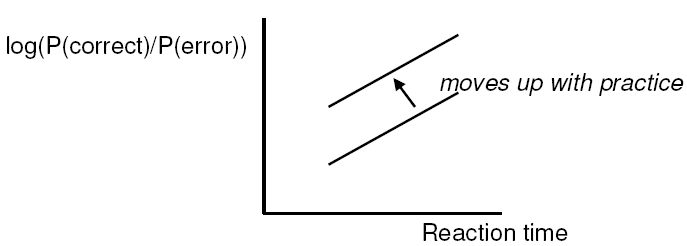
RT = c + d log2 1/Pr(stimulus)

1. **Speed-Accuracy Tradeoff 速度精度折衷**

准确度随反应时间而变化

可以选择曲线上的任意点

可以练习移动曲线



1. **Divided Attention (Multitasking) 分散注意力（多任务）**
2. Resource metaphor 资源隐喻

注意力是一种可以同时分配给不同任务的资源

1. Multitasking performance depends on: 多任务性能

Task structure 任务结构

Modality 形态: visual vs. Auditory 视觉与听觉

Encoding 编码: spatial vs. Verbal 空间与语言

Component 成分: perceptual/cognitive vs. motor vs. WM

知觉/认知vs.运动vs.WM

Difficulty 困难程度

简单或练习良好的任务更容易分享

## Motor Processing 动作处理

1. **Two ways of motor processor operate 电机处理器的两种操作方式**

Open-loop control 开环控制

电机处理器自己运行一个程序

循环时间为Tm~70ms

Closed-loop control 闭环控制

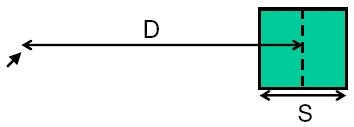
肌肉运动（或其对世界的影响）被感知并与期望的结果进行比较

循环时间为Tp+Tc+Tm~240ms

1. **Fitts’s Law菲茨定律**

将手移动到距离D处尺寸为S的目标的时间T为：

T=RT+MT=a+b对数（2D/S）



仅取决于难度日志索引（2D/S）

1. **Implications of Fitts’s Law 菲茨定律的含义**

屏幕边缘的目标很容易被击中

Mac菜单栏胜过Windows菜单栏

不可刮边是愚蠢的

层次菜单很难找到

Gimp/GTK：立即关闭菜单

Windows:0.5s超时破坏因果关系

线性弹出菜单与饼图菜单

1. **Power Law of Practice 实践幂律**

完成任务的时间第n次是：

Tn=T1 n-α

α 通常为0.2-0.6

# 第五章 可视化设计

## 关键词

Alignment（对齐）

Associativity（关联性）

Consistency（一致性）

Reduction（减少）

Regularity（规律）

Proximity（接近性）

Selectivity（选择性）

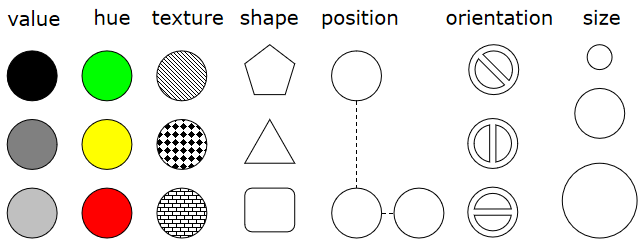
Simplicity（简单）

## Visual Interface Design Factors 视觉界面设计因素

1. **Contrast & Visual Variables 对比度和视觉变量**

Highlight differences 突出差异

视觉变量——1967年



**深浅 色调 材质 形状 位置 方向 大小**

**Temperature 温度**：position、hue、shape

1. **Characteristics of Visual Variables 视觉变量的特征**

**Quantitative 定量**：这个变量的变化是否有一个数值读数？

Position、size

**Order 顺序**：这个变量的变化是有序的吗？

Size、value、color、shape

**Length 长度**：在这个变量中有多少变化是可以察觉的？

Shape、orientation

1. **Attention 注意力**

回想一下聚光灯的比喻

注意力焦点从一个输入通道连续移动到另一个输入通道

聚光通道内的所有刺激都是并行处理的

输入通道=一个或多个可视变量

例如： 位置、色调

1. **Selectivity 选择性**

Selective perception 选择性感知：

是否可以将注意力集中在变量的一个值上，排除其他变量和值？

Selective选择性：位置、大小、方向、色调、值、纹理

Not selective非选择性：形状

1. **Associativity** **关联性**

Associative perception 关联感知：

在查看其他变量时可以忽略变量吗？

Associative关联：位置、色调、纹理、形状、方向

Not associative非关联：大小、值

小尺寸和低值干扰感知色调、值、纹理和形状的能力

## Visual Interface Design Principles 视觉界面设计原则

1. **Guidelines for Good Graphic Design 优秀平面设计指南**

Simplicity（简单）

Contrast（对比）

White space（留白）

Alignment （对齐）

Grouping and Structure（分组与结构）

1. **Simplicity 简单**
2. “完美不是在没有什么可加的时候实现的，而是在没有什么可带走的时候。”

“简单并不意味着没有任何装饰…这只意味着装饰应该与设计本身密切相关， 任何与它无关的东西都应该被拿走。”

“简单点，笨蛋。”

“少就是多。”

“如果有疑问，将其删除。”

1. 简单化技巧：

Reduction 简化： 删除不必要的元素

Regularity 规律性：布局整齐

Double-Duty 双重职责：组合元素以实现杠杆效应

找到一个元素扮演多个角色的方法

Find a way for one element to play multiple roles

如：滚动条 既显示位置，又能移动

1. **Contrast 对比**
2. 选择适当的视觉变量 Choose appropriate visual variables.

使用尽可能多的长度 Use as much length as possible.

锐化区别，更容易感知

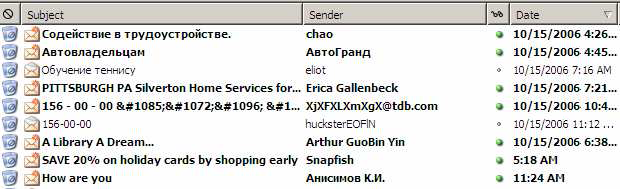
乘法标度，非加法

必要时进行冗余编码

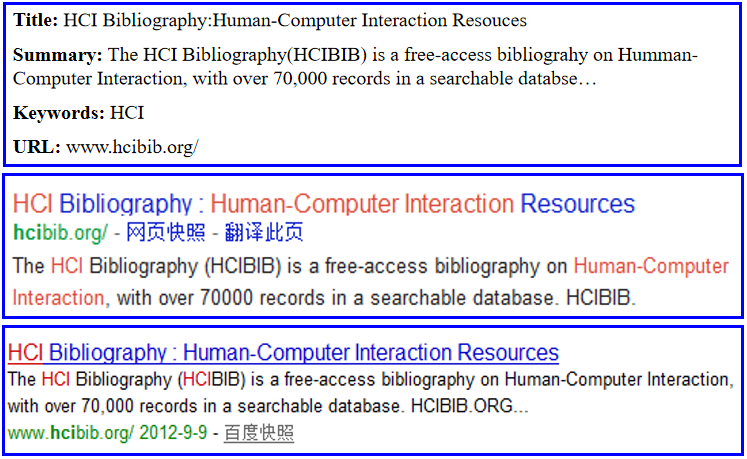
需要的卡通夸张

使用“斜视测试” Use the "squint test"

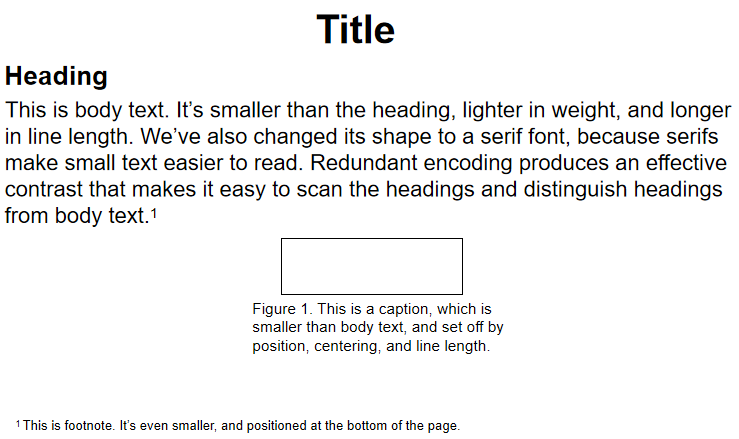
1. Choosing Visual Variables for a Display 为显示选择视觉变量



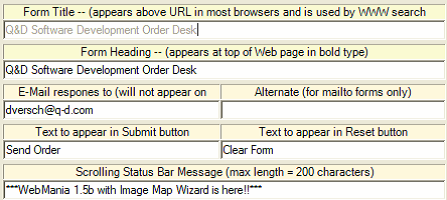
1. Designing Information Displays 设计信息显示



