

# 平衡5E:可用性

## Whitney Quesenbery

我们通过可用性想表达什么?在我们决定完成它之前,我们需要理解我们试图完成的它是什么。它不足使我们宣称:从现在开始我们的软件更加用户友好或我们将更加关注用户。

功能需求回答这样的问题:“这个程序需要做什么?”。可用性需求回答不同问题:用户如何完成这个工作,他们如何理解这些任务?他们如何判定一个成功的体验?

语言的弹性会引起一些混淆。很难用一个单词来覆盖一个大的概念,而且可用性就是一个大的概念。“可用性”这个词实际上已经表现一个概念的集合,它包含了用户、设计者、开发人员和软件的关系。在一个特定的句子中,它可以用来表示以下含义:

- 最终产品的质量
- 创建可用软件的过程
- 用于创造这个结果的特定技术
- 和人们一起用思想进行设计的哲学

在ISO924-11中有关可用性的正式定义是:“可用性是某个特定产品在特定使用环境下为特定用户完成特定用途时所具有的有效性(effectiveness)、效率(efficiency)和用户主观满意度(satisfaction)”。  
[1]

我后面用“可用性”这个词表示一个产品(软件、网站或其他开发项目)的质量或属性,即符合使用人们的需要,允许他们工作或娱乐,通过它可以实现他们自身的用途,而且很适合使用。虽然可用性有很多方法论和变化的方法,和其他任何一个相似的新实践一样,存在一些共同的地方:

**关注理解整体使用上下文。**它还不足以能引导软件;他们必须能完成他们的任务,符合他们的目标,而且以一种能够在他们环境中能够有意义的方式去这样做。

**在过程中评估和迭代。**大多数实践者依赖研究或用户中的反馈,而不是简单信任设计者中的专家经验,从而第一次就做正确。这同样意味着在发布之前我们必须改进产品。

**一个以用户为中心的设计方法。**当产品开发关注用户目标、精神模型、任务和需求时,整个过程从“使它工作起来”转换到“使它正确地工作”。当金·古德温(Kim Goodwin)在下篇文章指出,可用性测试和迭代优化可以使一个已有的产品正确地工作起来。而以用户为中心的产品可以提高一个阶-可以使我们做一个正确的产品。[注1]

注1: 见金·古德温(Kim Goodwin)在本期12-15页的文章《Are You Making the Product Right or Making the Right Product?》

**为特定的用户设计。**可用性意味着不只是容易使用。人们和软件都很复杂,而且可用性更包含了多个层面。它需要平衡用户交互中的很多不同方面。

所以是否有一个可用性简单的标准,为什么我们只写设计和开发的指导和创建风格指南就可以保证所用的应用是可用的呢?

答案的部分隐藏在设计的天性中。虽然有一些基本的、通用的原则我们可以在设计中使用,但有很少的准确和快速的原则。例如,在《通用的设计原则》(*Universal Principles of Design*) [2]书中给出了一个折中的集合,包含了100个基本的原则,如黄金分割、欧可汗姆的剃刀

(Ockham's razor), 校准和高亮。每一个对于设计决策来讲都是一个很有用的工具,但书里没有给出处方。书中通用的就是原则,而不是设计中的特定用例。相似的,可用性原则给了我们工作的一个起点。可用性专家有时调侃可用性的座右铭“这要看什么情况”(it depends)。所要看的,不仅是需要理解好的设计、或典型人物,而且特殊人物和他们的特定的目标和任务。所有工作的要点是允许你拥有足够的信息理解可用性在实际背景中的含义以做决策。

### 可用性的维度

当我检查可用性文献时,我发现可用软件所具有的大量的、好的品质。它们包含如用户友好、易学、可发现性、质量、有用的和阻止错误。在可用性工程中,雅可伯·尼尔森(Jakob Nielsen)给出一个产品的五个属性:易学性、效率、可记忆性、容错(低错误,容易恢复)和满意度。从我的观点,我认为有:

- 有效性 (Effective)
- 效率 (Efficient)
- 吸引人的 (Engaging)
- 容错 (Error tolerant)
- 易学 (Easy to learn)

