

1. 用户体验

- 用户体验并不是指一件产品本身是如何工作的，用户体验是指“产品如何与外界发生联系并发挥作用”，也就是用户如何“接触”和“使用”它。
- 用户体验总是体现在细微之处，却十分重要。
- 正确的产品形态绝不是由“功能”所决定的，而是应该由“用户自身的心理感受和行为”来决定的。
- 提供优质的用户体验是一个重要的、可持续的竞争优势。用户体验形成了用户对产品企业的整体印象，界定了企业和竞争对手的差异，并且决定了用户留存。
- “特性”和“功能”总是重要的，但在特性功能类似的情况下，用户体验对于用户的忠诚度有更大的影响。
- 创建吸引人的、高效的用户体验的方法称为“以用户为中心的设计(UCD)”。UCD思想非常简单,即开发产品的每一个步骤中都要把用户体验列入考虑范围。
- UCD的整个开发流程，都是为了确保用户在你的产品上的所有体验都在你“明确的、有意识的意图”之内。要考虑到用户有可能采取的每一个行动的每一种可能性，并且去理解在这个过程的每一个步骤中用户的期望值。

2. 产品设计五层面

- 战略层->范围层->结构层->框架层->表现层
- 在每个层面中我们可用的选择，都受到其下层面所确定的要点的约束。
- 并不是说“较低层面”的决策都必须在设计“较高层面”之前作出，在“较高层面”中作出的决定有时候会促成对“较低层面”决策进行一次重新评估。重要的是，无论不同层面决策的先后顺序和因果关系，保持相邻层面的严格依赖关系才是保证产品设计不偏离轨道的核心。
- 由于上述原因，不应该要求每个层面的工作在下一个层面可以开始之前完成，而是应该要求每个层面的工作可以在下一个层面可以结束之前完成。所以一个产品的设计层面的完成顺序是从下而上的，战略层最先完成，表现层最后完成。
- 推导出了一种层次设计的协作规则：同步决策，依赖调整，逐层确定。“同步决策”即各个层面可以根据“上一层的要素”、“新发现的用户需求”、“竞品功能”、“市场动态”等信息独自做出本层面的新决策。“依赖调整”即各个层面作出新决策后，应该从层次结构的依赖性上去重新分析和调整层次设计的要素，使得新的决策符

合层次依赖的原则；“逐层确定”是指在每个阶段性结束设计的过程中，整个层次结构的最终确定顺序是从下而上的，确定了下层，才能确定上层。

3. 战略层

3.1 战略层的两个基本问题

- 产品目标 - 产品的创造者要通过这个产品得到什么？
- 用户目标 - 产品的使用者要通过这个产品得到什么？

3.2 产品目标

- 为了创造成功的用户体验，每一个我们做出的决定都应该建立在我们确切地了解这个决定的影响力的基础之下。明确地定义“成功的条件”而不是定义“通向成功的路径”，才能保证我们不会再这个阶段跑得太快。
- 品牌识别是基础的产品目标之一。在用户与产品交互的过程中，企业的品牌形象就不可避免的在用户脑海中形成了。将品牌形象具体地写进产品目标，将会提高向用户呈现出积极的品牌形象的机会。
- 明确可度量的成功标准是产品目标必不可少的指标。对于指标的分析需要做到全面的了解事情的全貌，避免断章取义，做出了错误的判断。

3.3 用户目标

- 用户细分将产品用户划分为较小的、有共同需求的小组，以此来帮助我们更好地了解用户需求。
- 用户细分的常见标准有人口统计特性、用户对技术的熟悉程度等。
- 用户细分不仅是因为不同的用户组有不同的需求，还有可能因为不同的用户组的需求互相矛盾，此时要么选择针对单一用户群设计而排除其他用户群，要么选择为不同的用户群提供不同的功能。
- 在用户体验设计的过程中，人物角色是从用户研究中提取出的、可成为样例的虚构人物。当我们决定产品的用户体验时，必须要谨记这些虚构人物所代表的不同用户群的需求。为了帮助我们记住他们的需求，可以将他们的信息和照片做出人物卡片放在设计工作区最显眼的地方，当我们要做决定时，问问他们怎么有什么反应。、..

3.4 持续改进

- 战略应该是设计用户体验的流程的起点，但这并不意味着项目开始之前需要完全确定战略。战略应该是可以演变和改进的，当战略被系统的修改和校正时，这些工作就能成为贯穿整个过程的、持续的灵感源泉。

4. 范围层

4.1 文档

- 通过文档来定义项目范围和产品需求，从而使得团队明确项目的目标是什么，什么时候该达到这个目标。对于在项目过程中新发现的想法和创意，需要仔细考量是否符合当前的项目目标，当前项目阶段是否有资源完成。对于符合项目目标但是暂时难以实现的想法创意，使用文档管理起来，可以作为后续版本的设计基础

4.2 范围层的基本问题

- 为了达到战略层的产品目标和用户目标，我们要开发的是什么？

4.3 确定需求和规格

- 在范围层，我们要确定的是所有的功能需求和功能规格。功能规格在功能需求确定后才开始撰写，同时加入功能要求的具体细节。

4.4 功能规格说明应该是乐观的、具体的、客观的。

5. 结构层

5.1 结构层主要关注功能的交互设计和信息架构。

5.2 概念模型

- 用户对于“交互组件将怎样工作”的概念称为概念模型。使用人们熟悉的概念模型会使用户很快的了解这个功能。一个概念模型可以反映系统的一个组件或者整个系统。

- 我们不需要告诉用户我们使用了什么概念模型，好的概念模型使得用户使用产品时几乎是凭直觉的，因为这个产品的交互行为以他们隐含的期望值完全相符。
- 将现实世界中相对应的实物的比喻放入我们的概念模型中是重要的，这对系统功能的设计具有一定的价值，但是更重要的事不能将比喻从现实世界中一字不差的照搬过来。

5.3 错误处理

- 错误处理的问题-当用户犯错误时系统要怎么反应，并且当第一次出错发生时，系统如何避免用户持续出错。
- 最好的防止错误的方法：设计一个用户不可能犯错的系统。
- 防止错误的方法二：让错误难以发生（避免善意的自动修正）。

6. 框架层

6.1 结构层决定了产品将以什么方式运作，框架层则确定产用什么样的功能和形式来实现。

6.2 一个设计良好的界面是要组织好主流用户最常采用的行为，同时让这些界面元素以最容易的方式获取和使用。

6.3 仔细考虑界面呈现给用户的每一个选项和按钮的默认值，然后自动记住用户最后一次的选择状态。