1. Contrast in Publication Styles出版物风格对比



1. Contrast Problems 对比问题



1. **White Space 留白**
2. 分组时使用空格，而不是行

使用边距在设计周围画眼睛

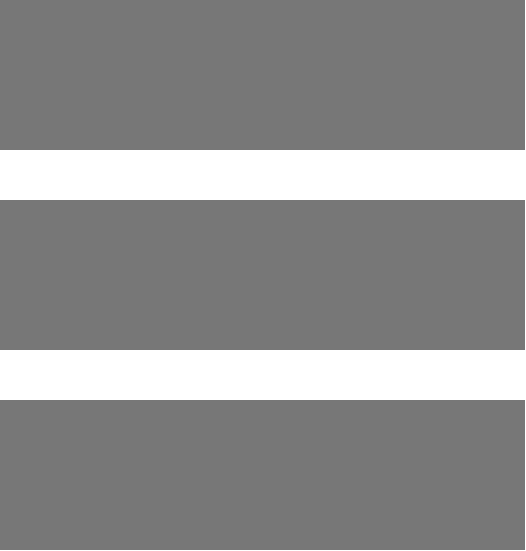
整合图形和背景

对象应与其背景成比例缩放

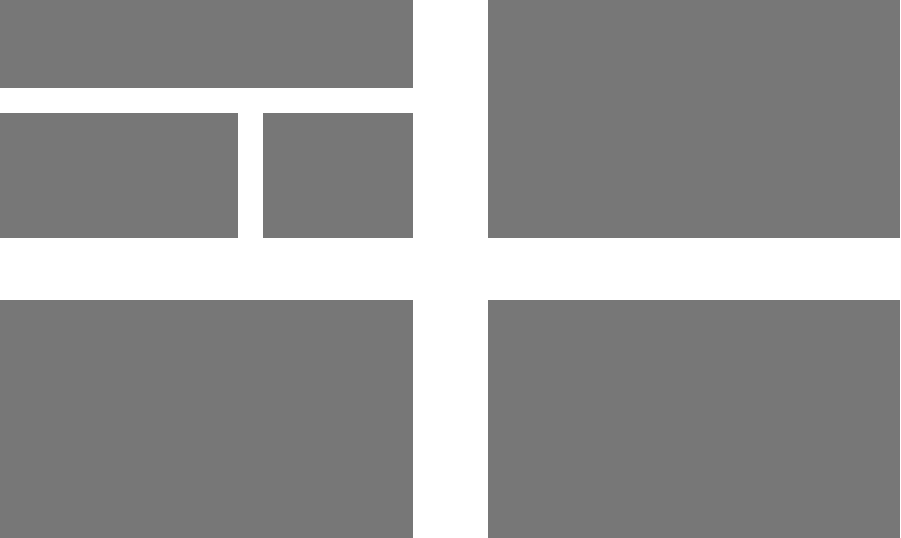
不要把控件挤在一起

拥挤会产生空间张力并抑制扫描

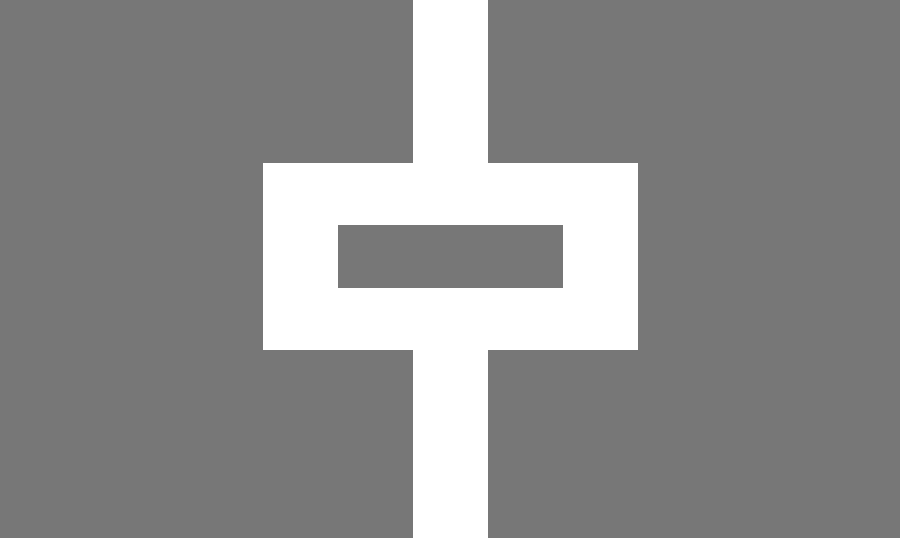
1. White Space - Space to Separate 空白-要分隔的空间



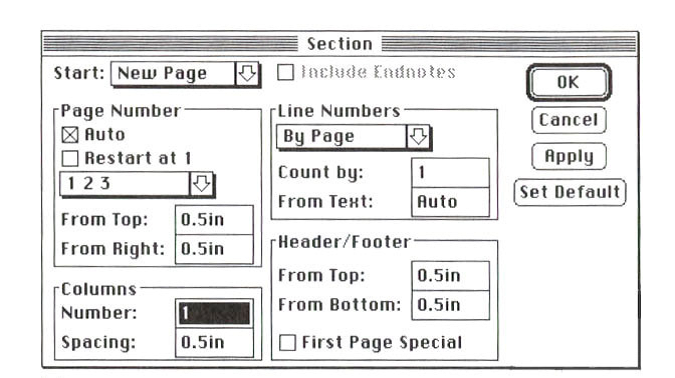
1. White Space - Space to Structure 空白-空间到结构

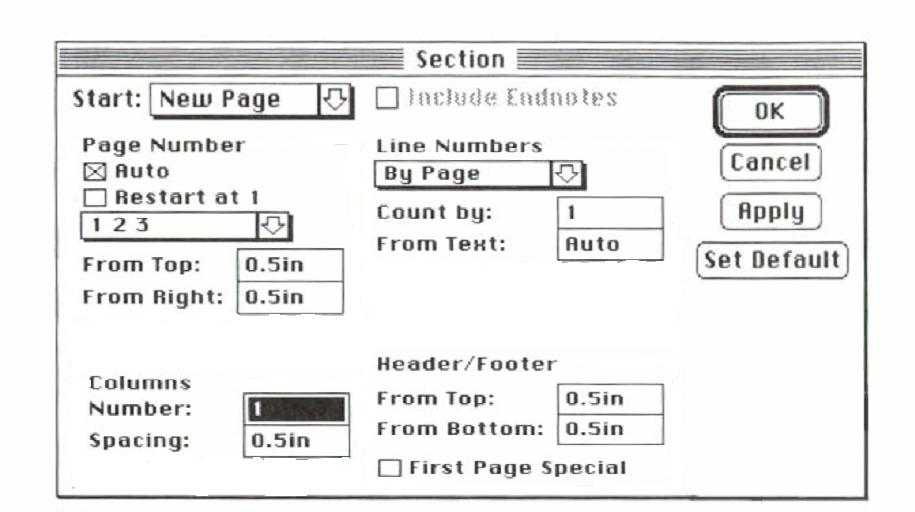


1. White Space - Space to Highlight 空白-要突出显示的空白



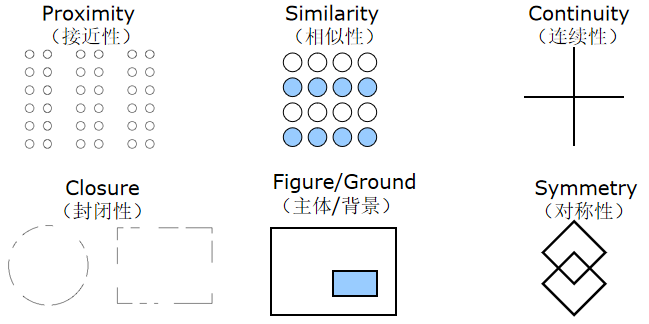
1. **Crowded Dialog 拥挤的对话**





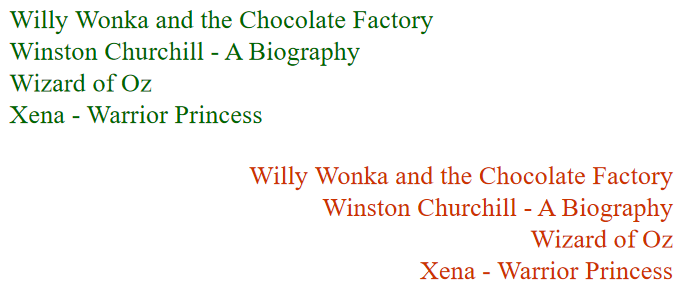
1. **The Gestalt Principles of Grouping 格式塔分组原则**

格式塔(格式塔） 原理解释了眼睛是如何从部分创造一个整体（完形）的



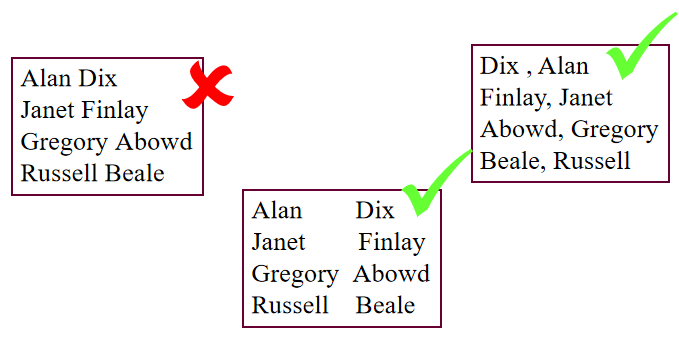
1. **Alignment 对齐**
2. Text 文本

你从左到右读 → 左侧对齐



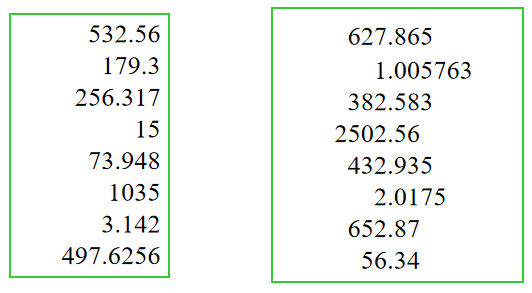
1. Names 名称

通常搜索姓氏 → 让事情简单



1. Numbers 数字

对齐小数点

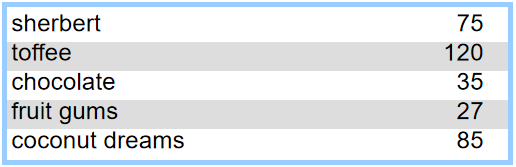


1. **Multiple Columns 多列**

很难跨越空白进行扫描：——通常很难避免使用大型数据库字段



可以使用灰色化：



1. **Grouping and Structure 分组与结构**

logically together → physically together 逻辑上一起 变为 物理上一起

1. **Physical Controls 物理控制**

grouping of items 元件分组：将负责不同功能的按钮分组

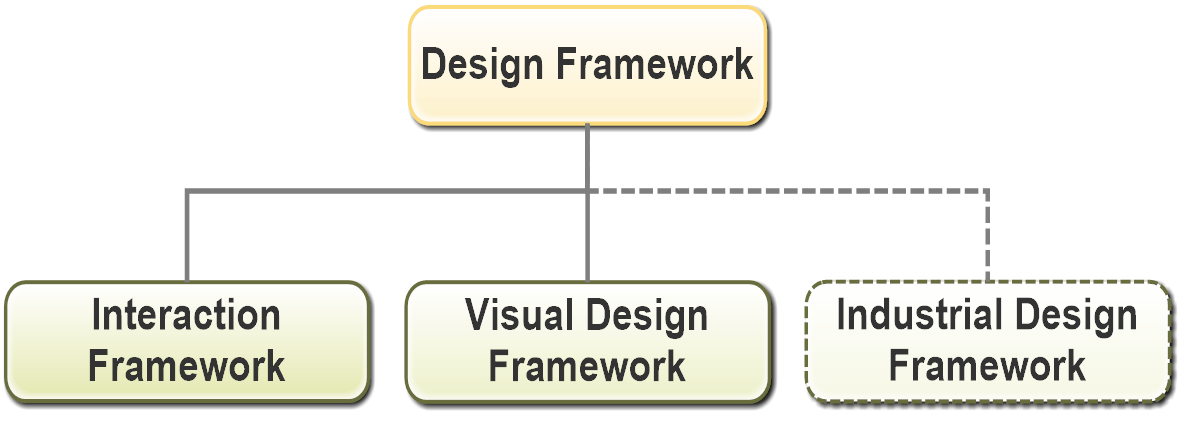
order of items 元件顺序：按照用户操作顺序从上到下设计元件位置顺序