首先，使用首字母都是E的词刚开始看做是一个游戏，但我同样想寻找一个方法来使可用性的维度方便记忆，于是5E诞生了。

### 有效性

有效性是第一个E。主要表明软件是可用的，而且帮助用户准确地实现他们的目标。如果用户不能实际完成他们准备做的事情（或做了不必要的事情），无论体验短还是长，容易还是长，它可能都没有意义。最后他们没有完成任务或达到他们的目标。如果我们能够测量有效性，我们可以理解人们如何定义成功或有用，而且这个相对容易明白或更微妙。

### 效率

效率是所做工作的速度（与精确性要求相关）。效率可以被详细地定义；例如，在一个呼叫中心，衡量客服人员一天能处理的呼叫数量。或者它是一个主观的判断，当一个任务执行“太长”或需要“太多的点击”。

### 吸引人的

关于这个的简单定义就是一个要用界面所带来的愉快、满意或兴趣程度。所有的软件都会给用户带来情绪上的影响，但这个维度的重要性随着程序的类型会发生变化。在一个工作应用中，一个吸引人的界面可以使人投入工作，帮助他在工作中建立信心，或在表达信息方式上特别容易阅读。这个可视的表示和风格或交互的质量都会使软件吸引人，或放弃。

### 容错

容错包含产品防止错误的程度和帮助用户从错误出现中恢复。一般都被称为“错误无关”或“防止错误”，虽然错误和事故和误解。轻触鼠标就会点击一下。如果你错读了一个连接，需要按原路返回，或输入相关内容。实际测试是在错误产生时了解软件如何进行帮助的。

### 易学

易学和产品如何支持首次引导和更深度的学习相关。一个产品可以被使用一次，或一会儿，或一天。它可以用于支持一个容易的或复杂的任务，和任务中的用户是一个专家或新手。但每一次使用，界面必须被记忆或重新学习，而且过一段时间产品可能被开发新的特性。

### 平衡的问题

如果在每个产品中，对于每个用户，可用性的每个要素都是同等重要，那么就会出现图1中的情景。但他们不是，而且这个提供了一个机会来与5E工作以更好地理解一个产品的可用性需求。

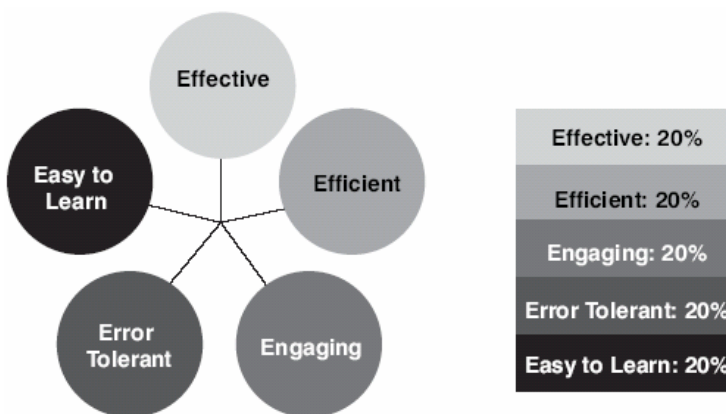


图1 这个例子中，5E是平衡的

在5E中平衡可以确定界面设计的方向。理解可用性的这些方面，换句话说，是理解可用性依赖哪些内容的开始。在表1中，我们可以看到一个利润管理业务的两

类用户—一个雇员和一个利润专家—有不同的需要。对于两个用户，有效性都需要，但利润专家的需求更多地放在效率上。对于雇员，效率与易学、容错和软件吸引人的程度相比，要被放在一个次要的地位上。

易用性需求价值思考，在细化时进行考虑，要比简单理解用户的利益走得更远。它可以作为项目管理的一个工具，帮助选择项目过程中所用于用户研究和可用性评估技术。它可以在必要时进行平衡，给出设计方法和辨别地方。这个在设计 and 开发过程中体现其价值。