white space 留白：运用空白区域将元件分组

# 第六章 交互设计

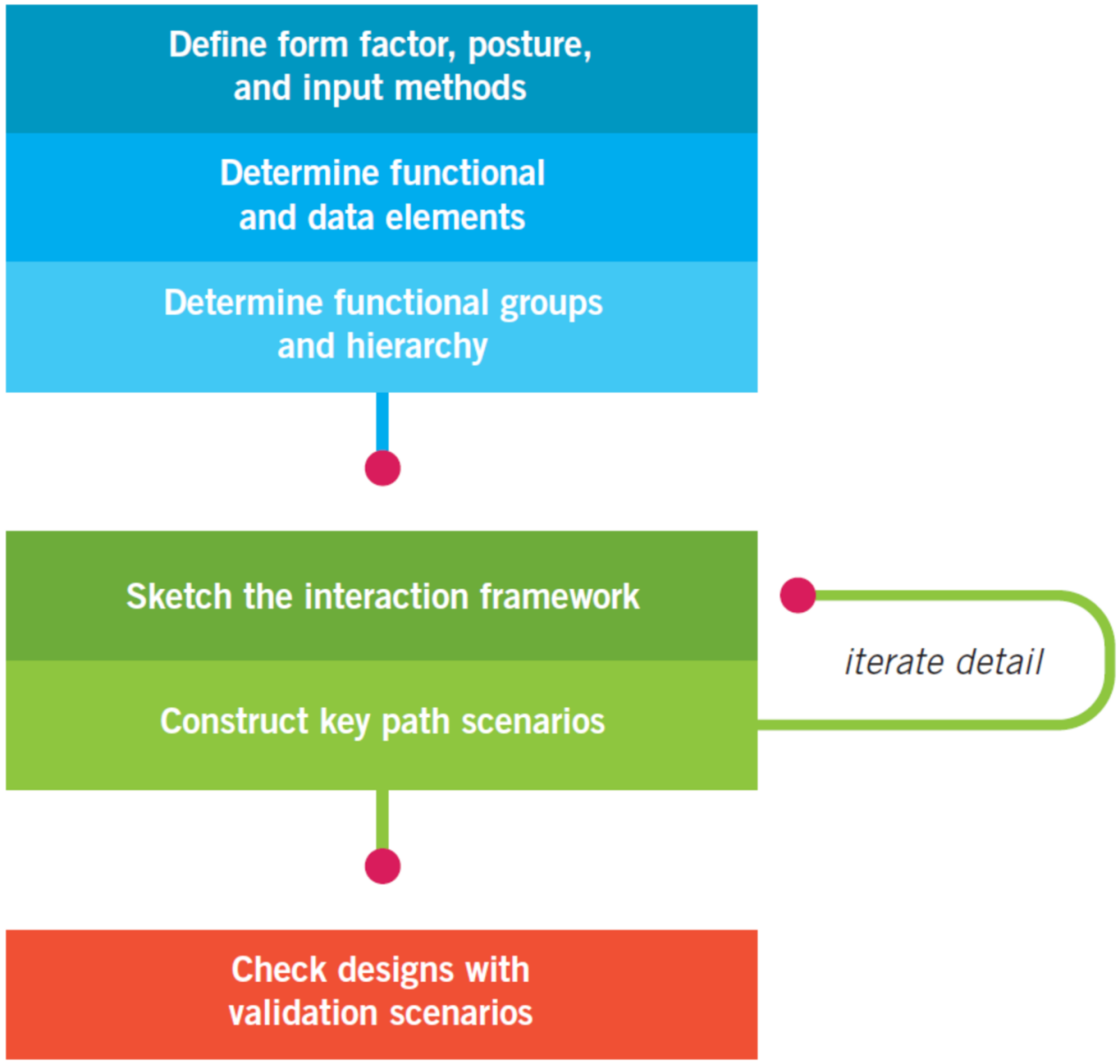
## Design Framework 设计框架

1. **定义用户体验的总体结构**



包括 交互框架、视觉设计框架、工业设计框架

1. 在项目的这个阶段，交互设计师使用**场景**和**需求**来创建构成交互的**屏幕**和**行为草图（rough sketches），**完成**交互框架（Interaction Framework）**。
2. 同时，视觉设计师利用视觉语言的研究来开发一个**视觉设计框架（Visual Design Framework）**，这个框架通常表示为单个屏幕原型的详细呈现。
3. 工业设计师进行**形式语言（form）**的研究，以完成一个粗略的物理模型和**工业设计框架（Industrial Design Framework）**。
4. 服务设计者为**服务框架（service framework）**中的每个接触点构建信息交换模型。
5. 我们发现，素描般的情节串连图板和屏幕，以及以场景形式的叙述，是探索和讨论设计解决方案的一种非常有效的方式，不会产生过多的开销和惯性。
6. **The Framework Definition Process 框架定义过程**

****

尽管我们已经将这个过程分解成了数字顺序的步骤，但这通常不是一个线性的 努力。相反，它发生在迭代循环中。具体而言，步骤3到步骤5可以根据设计 者的思想过程进行切换。

1. Define form factor、Posture and Input Method 定义形状要素、姿态 和输入方法

* 这些形状因素对任何设计都意味着什么约束？

它是一个可以在高分辨率计算机屏幕上查看的web应用程序吗？

它是一款必须小巧、轻便、低分辨率，并且在明亮的阳光和黑暗中都能 看见的手机吗？

它是一个必须坚固耐用的信息亭，以承受公共环境，同时容纳成千上万 分心的新手用户？

* 产品的姿势与之相关：

用户将对与产品交互的关注程度。

产品的行为如何回应用户对产品的关注。

此决定应基于使用上下文和环境。

* 用户将如何与产品交互。

这是由形状因素和姿势，以及你的人物角色的态度，能力和偏好驱动的。

决定哪个组合适合你的主要角色和次要角色。

1. Define Functional and Data Elements 定义功能性和数据元素

* 数据元素通常是互动产品的基本主题。

考虑数据元素之间的关系。

* 功能元素是可以对数据元素及其在接口中的表示进行的操作。

一般来说，它们包括用于执行操作的工具以及可视化和结构化地管理数 据元素的方法。

一个需求通常需要多个接口元素。

* 问问自己哪种可能的解决方案最有可能做到以下几点：（**方案优选原则**）

最有效地完成用户目标。

最符合您的设计原则。

符合技术或成本参数。

可能把互动和竞争区分开来。

最适合其他要求。

1. Determine Functional Groups and Hierarchy 定义功能组合层级

* 以下是一些需要考虑的问题：

哪些元素需要大量的视频区域，哪些不需要？

哪些元素是其他元素的容器？

如何安排容器以优化工作流？

哪些元素一起使用，哪些不一起使用？

一组相关元素将按什么顺序使用？

哪些数据元素有助于人物模型做出选择？

什么样的交互模式和原则适用？

人物模型的心理模型如何影响组织？

* 将数据和函数组织到顶级容器元素中非常重要。

例如屏幕、框架和窗格。

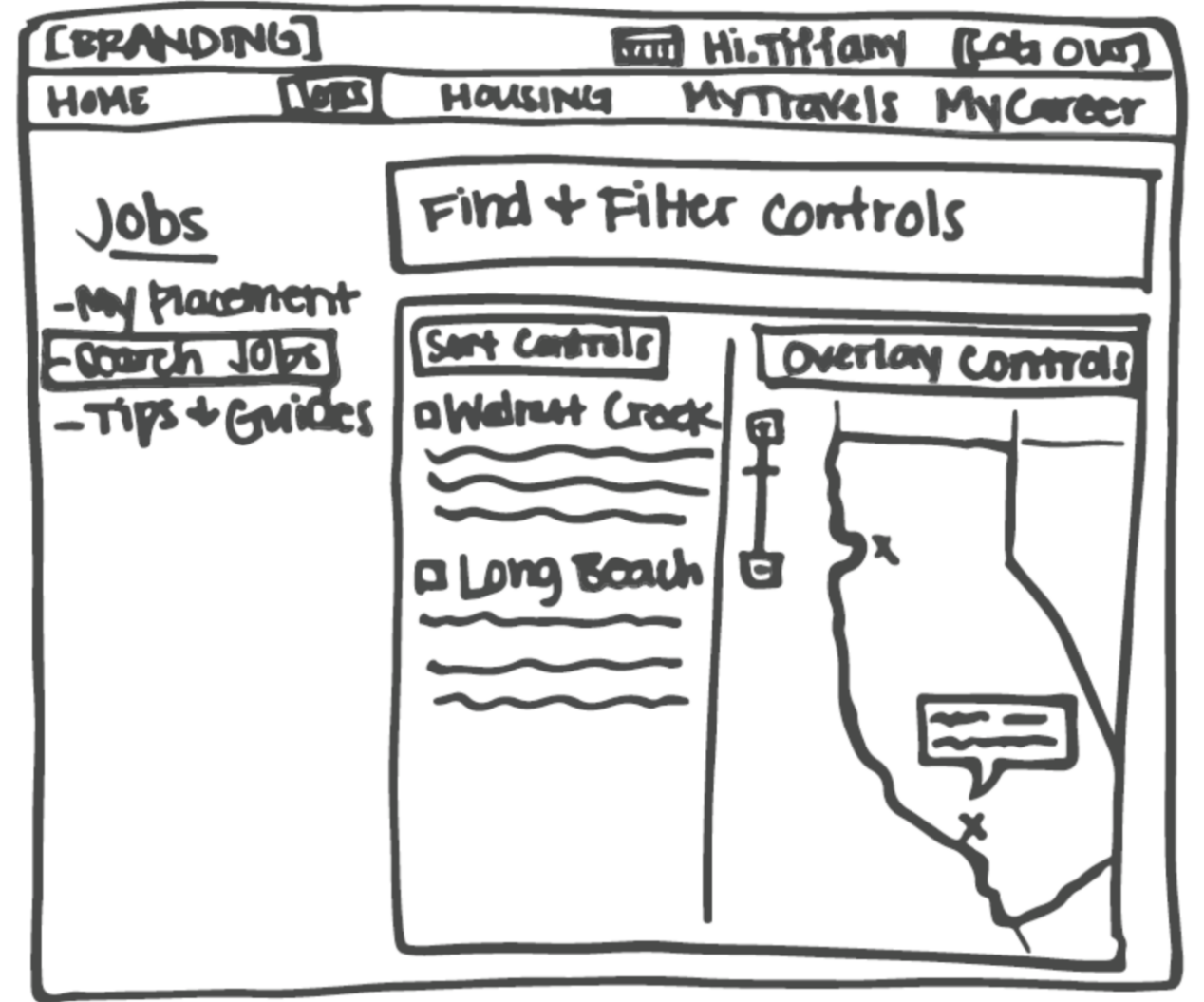
考虑产品需要哪些主屏幕或状态。

在对功能元素和数据元素进行分组时，请考虑在给定产品平台、姿 势、屏幕大小、形状因素和输入方法的情况下应如何安排这些元素。

* 这些分组可能会随着设计的发展而发生一些变化（特别是在绘制界面时），但暂时进行分组仍然很有用。将元素分组。这将加快创建初始草图的过程。同样，缩进列表或简单的维恩图在这一点上适合记录这些关系。

1. Sketch the Interaction Framework 勾画交互框架

* 矩形相位
* 方块图阶段，将视图分为粗略的方块图，对应窗格、控制部分（如控制栏），以及其他高层次容器。
* 然后为每个方块图添加标签和注解，并描述每个分组或者元素如何影响其他分组和元素。
* 方块间的剪头代表流程或状态的改变。



1. Construct Key Path Scenarios 构建关键路径场景

* 关键路径场景使用交互框架的词汇表描述角色如何与产品交互。

A key path scenario describes how the persona interacts with the product, using the vocabulary of the interaction framework.

这些场景描述了通过界面的主要路径，人物角色通常每天都会以最大的 频率使用这些路径。

These scenarios depict the primary pathways through the interface that the persona takes with the greatest frequency, often on a daily basis

关键路径场景侧重于上下文场景中广泛描述和暗示的任务细节。

Key path scenarios focusing on task details broadly described and hinted at in the context scenarios

关键路径场景必须详细描述每个主要交互的行为，并提供每个主要路径 的演练。

Key path scenarios must not describe in detail the behavior of each major interaction and provide a walkthrough of each major pathway.

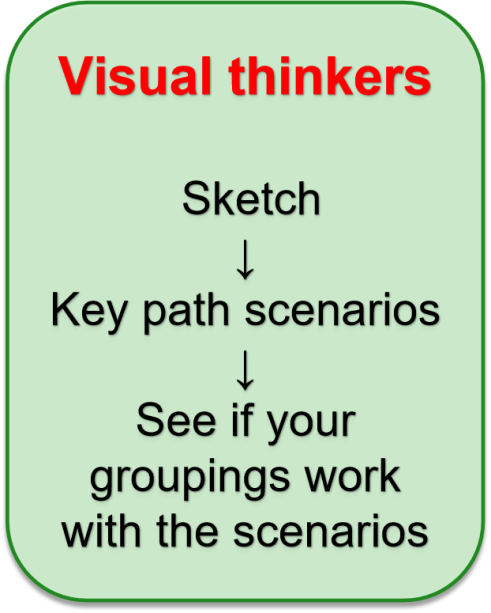
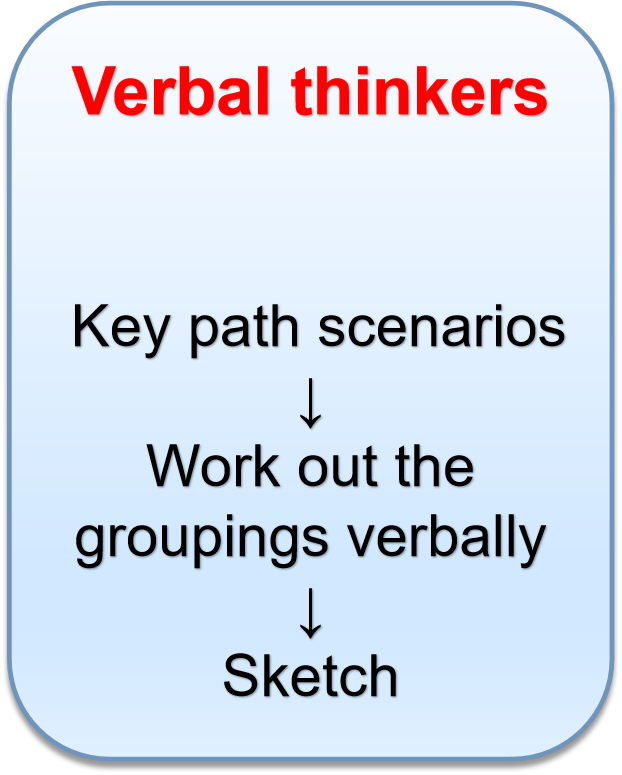
* 故事板

通过使用一系列低保真的草图，并结合关键路径场景的叙述，您可以丰 富地描述所提出的设计解决方案如何帮助角色实现其目标。这种故事板 技术是从电影制作和卡通中借鉴的，在那里，类似的过程被用来规划和 评估想法，而不必处理拍摄实际电影的成本和劳动。

* 过程变化和迭代

通常在步骤之间来回移动，并将整个过程反复几次，直到得到一个可靠 的设计解决方案。

根据您的思维方式，您有几种不同的方法来完成步骤3到步骤5。



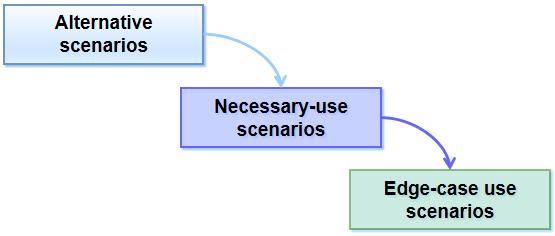
语言思维者 视觉思维者

1. 关键路径情景剧本 （1）勾画草图
2. 进行口头分组 （2）关键路径情景剧本
3. 勾画草图 （3）查看分类在场景是否行得通
4. Check Designs with Validation Scenarios 运用验证性场景来检查设计

* 这些验证场景通常不像关键路径场景那样详细。

目标是在设计中戳出洞，并根据需要进行调整

* 您应该按照以下顺序处理三大类验证场景：



1. 替代场景 （2）必须使用的场景 （3）边缘情形场景
2. **Defining the Visual Design Framework 定义视觉设计框架**

视觉设计框架通常遵循以下过程：

开发视觉体验特征。

开发视觉语言研究。

将选定的视觉风格应用于屏幕原型。

1. Develop Experience Attributes 开发视觉体验特征

* 定义视觉设计框架的第一步是选择一组三到五个形容词，用来帮助定义产品的基调、声音和品牌承诺。这组描述性关键字统称为experience attributes(体验特征）.
* Cooper公司构筑体验特征的过程：

1. 收集任何现有的品牌指南。熟悉它们。如果公司有明确的品牌指导方针建立在一个产品上，那就是你正在设计的产品-你的大部分工作可能已经为你完成了。
2. 收集强大的品牌产品、界面、对象和服务的示例。包括来自特定领域的多个例子会有所帮助。利益相关者思考他们的差异。例如，如果我们包括汽车的图片，我们可能会包括宝马、丰田、法拉利和特斯拉的例子。
3. 与利益相关者合作，确定直接和间接竞争。收集这些产品和服务的产品和接口的示例，并将其包含在示例中。
4. 在你的定性研究过程中，拉取受访者提到的相关术语。特别注意提到的任何痛点。例如，如果许多人提到竞争对手或现有版本的产品很难使用或“不直观”，你可能想讨论“友好”、“简单”还是“简单”“可以理解”是否应该是一个属性。
5. 通过展示的品牌指南、产品示例、竞争情况和用户说明作为参考，与利益相关者讨论您正在设计的产品的子品牌。我们经常要求利益相关者在例子上贴上红色或绿色的标签，投票赞成或反对，然后讨论任何明显的赢家、输家或有争议的例子。
6. 从讨论的结果中，找出定义和区分产品的形容词的最少数量。
7. 如果其中任何一个词有多种含义，请记录其确切含义。”例如，“犀利”可以指精确和圆滑，也可以指智慧和机智。
8. 考虑竞争对手。如果你的一系列属性无法将品牌与竞争对手区分开来，那就对它们进行细化，直到它们区分开来。还要确保个人特质是让人梦寐以求的。聪明固然不错，才华横溢更佳。
9. 与利益相关者（尤其是任何营销人员）就您提议的属性集进行核对，讨论并最终确定它们，然后再继续。
10. Develop visual language studies 开发视觉语言研究

* 这些研究基于体验属性，包括颜色、类型和小部件处理。

它们还包括界面的总体维度和任何“材料”属性。

这些方面抽象而独立于交互设计。

视觉语言研究应与人物角色的体验目标以及需求定义阶段开发的任何 体验或品牌关键词相关。

* 大量的工作往往需要翻译成一个有意义的外观和感觉的互动产品或网站营销宣传品的风格指南。
* 在设计视觉风格时，考虑环境因素和人物性格也是很重要的。
* 一旦你开发了一系列反映人物角色体验目标、品牌指导方针和体验关键词的视觉语言研究，就应该将它们呈现给利益相关者以获得反馈。
* 重要的是根据这些目标和关键词对它们进行语境化，并描述每个方向的基本原理及其相对优点。
* 在进入下一步之前，反复进行视觉语言研究是很常见的。

1. Apply the chosen visual style to the screen archetype 将选定的视觉风格应用于屏幕原型

* 最后一步是将一个或两个选定的视觉样式应用于关键屏幕。

1. **Defining the Industrial Design Framework 定义工业设计框架**
2. 工业设计框架的开发方式与视觉设计框架基本相同。
3. 但是，由于形式因素和输入方法对工业设计和交互设计都有重要影响，因此早期协作以确定相关问题非常有用。
4. 工业设计框架通常遵循以下过程：

与交互设计师就外形和输入法进行合作。

开发粗略的原型。

发展形式语言研究。

1. **Defining the Service Design Framework 定义服务框架**
2. 因为服务设计常常影响组织的业务模型，所以**服务设计框架（service design framework）**可能在其他设计领域之前进行。
3. 服务设计框架通常遵循以下过程：