### 一个以用户为中心的步骤

大多数UCD过程都符合ISO13407的通用纲要：在交互系统中以人为中心的设计过程。他们主要以一个发现过程开始，然后在研究-设计/原型-评估步骤中循环直到项目完成。整个项目创建一个设计步骤，并且测试确保任务的结构和组织是正确的。然后每一个功能北设计，用同样的迭代进行分

析、设计和评估，直到所有的功能都被集成到整个设计架构。最后可用性测试提供了软件发布前的最后结果的检查。这些过程在整个生命周期都可以被应用，将可用性和交互设计工作嵌入过程的所有阶段，从最初的产品概念到整个开发周期（见图2）。

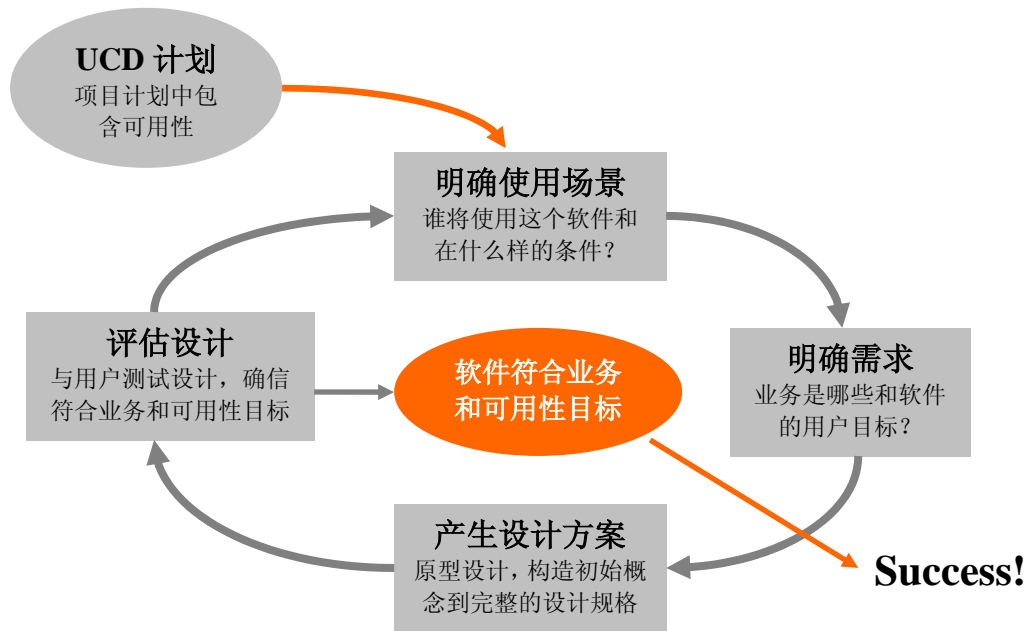


表2 - 以用户为中心的生命周期

当然，UCD工作的强度是不一样的-在设计阶段高，在实现阶段低。在每一个阶段，通过寻找会影响产品设计和构建的理解和回答询问间的差距，团队必须决定哪些UCD活动可以最好支持产品。

### 依表进行设想

我们不是每一个新项目都是一块白纸。无论项目是一个已有程序的新版本还是完全一个新产品，团队中人们一般都会在工业或商业领域中工作过。他们会成功，也有问题（或完全直接的失败），而且他们都会带来设想和基于过去经验的信念到新产品中。根据表中获得这个信息，而且因为团队需要给出愿景，构建一个用户和可用性需求的蓝图。这个是针对从用户研究获得的、可以被测量的新信息形成的基线：它增强或相背与流行的理解？如果一个有关用户和全景的更好理解可以带来一个新蓝图，这可以修改设计假定，甚至开发方法。