描述客户旅程。

创建服务蓝图。

创建体验原型。

1. **Refining the Form and Behavior 细化外形和行为**
2. 在这个阶段，过渡到细化阶段，在细化阶段，设计被转化为最终的具体形式。
3. 在这个阶段，原则和模式对于给设计一个良好的形式和行为完成仍然很重要。
4. 细化阶段的标志是将略图故事板转换为全分辨率屏幕，在像素级描述用户界面
5. 在这个阶段，您应该尽可能地处理每个主要视图和对话框。
6. 在整个优化阶段，视觉设计师应该开发和维护一个视觉风格指南。
7. 开发人员在创建界面的低优先级部分时，通常没有时间和资源来完成自己的工作，因此使用本指南可以始终如一地应用可视化设计元素。
8. 同时，工业设计师与机械工程师一起完成部件和装配。
9. 虽然设计过程的最终产品可以是各种输出中的任何一种，但我们通常创建一个可打印的形式和行为规范。

带有足够详细的标注的屏幕渲染，供开发人员进行编码

详细的故事板，以说明随着时间的推移的行为

HTML或Flash中的交互式原型

1. 但是，请记住，原型本身很少足以传达底层的模式、原则和基本原理，而这些是与开发人员沟通的重要概念。
2. **Validating and Testing the Design 验证和测试设计**
3. 要测试的内容

命名分区/按钮标签有意义吗？某些词比其他词更能引起共鸣吗？

组织信息是否按有意义的类别分组？

物品是否位于客户可能寻找的位置？

首次使用和可发现性是新用户容易找到的常见项目吗？

说明清楚吗？需要说明吗？

有效性客户能否高效地完成特定任务？他们在做什么

失误？哪里？多久？

1. 何时考试：总结性和形成性评价

总结性评估用于产品比较，在重新设计之前确定问题，并调查产品退货 的原因以及培训和支持请求。

总结性研究通常由专业的第三方评估人员进行并全面记录。

1. 形成性评估是在设计过程中进行的，通常是在细化阶段。
2. 它可以让设计师看到他们的目标受众对他们提供的帮助他们完成任务的信息和工具的反应。
3. Conducting Formative Usability Tests 进行形成性可用性测试
4. 成功形成性可用性测试的基本要素：

* 在这个过程中测试得足够晚，以至于有一个实质上具体的设计需要测试，而测试得足够早，以至于可以在设计和实现中进行调整。
* 测试适用于手头产品的用户体验的任务和方面。
* 从目标人群中招募参与者，使用你的角色作为指导。
* 要求参与者在大声思考的同时执行明确定义的任务。
* 让参与者直接与低技术原型交互（除非测试专用硬件时，纸质原型无法反映细微的交互）。
* 调整会议的力度，找出问题并探讨其原因。
* 通过使用以前没有参与项目的主持人来最小化偏见。
* 关注参与者行为及其理论基础。
* 在进行测试后向观察者汇报，以确定观察到的问题背后的原因。
* 让设计师参与整个学习过程。

1. **Designer Involvement in Usability Studies 设计师参与可用性研究**
2. 人物角色帮助设计师理解用户的目标、需求和观点，为有效的交流创造基础。
3. 设计师必须通过以下方式参与进来：

规划研究以关注设计的重要问题

使用角色及其属性定义招聘标准

使用场景开发用户任务

观察测试过程

合作分析研究结果

## Design Techniques and Rules 设计技术与规则

1. **Design Solution 设计解决方案**
2. Ethical (considerate, helpful) 合乎道德（体贴、乐于助人）

不要伤害别人

改善人类处境

1. Purposeful (useful, usable) 有目的的（有用的，有用的）

帮助用户实现他们的目标和愿望

适应用户上下文和容量

1. Pragmatic (viable, feasible) 务实（可行、可行）

帮助委托组织实现其目标

适应业务和技术要求

1. Elegant (efficient, artful, affective) 优雅（高效、巧妙、情感）

表示最简单的完整解决方案

具有内在的（自我揭示的，可理解的）连贯性

适当调节和激发认知和情感

1. **Interaction Design Principles 交互设计原则**
2. 交互设计原则是解决行为、形式和内容问题的普遍适用的指导原则。
3. 他们鼓励设计支持用户需求和目标的产品行为，并为我们设计的产品创造积极的体验。
4. **Principles operate at different levels of detail 原则在不同的细节层次上运作**
5. Conceptual概念性原则

帮助定义数字产品应该是什么样的，以及它们在结构上如何适应用户所需的 广泛使用环境。

1. Behavioral行为原则

描述产品在一般和特定环境中的行为。

1. Interface-level接口级原则

描述行为和信息的组织、导航和交流的有效策略。

1. **What is a prototype? 什么是原型**

...正在开发的软件程序版本不完整。原型通常只实现最终程序的一小部分特性， 并且实现可能与最终产品的实现完全不同。

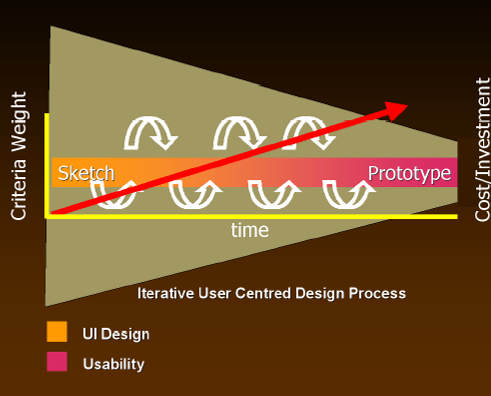
1. **Why prototype? 为什么使用原型**
2. 概念证明
3. 设计验证
4. 管理方式
5. 减少误解
6. 节省时间和金钱
7. 共享通信
8. 用户测试
9. 展示和讲述的力量
10. **Types of Prototyping 原型类型**
11. Low fidelity 低保真
12. Medium fidelity 中保真
13. High fidelity 高保真

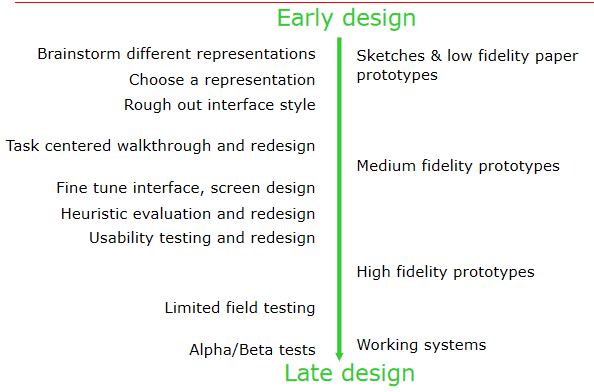
**越高的保真不一定对交互设计更有帮助**

1. **Sketching and Prototyping 草图和原型**

随着设计投资的增加（红色箭头），从设计到评估，审查或接受概念的标准的 形式也随之增加。

同样地，界面设计（创意产生）也发展到可用性测试（创意调试和优化）





1. **Scribble Sketching 自由曲线草绘**
2. 通过快速地勾勒出想法来捕捉想法的本质，从而对现实世界进行采样。
3. 怎么

画得很快（几秒钟）

非常低的保真度

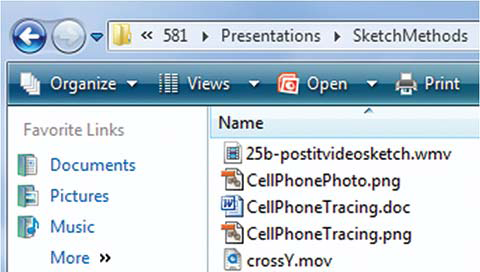
关注和强调理念本质

牺牲所有其他细节

1. 我给你看一张照片

15秒：从图片中选择一个想法或概念进行拍摄

30秒：草草画草图



1. 示例：



示例结果：

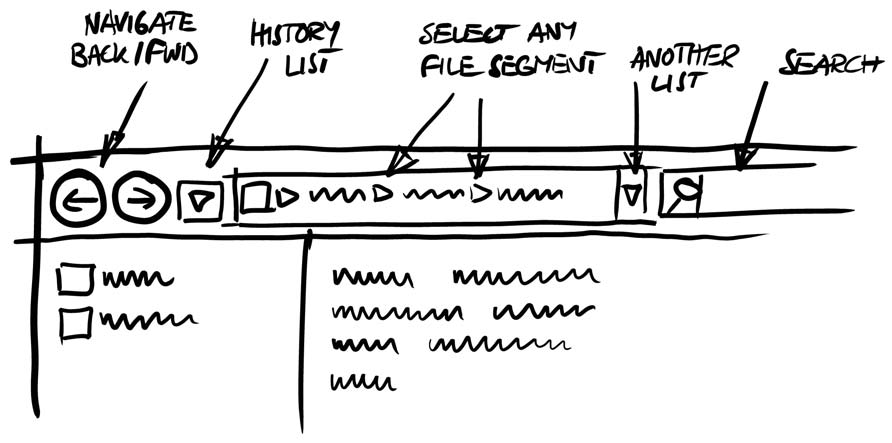
重点：布局

细节：突出显示窗格、关键按钮和字段的结构

摘要：图标/标签/漫画文本(漫画) 涂鸦

遗漏：装饰，实际文本，较少的接口控件

1. 示例：



示例结果：

重点：文件夹导航

详细信息：导航栏中带注释的交互方法

抽象的：图标/标签/文字作为漫画涂鸦

遗漏：装饰，实际文本，较少的接口控件

1. **Low Fidelity Prototyping 低保真模型**
2. 快速发展
3. 允许思想的探索
4. 可能更难进行用户研究
5. 零编码！
6. 纸质模型：界面外观、感觉、功能的快速和廉价的准备和修改
7. 目的

集思广益

引出(引出） 用户反应

引出用户修改/建议

1. **Paper Prototyping 纸张原型**
2. 预期系统外观图
3. 粗糙意味着人们专注于高层次的概念
4. 但很难想象对话的进展
5. 草图的属性：
6. 快：使
7. 及时：需要时提供
8. 可任意处理的：投资于概念，而不是执行
9. 丰富：它们在一系列的想法中是有意义的
10. 清晰的词汇：渲染和样式表示它是一个草图，而不是一个实现
11. 受限分辨率：不妨碍概念探索
12. 与国家的一致性：渲染的细化与概念的实际发展状态相匹配
13. 建议和探索而不是确认：价值在于提出和激发可能的东西，即它们是对话和互动的催化剂
14. **Catalog Shopping System 目录购物系统**
15. 实体店