<p><b>雇员</b></p> <p>公司的所有雇员，必须在他们的个人信息或利益选择中使用利润管理业务。他们并不经常使用这个应用。在这个业务应用之前，他们通常要通过访问HR部门和在一个利润专家帮助下填写相关内容来进行一些修改。他们经常不确定一些选项，而且担心做某事会“搞乱”他们的信用。他们需要：</p> <ul style="list-style-type: none"> <li>• 好的指令，以替代个人访问（帮助，体现易学）</li> <li>• 确认不仅他们的数据更新，而且和这些变更的相关利益（容错）</li> <li>• 在确保整个过程中和确信在他们的入口中是精确的（吸引人的和有效的）。</li> </ul>	<p><b>利润呼叫中心专家</b></p> <p>这个公司有一个呼叫中心，利润专家可以在此处支持有困难和问题雇员。他们也完成一些雇员自己完成不了的一些过程。他们在业务中心每天工作，经常重复回答同样的问题，而且大多数呼叫下使用同一界面。他们不仅在提供的利润中，而且也在业务使用中被培训。他们需要：</p> <ul style="list-style-type: none"> <li>• 快速完成日常功能（效率）</li> <li>• 在一个简单界面中看到他们工作相伴的雇员，这样可以集中在会话中，而不是界面（吸引人的/效率）</li> <li>• 在完成之前，可以确认和雇员相关的所有变更（有效性）</li> </ul>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

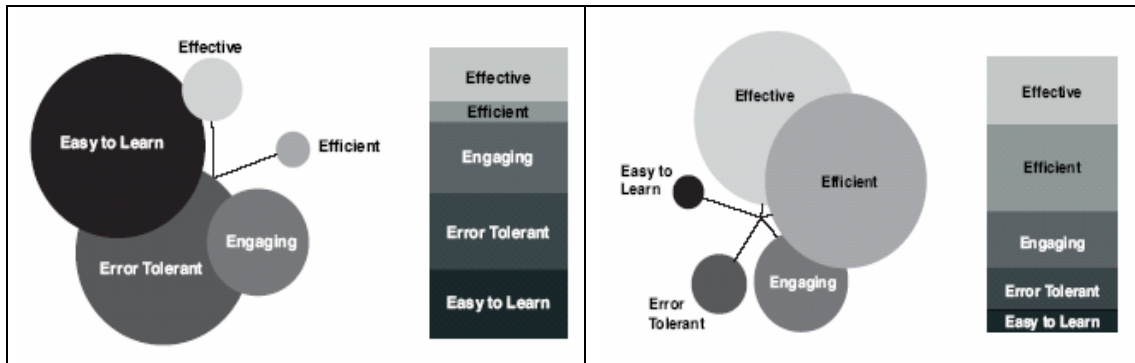


表1 不同的用户，不同的可用性需求

### 从用户中获得用户的知识

有很多技术用于获得用户的知识，每个可以帮助我们发现用户体验的不同方面。如果我们关注效率，我们将使用一个技术可以使我们看到真实的人们在实际的环境中完成实际任务；对于容错，我们可以想使用关键事件分析和与报告或记录的实际错误相比较。使用可用性的不同方面作为选择研究技术的一个工具，可以帮助确信我们获得了我们问题的答案和信息，有助于我们做出好的设计选择。

当用户研究和分析完成时，我们有一个机会比较我们团队的最初观点和用户的新理解。这是一个更新蓝图和修正任何被证明是不正确的假定的一个机会。

可用性文献充满了由于用户形象变化发生产品革新的例子。例如，我曾经工作在一个关于薪水册和雇员管理程序的小业务。当我们开始设计，我们被告知典型用户与很多不同的程序工作，对微软办公软件很熟悉，定期使用email，而且喜欢学习使用程序以改进他们的业务。产品开发团队建议用户最关注的可用性需求是效率，这样他们可以很快处理他们的工资；有效性或精确性，可以作为第二重要的需求。

当我们开始和用户工作时，我们很快发现，这个描述多么不准确。这些小业务办公室经理只与一个或两个程序工作，经常使用行业内的专业软件-而且他们想这个为很多不同的工具。他们，很少使用通用办公软件，从微软或其他供应商，而且他们工作中不用email(虽然很多有个人帐户)。最重要的是，他们只感兴趣获取足够的学习内容；他们想完成他们的工资帐目，不想改变他们业务的方式。在这个项目中，我们开始用户研究，关注速度和精确性，并试图了解用户如何完成创建工资帐目的特定任务，但我们发现这是个错误的方法。反而，我们更改我们的技术，关注易学性和容错。我们想知道，例如，在工资帐目中，软件需要培训过程中的哪些内容，哪些困难经常会导致错误。

表2 - 5E和可能的设计方法

维度	用户需求	可能的设计方法
有效性	精确性	<ul style="list-style-type: none"> <li>● 提供所有关键活动的反馈</li> <li>● 消除错误机会</li> <li>● 为用户决策提供充足的信息</li> </ul>
效率	操作速度	<ul style="list-style-type: none"> <li>● 为理想的工作流设计导航，也同时兼容替代方案</li> <li>● 提供快捷键</li> <li>● 通过交互风格和设计徽标提升速度</li> <li>● 将界面中无关要素最小化</li> </ul>
吸引人的	被吸引住	<ul style="list-style-type: none"> <li>● 使用清晰的语言和适当的术语</li> <li>● 通过适合用户的会话水平设定一个帮助声音</li> <li>● 功能结构化以匹配用户任务</li> </ul>
容错	有效和确认	<ul style="list-style-type: none"> <li>● 将错误转化成替代路径</li> <li>● 使用控件有助于准确选择</li> <li>● 确信活动容易回溯</li> </ul>
易学	及时的信息	<ul style="list-style-type: none"> <li>● 通过最少的快捷键和说明使界面有帮助</li> <li>● 针对困难或不常用任务创建引导界面</li> </ul>

表3 - 5E和可能的评估技术

维度	可能的评估技术
有效性	<ul style="list-style-type: none"> <li>● 针对困难或模糊的任务创建场景</li> <li>● 评估任务的成功完成的程度和产生没有测量到的错误频率</li> </ul>
效率	<ul style="list-style-type: none"> <li>● 创建一个实际工作节奏典型任务,充分重复地进行测试</li> <li>● 使用工作软件或高保真原型</li> <li>● 在用户工作时进行观察,发现打断他们或减缓他们的情况</li> <li>● 收集计时数据,但同样掌握参加用户关于程序的主观印象</li> </ul>
吸引人的	<ul style="list-style-type: none"> <li>● 使用满意度访谈问题或调查作为评估的一部分</li> <li>● 对现有设计测试,进行有缺点比较</li> <li>● 测试使参加用户能在他们想放弃时放弃一个任务</li> </ul>
容错	<ul style="list-style-type: none"> <li>● 创建容易犯错误或其他问题的场景</li> <li>● 观察用户能够从所发生问题中恢复过来的容易程度或精确度</li> </ul>
易学	<ul style="list-style-type: none"> <li>● 掌握提供给测试用户的指令数量,或招募不同经验或知识的参加用户</li> <li>● 混合较少用的功能到常用任务中或不常发生的变化到任务中</li> </ul>

### 创建可用性目标和需求

5E中的每一个都可以作为可用性目标的依据。一个用户的表述如“我怎么知道每个人是否会收到他们下一张支票的正确利息”，可以引出一个需求为用户能看到并在最后活动之前确认所有选择。或一个有许多不常用任务的程序可有一个可用性目标：对于一个（典型、培训过的）用户可以在没有额外培训或使用一个外部手册完成这样的任务。

无论这个表述可以变成一个功能需求或可用性目标，将他们中每一个和可用性维度相联系，使最初会话的表述和从中产生的共享愿景联系起来。例如，一个经理可以关注工作被高效完成，认为一个任务的时间问题，同时，工人却看它为一个容错问题和他们工作时业务支持的程度。

### 形成一个设计方法

关注可用性的错误方面是不可用产品的一个常见原因。因此，设计方法应当从可用性需求开始或被裁减。例如，一些用户需要快捷键或方法来一次处理一个以上的数据记录吗？或不经常用的用户需要内建的支持来提醒他们如何使用界面？5E中的每个提示了一些可能的设计需求（见表2）。