纸质目录

用条形码阅读器扫描所需项目

请参阅计算机屏幕上的项目

完成并支付订单（提交订单）

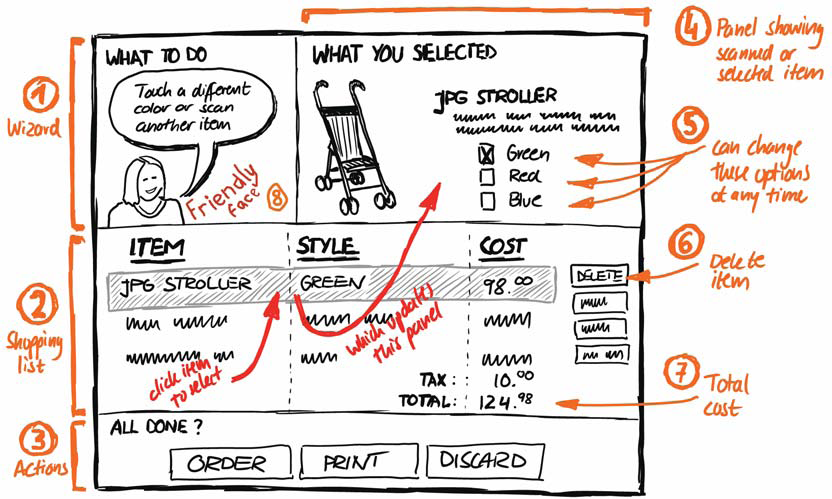
打印出来交给售货员

售货员给您商品

1. 情节提要（故事板）

买一辆蓝色的婴儿车(婴儿车）

1. **Annotations(注释)**



1. 名称、标签、注释

位置标识它们引用的草图零件

1. 例子

邻近区域

大面积支撑

指向特定区域的箭头

1. 也可以

强调感兴趣的领域

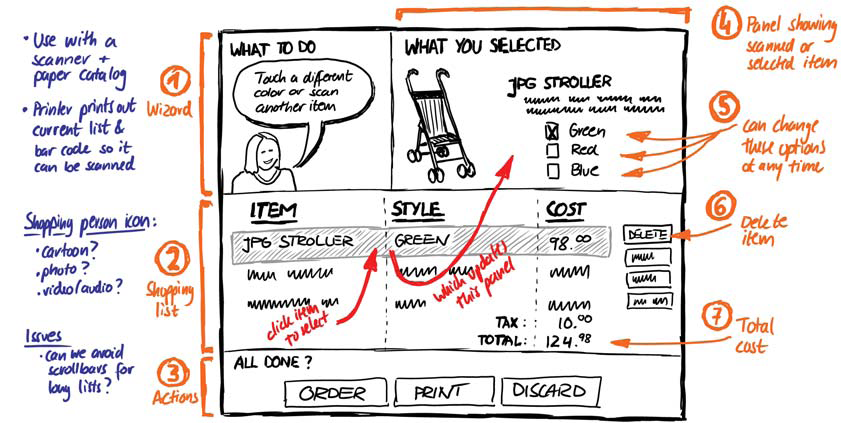
关联草图的不同部分

指示方向

显示移动

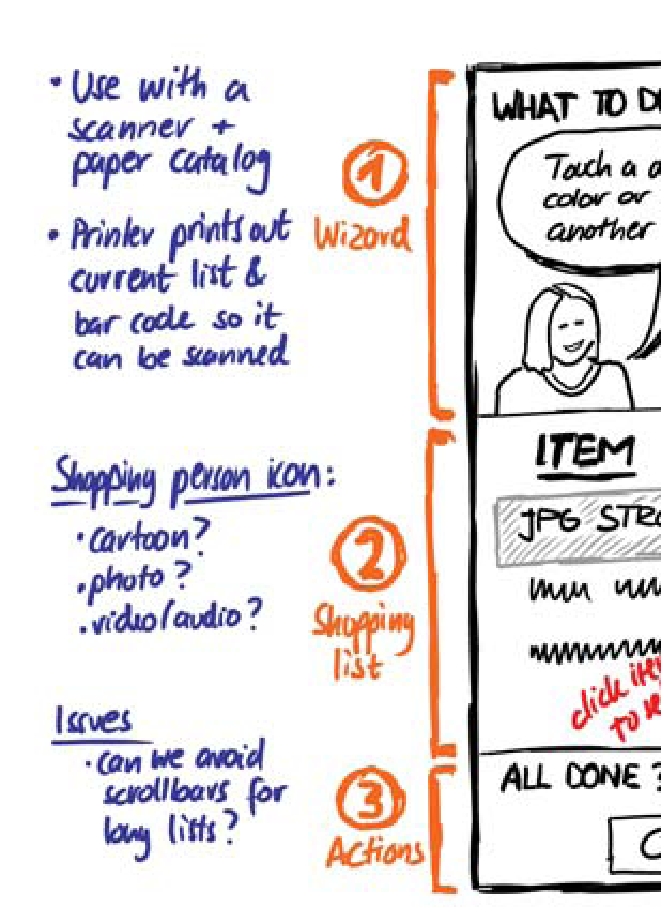
指示事件序列（交互流）

1. 草图中的交互流程：图中央的红色箭头
2. Notes 笔记
3. 空间位置不重要的地方添加文本



（1）关于图纸中遗漏的设计元素的想法

（2）草图元素的替代设计选项 （3）设计问题；



1. 关于图纸中遗漏的设计元素的想法
2. 悬而未决的问题
3. 假设
4. 使用上下文…
5. 可以是单词，列表，段落，句子片段…
6. **Story boarding 故事板**
7. 作为草图的一系列关键帧

源于电影；习惯了一个场景的想法

界面在交互中特定点的快照

1. 用户可以快速评估界面的方向
2. **Sequential Storyboard 连续故事板**
3. Series of key frames as sketches视觉叙事

一系列关键帧作为草图

交互中各点的界面快照

1. Portrays刻画

界面关键场景

导致更改的转换

1. 总之，为每一个界面状态画一个图：如初始界面，扫描完婴儿车后的界面，更换颜色后的界面，点击购买后的界面
2. **Branching Storyboard 分支故事板**
3. 从一个状态的多次转换
4. 结果

用户操作

环境行动

系统配置…

1. 扩展他以允许一个人：扫描2个或更多项目、修改列表中的项目、打印出列表，然后稍后返回并在列表中扫描、不买任何东西、走开、explicitly（明确的）取消订单
2. **Narrative Storyboard（叙述性故事板）**
3. 讲个故事
4. 用户体验

界面只是故事的一部分

1. 叙述的

包括使用上下文

更完整的故事

1. **Interface vs. Narrative Storyboard 界面与叙事故事板**
2. 大纲情节提要框架

5个方框通常足以说明一种情况

1. 发展故事情节
2. 互动发生在哪里？

有什么问题？

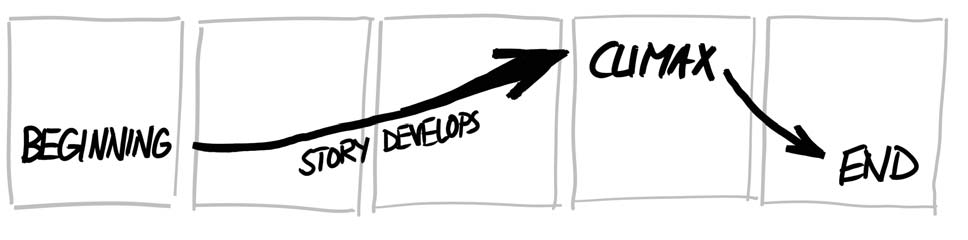
一个人要做的任务是什么？

哪些人在场/他们的行动是什么？

他们使用什么对象/设备？

输入/输出是什么？

他们的行动如何解决问题？



（1） （2） （3） （4） （5）

远景 故事发展到高潮 结局

establishing shot story develops to climax conclusion

1. A Storyline 故事情节

弗雷德经过一个公告板

他对1号公告的更多信息很感兴趣

他用手机捕捉显示屏旁边的条形码

他阅读手机上显示的详细信息

他走开了

1. 1一个人经过广告牌

2注意到一则广告并对更多信息感兴趣

3为海报上的条形码拍照

4移动电话下载有关该产品的详细信息

5那个人把电话收起来，转过身来

1. 草图建立远景
2. 导言

互动发生的地点

相关人员

设置

1. 镜头

长镜头

1. 交替射击好吗

走廊(走廊）

机场

多人…

1. 使用适当的镜头

极远镜头

强调上下文

1. 继续故事情节
2. 使用适当的镜头

越过肩膀

强调某人在看感兴趣的东西

视角

强调人的行动

1. 高潮

特写镜头

显示详细信息

1. 结局

极远射镜头

完成时显示上下文

1. 强调动作和动作
2. 大箭头强调一个人走过
3. 盘旋箭头强调“当他注意到感兴趣的东西时”的运动

问号揭示情感（兴趣）

1. 区域强调拍照行动
2. 重点强调收到的信息的作用
3. 箭头强调人走开
4. 演示和迭代
5. 展示并接受批评(评论）

现实吗？

上下文可信吗？

这个人的动机可信吗

他们真的会这么做吗？

1. 制定替代方案
2. **Medium Fidelity Prototyping 中保真原型**
3. 更“真实”的用户体验
4. 更长的设计时间
5. 更长的开发时间
6. 某种程度的编程
7. “黄金之路”/幻灯片放映
8. **Scripted simulations 脚本模拟**
9. 使用媒体工具创建情节提要

由简单用户输入激活的场景转换

一个简单的垂直原型

1. 用户给出了一个非常紧凑的脚本/任务

看起来像一个真实的系统

脚本会破坏模拟

1. **High Fidelity Prototyping 高保真原型**
2. 更接近现实
3. 更高的设计要求
4. 更多开发时间
5. 可作为其他团队（工程、质量保证、市场营销）的参考平台