### 可用性测试计划

需要什么样的可用性评估以确信设计符合可用性目标？需要什么样的原型以获得有用的结果？与用户研究相同，答案在于我们所感兴趣的维度（见表3）。例如，一个业务需要支持非常高效的的操作，很可能需要在一些初始的培训和一套与实际任务相吻合典型工作条件下用一个高保真原型或程序的早期版本进行测试。为了测试产品在复杂任务中的吸引程度，与早期概念原型一起工作会帮助聚焦在整个过程，而不是特定的细节。

### 在计划中引入可用性

可用性或以用户为中心的设计的大多数异议之一是实践问题。如何将所有的额外工作融入已经安排得很紧、有关键点约束的计划？让我们将这个问题变换一个角度，问一个不同的问题：可用性如何帮助治愈那些困扰我们软件开发的过程？

不仅仅是开发软件过程难以避免。实际上，当你问开发人员他们工作中最恨什么？他们将告诉你：

- 需求发生了变化、变化和变化
- 客户或业务分析家不能理解你用软件能做的和不能做的
- 构建某物，结果被告知总之它不是用户（或市场）想要的。

有趣的是，这些在咨询机构Standish Group 1994年的*混乱报告*中已经包含：34%的项目在完成之前被取消，50%是实现最初设想部分内容的版本，只有16%成功[4]。而且在项目的最初阶段就失败存在以下失败原因：

- 缺乏用户输入（12.8%）
- 不完整的需求和规格（12.3%）
- 变更需求和规格（11.8%）

从这些发现看到，很多需求的变化出现是因为需求没有开始被很好地理解。这样，没有提前做

用户研究导致了缺陷和不可管理的项目。

令人感兴趣地，这些是可用性专家和用户界面设计者抱怨的主要问题：

- 软件不能考虑用户的实际工作、任务和环境
- 太多重点放在技术需求，而没有充分平衡用户需求
- 在他们有了任何实际影响的很久以后，设计和可用性才被加入产品

换另外一句话，经理、开发人员和设计者有所有标识出的相似的“痛苦点”。这个建议不仅是我们看到比想到的问题有更多的相似处，而且他们都来自同一个源泉。我们需要针对功能和用户需求的共同语言，和整个开发过程中可用性评估反馈在我们的工作中一样。没有这个共同的基础，我们不能描述准确产品，也无法在已经成功创建时认可它。

## 结论

如果你想创建一个能用和有用的产品，使用它的人们的知识和理解必须被构建到概念和架构中。引用一个喜欢的可用性谚语：“可用性不是某些东西如花生油一样可以在项目最后抹在上面”。在今天软件开发的混乱世界中，很容易抛弃新思想；他们增加复杂性，可以看到减少开发过程的控制。无论如何，如果集成得很好，以用户为中心的设计不仅能帮助创建更好的产品，而且可以减少奉献和重复。当产品设计和理解用户需求结合，可以创造更多的机会符合那些的需求。

## 参考

1. ISO/IEC. *9241-11 Ergonomic Requirements for Office Work with Visual Display Terminals (VDT) s - Part 11 Guidance on Usability*. 1998: ISO/IEC 9241-11: 1998 (E)。
2. Lidwell, W., K. Holden, and J. Butler. *Universal Principles of Design*. Rockport Publishers, 2003。
3. Nielsen, J. *Usability Engineering*. Academic Press, 1993。
4. Standish Group. *The CHAOS Report*. Standish Group International, 1994 ([www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php))。

*Whitney Quesenbery* 是一位用户界面设计师、设计过程顾问和可用性专家。

她致力于发展产品设计中的新概念，并且开发了很多获奖的多媒体产品，网站，以及网络和软件应用产品。

*Whitney*领导着自己的设计咨询公司*Whitney Interactive Design*，是公司的首席顾问。可以从网站[www.wqusability.com](http://www.wqusability.com)了解更多信息。

This article was originally published in Cutter IT Journal - February 2004, pp 4-11 and republished in **Software Usability: Strategies for Meeting Business Goals and Users' Needs**, A Cutter Consortium Report. Translation by a Chinese friend.