## Design Details 设计细节

1. **Huge amount of knowledge points, omitted 巨量知识点，省略了**

# 第七章 评价技术

## 关键词

1. Heuristic Evaluation 启发式评价
2. UAR（Usability Aspect Report） 可用性方面报告
3. Think-Aloud 大声思考

## Heuristic 启发式

1. **What is evaluation? 什么是评估？**

评价的作用：

测试系统的可用性和功能性

评估设计和实现

应在设计生命周期的所有阶段予以考虑

1. **Goals of Evaluation 评价目标**
2. 评估系统功能范围

系统的功能必须符合系统的要求

1. 评估界面对用户的影响

学习这个系统是多么容易啊

可用性

1. 确定具体问题

意外结果/用户困惑

1. **Heuristic Evaluation 启发式评价**

由尼尔森和莫利奇提出。

确定可用性标准（启发式）

10条规则

由**专家**检查设计，看是否违反了这些规定

评价者

评估必须独立进行

数量:3-5（尼尔森经验）

1. **10 Rules 10条规则**
2. Match the Real World 符合现实世界
3. 说用户的语言
4. 使用普通词汇，而不是技术术语

但在适当的情况下使用特定领域的术语

1. 不要限制用户定义的名称
2. 允许命令语言中使用别名/同义词
3. Consistency and Standards 一致性和标准
4. 最小惊奇原则

相似的事物应该看起来和行为相似

不同的东西看起来应该不同

1. 其他属性

尺寸、位置、颜色…

1. 命令

前缀与后缀

1. Help and Documentation 帮助和文档
2. 用户不阅读手册

更愿意花时间朝着他们的任务目标努力，而不是学习你的系统

1. 但手册和在线帮助是至关重要的

通常在用户感到沮丧或处于危机时

1. 帮助应该是

始终可用

易于搜索

Easy to search

与任务相关

1. User Control and Freedom 用户控制和自由
2. 提供撤消
3. 需要确认
4. 让用户控制
5. Visibility of system status 系统状态可见性
6. 让用户了解系统状态

光标更改

选择突出显示

状态栏

别做过头了…

1. 响应时间

<0.1秒：似乎是瞬时的

0.1-1s：用户通知，但不需要反馈

1-5秒：显示忙碌光标

>5s：显示进度条

1. Flexibility and Efficiency 灵活性和效率
2. 为频繁操作提供易于学习的快捷方式

Keyboard accelerators键盘加速器

Command abbreviations命令缩写

书签Bookmarks

历史History

1. Error Prevention 错误预防
2. 比好的错误消息更好的是一个谨慎的设计，它可以在第一时间防止问题的发生。
3. 选择比键入更不容易出错

但别过火了…

1. 禁用非法命令
2. Description Error 描述错误
3. 预期动作被另一个具有许多共同特征的动作所取代

把橙汁倒进麦片粥里

把碗盖错了

1. 避免使用非常相似的描述

长排相同的交换机

相似的相邻菜单项

1. Capture Error 捕获错误
2. 在执行两个不同操作所需的**序列中存在重叠时**发生
3. 尤其是当一个动作比另一个动作频繁得多的时候。
4. 避免此类错误的可能方法

就是最小化重叠序列

试着抓住它发生的地方

1. Mode Error 模式错误
2. 模式：动作有**不同含义**的状态

大写锁定

1. 避免模式错误

消除模式

模式可见性

弹簧加载或临时模式

不同模式下的不相交动作集

1. Recognition, Not Recall 承认，而不是回忆
2. 使用菜单，而不是命令语言
3. 使用组合框，而不是文本框
4. 尽可能使用通用命令（打开、保存、复制粘贴）
5. 所有需要的信息都应该是可见的
6. Error reporting, Diagnosis, Recovery 错误报告、诊断、恢复
7. 要精确；重述用户的输入

不是“无法打开文件”，而是“无法打开名为paper.doc的文件”

1. 给予建设性帮助

发生错误的原因及解决方法

1. 要有礼貌，不要责怪别人

不是“致命错误”，不是“非法”

1. 在要求之前隐藏技术细节
2. Aesthetic and Minimalist Design 美学与极简主义设计
3. “少就是多”

省略无关的信息、图形、特征

1. 好的平面设计

颜色和字体很少，选择得很好

带空格的组

合理对齐控制装置

1. 使用简洁的语言

仔细选择标签

1. **Chunking the Heuristics Further 进一步划分启发式**
2. 满足期望

符合现实世界

一致性和标准

帮助和文档

1. 用户就是老板

用户控制和自由

系统状态可见性

灵活性和效率

1. 处理错误

错误预防

承认，而不是回忆

错误报告、诊断和恢复

1. 保持简单

美学与极简主义设计

1. **Tog’s 16 principles 列车16原则**
2. Anticipation 期待

向用户提供流程每个步骤所需的所有信息和工具。

1. Autonomy 自治
2. Color blindness 色盲
3. Consistency 一致性
4. Defaults 默认值

违约应该容易“吹走”

不要在应用程序或服务中使用“default”一词。

1. Efficiency 效率
2. Explorabel interfaces 探究性接口

为用户提供稳定的“家”的感觉知觉线索

始终允许“撤消”

1. Fitt’s Law 菲兹定律
2. Human interface objects 人机界面对象
3. Latency reduction 延迟削减
4. Learnability 可学习性

理想情况下，产品将没有学习曲线。

在实践中，所有的应用程序和服务，无论多么简单，都将显示一条学习曲线。

1. Metaphors 隐喻
2. Protect user’s work 保护用户的工作

确保用户不会因为错误而丢失工作。

完全不可避免的原因

1. Readability 可读性

高对比度

喜欢白色或淡黄色背景上的黑色文本。

避免灰色背景。

特别注意老年人的需要。

1. Track state 跟踪状态
2. Visible navigation 可见导航
3. **Where You Are – breadcrumbs 你在哪里-面包屑**

显示网站层次结构的路径

1. **Shneiderman’s 8 Golden Rules 施奈德曼的八大金科玉律**
2. Consistency 一致性
3. Shortcuts 快捷键
4. Feedback 反馈
5. Dialog closure 对话框关闭
6. Simple error handling 简单的错误处理
7. Reversible actions 可逆动作
8. Put user in control 让用户掌握一切
9. Reduce short-term memory load 减少短期内存负载

## Heuristic Evaluation 启发式评估

1. **步骤**：
2. 比较UI和启发式
3. 列出可用性问题

用启发法解释和证明每个问题

1. 用启发式方法证明每个问题的正确性

不能只是说“我不喜欢这些颜色”

1. 列出每个问题

即使一个接口元素有多个问题

1. 至少通过接口两次

第一步：了解交互的流程和系统的一般范围。

第二步：关注特定的接口元素

1. 不要把你自己局限于10个启发式

我们见过其他的：菲茨定律，知觉融合，色彩原理

但是这10种启发式比较容易

1. **Heuristic Evaluation Is Not User Testing 启发式评估不是用户测试**
2. 评估者不是用户

用户对用户界面设计一无所知

1. 他发现了UT经常忽略的问题

字体不一致

菲茨定律问题

1. 但是UT是可用性的黄金标准
2. **Hint for Better Heuristic Evaluation 更好的启发式评估提示**
3. 使用多个评估器

不同的评估者发现不同的问题

越多越好，但回报递减

尼尔森推荐3-5名评估师

1. 每个单独的评估者单独检查接口

只有在完成所有评估之后，才允许评估人员进行交流，并汇总他们的调查结 果。

确保每个评估人员进行独立和公正的评估

1. 交替启发式评估与用户测试

每种方法都会发现不同的问题

启发式评估更便宜

1. 可以记录评估结果

每位评估人员的书面报告

需要评估人员付出额外的努力

需要由评估经理阅读和汇总。

评估者在通过界面时向观察者口头表达他们的意见。

上一次评估会议结束后不久即可进行评估

1. 观察者可以帮助评估者

评估人员的领域专业知识有限/需要解释接口的某些方面。

这在用户测试中是不行的

对于传统的用户测试，人们通常希望发现用户在使用界面时所犯的错误

通过使用系统发现他们问题的答案

1. **Formal Evaluation Process 正式评估过程**
2. Training培训
3. 设计团队和评估人员会议
4. 介绍应用程序
5. 解释用户群、域、场景
6. Evaluation评价
7. 评估人员分开工作
8. 生成书面报告，或由观察员记录的口头评论
9. 专注于产生问题，而不是对问题的严重程度进行排名
10. 每个评估员1-2小时
11. Severity Rating严重等级
12. 评估人员优先考虑发现的所有问题（不仅仅是他们自己的问题）
13. 取评估者评分的平均值
14. Debriefing汇报
15. 评估人员和设计团队讨论结果，集思广益解决方案
16. **Severity Ratings 严重等级**
17. Contributing factors 促成因素

Frequency 频率：有多普遍？

Impact 影响：克服有多难？

Persistence 坚持：多久克服一次？

1. Severity scale 严重程度量表

Cosmetic化妆品：不需要修理

Minor 次要：需要修复，但优先级较低

Major 专业：需要固定和高度优先

Catastrophic灾难：必须修复

1. **Evaluating Prototypes 评估原型**
2. 启发式评估适用于：

草图

纸质原型

不稳定原型

1. 在草图上很难找到“缺少元素”的问题

因为您实际上并没有使用该接口，所以不会因为缺少该功能而受阻

仔细找他们

## How to Write a Usability Aspect Report (UAR) 如何编写可用性方面报告（UAR）

1. **The Elements of a UAR UAR的元素**
2. UAR Identifier — <Problem or Good Feature> UAR标识符-<问题或良好功能>
3. 这应该是一个唯一的标识符，以便归档。
4. 如果不止一个人在项目中工作或使用了不止一种分析技术，则此标识符可以包含字母和数字。
5. 如果报告涉及到界面的可用性问题，则在唯一标识符后面加上“问题”，如果报告描述了界面的某个方面，则在重新设计时应保留“良好特性”。
6. Succinct description of the usability aspect 可用性方面的简洁描述
7. 当您谈论此UAR与其他UAR的关系时，此描述将用作此UAR的“名称”。
8. 使名称尽可能短（大约三到五个单词），但仍然是描述性的，并与系统的其他方面区分开来。
9. 如果这个UAR是关于一个问题（而不是一个好的特性），请确保您有一个描述问题的名称，而不是一个解决方案。
10. Evidence for the aspect 这方面的证据
11. 这是客观的支持材料，证明你认为值得报告的方面是正确的。
12. 本节需要包含足够的信息，以便此UAR的读者了解是什么触发了报告。
13. Explanation of the aspect 方面的说明
14. 这是你对证据的解释。
15. 您需要在本说明中提供足够的上下文，以便读者理解问题，即使他们对系统或域的了解不如您。
16. Severity of the problem or benefit of the good feature 问题的严重性或良好特性的好处
17. 这就是你的推理，它是多么重要，要么修复这个问题，要么保持这个良好的功能。
18. 这包括用户体验这方面的频率，他们是否可能了解它的工作原理，它是否会影响新用户、临时用户、有经验的用户等。
19. Possible solution 可能的解决方案
20. 如果这方面是一个问题，这是提出解决办法的地方。
21. 一旦发现问题，就不必有解决方案。
22. Relationship to other usability aspects (if any) 与其他可用性方面的关系（如有）
23. 这是您记录此UAR与哪些UAR相关以及它如何相关的语句的地方。
24. 确保所有相关的UAR相互指向。

## Basic Issues in Think-Aloud Testing 有声思维测试的基本问题

1. **What is Think-Aloud Usability Testing? 什么是高声思考可用性测试**
2. 它是一种评估接口原型可用性的技术。
3. 该技术的本质是让用户在系统上执行任务时大声思考；您可以静默观察并了解用户对任务的看法以及用户在使用系统时遇到的问题。
4. **Think-Aloud Protocol Analysis 大声思考协议分析**
5. 认知心理学研究者对
6. 人们如何解决问题
7. 人们关注哪些信息的细节
8. 它们是如何表示这些信息的
9. 他们是如何带来先前的知识来承担的
10. 在解决某些难题或执行某些任务的过程中，他们对信息做了哪些转换
11. 此方法的两个部分
12. 收集有声思维数据（协议）

对理解计算机系统的实用性非常有用

1. 通过建立数据模型来分析数据

在大多数UI设计中没有用处

1. 思想的三种表达方式
2. 类型1：大声说话
3. “大声说”：试着让这些信息在他们进入WM后“从他们嘴里”出来
4. 例子：

给一个人一个简单的加法问题：2+4=？

1. 要求一个人在工作时“大声说话”并不会改变他们的思维策略，也不会减慢他们的思维速度。
2. 人们用来解决现代GUI计算机系统问题的许多信息本质上不是语言信息。
3. 类型2：大声思考
4. “大声思考”并不能改变人们思考问题的方式。
5. 但它确实让他们慢了下来。
6. 这是可用性研究中最常用的协议类型。
7. 类型3：中介过程
8. 当你要求某人口头表达，但要求他们对信息进行更多处理时。

解释他们的想法。

只报告一种类型的信息

1. 不建议用于UI设计，因为心理学家已经证明，这确实会改变人们解决问题时的思维方式。
2. 回指WM

一旦这些东西被人理解了，就存储所有的感知结果。

1. WM

问题解决方案中的所有中间状态，在解决方案的过程中得到的信息。

1. 大声思考协议背后的理论
2. WM有很多线索可以帮助你了解一个人在解决问题或完成任务时的想法。
3. 人们可以用言语表达他们工作记忆的语言内容。
4. **Critical Incident Analysis 关键事件分析**
5. 第二次世界大战

收集数据

分析一下

1. 原始关键事件技术

观察员报告重大事件

观察结果通过访谈或问卷收集

分析人员对收集到的观察结果进行分类和解释，并将其纳入最终报告**。**

1. 关键事件技术的一般程序

1有人在现实世界中执行任务。

2观察者在事后通过分析员的访谈或问卷调查报告重大事件。

3分析员对观察结果进行分类和解释。

4分析员写一份数据和解释的总结报告。

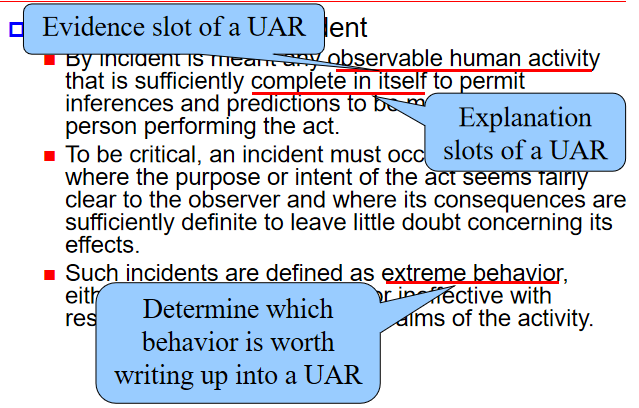
1. 用户界面设计过程中的可用性研究

1当用户使用被评估的计算机系统的原型执行任务时，通常在实验室环境中， 通常是录像或使用屏幕和语音捕获软件，用户会大声思考。

2分析员查看有声思考协议会话的记录，并使用UAR格式报告关键事件。

3分析员对观察结果进行分类和解释。

4分析员写一份数据和解释的总结报告。



1. **Ethics for Empirical Studies 实证研究伦理学**
2. 四个重要的伦理观念
3. 测试接口，而不是参与者
4. 自愿参与
5. 一旦人们开始研究，就不要对他们施加任何压力。
6. 允许参与者随时停止。
7. 对人们表达停止会话愿望的多种方式保持敏感。
8. 保持匿名
9. 在讨论或撰写数据时，请始终使用参与者编号，而不要使用参与者的姓名。
10. 你必须有参与者的名字，因为你可能在将来某个时候需要同意书
11. 知情同意
12. 这个实验是关于什么的
13. 将采用什么程序
14. 他们将得到什么补偿
15. 他们随时可以停下来离开

## How to Conduct a Think-Aloud Usability Test 如何进行自言自语可用性测试

1. **Think-Aloud usability test procedure 可用性测试程序**
2. Define the study’s framework 确定研究框架

开发团队应该自问问题

1. 我们希望在可用性测试中评估哪些类型的使用？走上去用，第一次用？第一次使用这个特殊系统的任务领域熟练的用户？没有时间压力的探索性使用还是时间压力下的目标导向使用？或者，其他方面？
2. 这个系统的可用性目标是什么？90%的用户不用培训就能在3分钟内完成一项简单的任务？接受过使用该系统培训的一半人在执行每项任务时会出现少于一个可恢复错误？
3. Choose what to observe 选择要观察的内容

任务的选择受几个因素的影响

1. 任务的内容
2. 类似的系统
3. 包括最频繁的任务
4. 包括涵盖系统功能范围的任务

从头开始创造事物

删除或修改现有内容

错误恢复任务

1. 非常重要的任务，尽管可能很少
2. 需要培训才能完成任务
3. 这个集合变成了“学习做任务A”和“任务A”，而不仅仅是“任务A”
4. 测试培训材料以及系统

纸质标准物质

网页

讲座

在线教程

1. 任务的持续时间
2. 持续时间

每次不超过一个小时

一天不超过两个小时

包括完成任务的培训

第二天回顾培训

1. 任务的集成
2. 测试套件中足够长的任务
3. 测试您的系统与他们可能使用的其他系统的集成
4. Prepare for the think-aloud usability test 准备“大声思考”可用性测试
5. 在进行任何实际的有声思考之前的准备工作
6. 建立数据收集的现实情况

* 这种情况会因所测试的系统而异。

个人电脑的桌面系统——一种类似办公室的装置

PDA——几乎任何地方

汽车导航装置驾驶或驾驶模拟器

1. 写下任务场景

* 在另一张纸上
* 一次完成一项任务
* 纯口头的
* 包括图片或图表
* 将系列中的任务链接成有意义的叙述

1. 练习课时

* 不要低估困难

硬件和软件

说明必须清楚

数据采集设备必须工作

* 测试的准备阶段

先自己练习

接下来和朋友一起练习

最后，与你将在学习中使用的用户有相似背景的人一起练习

1. 招募用户

* 与最终用户拥有相同的背景知识
* 可能需要补偿用户的参与
* 至少需要三到四个参与者
* 迭代测试

1. Introduce the participants to the observation procedure 向参与者介绍观察程序
2. 概括描述研究目的
3. 简要介绍整个研究，包括以下内容。

* 自我介绍（姓名和职务）
* 介绍您的组织及其宗旨
* 说明特定研究的目标
* 说明您正在测试计算机系统；您没有测试用户
* 告诉他们，他们的参与完全是自愿的，如果他们想在任何时候停止，这是可以的。
* 给参与者一份同意书，给他们时间安静地阅读，并要求他们签名。
* 向参与者展示房间里的设备，解释每件物品的用途，并演示他们将要使用的设备
* 向学员演示如何录制他们的声音和动作。

1. 训练用户“有声思考”
2. 下面的说明引出了用户良好的有声思考行为。

* 你对什么感兴趣？您希望用户做什么？

你在执行任务时的想法是什么

告诉我你在想什么

我希望你经常大声说话

我不想你试图计划你说什么或试图向我解释你在说什么。

如果你长时间保持沉默，我会请你谈谈。

* 从一些解决我们头脑中所有问题的练习题开始。

开发人员先演示

我妈妈家有多少扇窗户？

然后用户转向

当你说出20种动物的名字时，请大声思考。

1. 从一些练习题开始
2. 解释观察的规则
3. 在观察期间，你将无法回答问题。
4. 不管怎样，他们应该问自己的问题。你将记录下来，并在观察结束时回答他们。
5. 如果他们沉默了很长时间，你会要求他们继续说话。
6. Conduct the observation 进行观察
7. 介绍观察阶段
8. 开始观察
9. 总结观察结果
10. Analyze the observation 分析观察结果
11. 建立关键事件的标准
12. 查看记录的行为并写入UAR
13. Find possible redesigns 寻找可能的重新设计
14. 关联不同的可用性方面
15. 确定可能的解决方案
16. Write a report 写报告

三级严重程度

1. 这个可用性方面必须在产品发货前修复，否则产品将对客户不可用。
2. 这个可用性问题应该得到解决，即使这意味着要花费一些资源来解决。
3. 在产品发布之前解决这个可用性问题是可取的，但前提是这样做的努力微不足道（也就是说，如果修复成本几乎为零）。

## Think-Aloud Testing vs. Heuristic Evaluation 有声思维测试与启发式评价

1. **Think-Aloud Testing vs. Heuristic Evaluation 有声思维测试与启发式评价**
2. 在HE中确定的许多可用性方面在“有声思考”可用性测试中得到了确认。
3. 这些技术不会完全重叠，最好是相互配合使用。
4. **"False Alarms" vs. True Problems “假警报”与真问题**
5. 适用于可用性问题的两条经验法则，这些问题在启发式评估中已经确定，但在有声思维测试中没有得到确认。
6. 回顾对有声思维数据的观察，看看在HE UAR测试过程中是否确实出现了这种情况。
7. 特别是，如果你正在设计的系统是一个真正的用户会经常使用的系统（例如字处理器或电子表格），那么即使在有声思维测试中没有出现问题，也应该对使用启发式的灵活性和效率预测的问题采取行动。这种启发式方法预测的是熟练用户的可用性，而不是新用户，而有声思维测试通常不会为这种类型的用户提供数据。
8. **Think-Aloud Usability Tests Can Show Things HEs Can't Show 高声思考可用性测试可以展示HEs无法展示的东西**
9. 当一个HE是用一个纸上的原型（即，仅仅是一个设计的草图）完成的时候，它不能识别系统的动力学问题。
10. 在实践中，HEs通常不根据真实世界的任务来构建分析框架，而有声思维测试通常试图尽可能地模拟真实的任务情况。
11. 可以肯定的是，任何可用的分析技术都无法预测用户所做的事情。