

3.1 引言

想画好一幅画需要很多步骤,例如,先想清楚一幅画整体给人的效果,再勾勒出大体的线条,最后进行细节的补充和色彩的填充。新的交互产品的开发与绘画一样,是由一系列基本开发过程构成的。想科学地进行交互式系统的开发,需要了解交互设计过程的基本活动和关键特征,真正做到以用户为中心进行开发,准备候选设计方案,等等。

同时,如果开发者具有一定的软件开发经验,应当已经对术语“生命周期”有一定的了解。与传统软件的生命周期模型类似,交互设计领域的生命周期模型也能合理地指导人机交互系统的设计与实现。生命周期模型体现了所涉及的各种活动之间的关系,复杂模型的建立还可以辅以文字说明,包括但不限于在何种情况下,系统应该怎样操作、每个活动的输入输出等。而一旦设计者真正学会用生命周期模型指导开发过程,会发现自己从宏观上把握开发过程、分配资源、进行更新和修正的能力有所提升。根据过往的设计经验来看,每个设计者平均有 50% 的时间花费在用户界面与交互的编码设计上,因此,理解掌握交互式设计过程和技术是必要的,本章将展开对上述过程和技术的讨论。

本章的主要内容包括:

- 介绍人机交互设计的活动和特征。
- 理解交互设计领域的生命周期模型。
- 学习通过框架科学地进行交互设计。
- 了解如何对已有设计进行测试和改进。

3.2 人机交互设计的基本活动

3.2.1 4 项基本活动

交互设计的基本活动关注用户与系统之间的交互过程。人机交互设计有以下 4 项基本活动,如图 3.1 所示。

1. 标识用户需要并建立需求

了解了目标用户和用户的主要需求,才能明白交互产品应该提供哪些支持,怎样开发出更符合用户需求的系统产品。在以用户为中心的方法中,想了解用户的主要需求就需要做好需求获取,可以采用访谈法、调查问卷法、联合小组分析等方法。假设一个设计团队正在

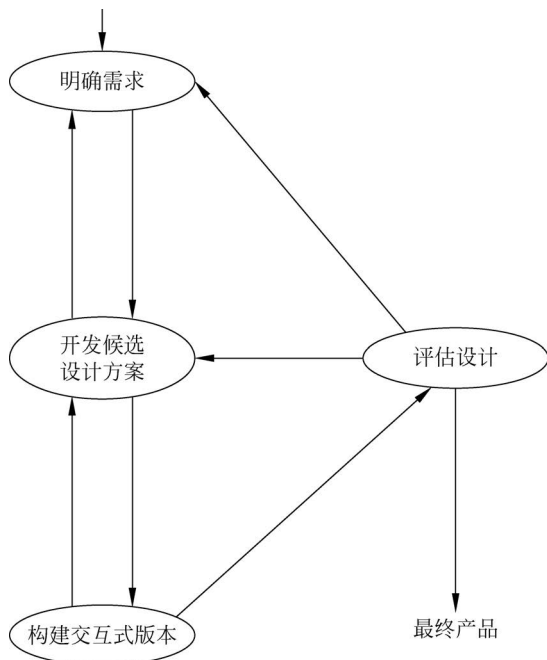


图 3.1 人机交互设计的 4 项基本活动

开发一个电子商务网站。他们可以进行用户调研,通过问卷调查和用户访谈来了解用户的购物习惯、偏好和当前需求。他们还可以观察用户在其他类似网站上的行为,以获取更多的参考信息。除此之外,设计团队还可以通过编写用户故事和用户场景来进一步明确需求。用户故事和用户场景用来描述用户在特定情境下的需求和期望。用户故事较为简短,用户场景则是更详细的情境描述,在其中可以包括用户的目标、行为和期望结果。

2. 开发满足需要的设计候选方案

在确认好需求后,可以考虑符合当前要求的设计方法,并可根据具体需要设计多个方案进行选择 and 调整。这一核心活动包括两个子活动:概念设计和物理设计。概念设计致力于描述产品应该做什么、如何操作,同时会对产品的外观等进行介绍。在物理设计中,可以具体写明产品实现的部分细节,如要使用的色彩、图像、图标、交互行为等内容。

3. 构建设计的交互式版本

设计者在长期工作中发现,让用户与产品进行交互,能更好地发现问题,进行评估。但实际能运行的脚本语言和高保真原型设计的成本又太高,在此时就可以选择线框图、草图等保真程度不高的演示形式。在本阶段的活动中,目的不是提供可运行的程序,而是大致地演示流程,并在其中发现问题。

4. 评估设计

在交互设计领域及其他众多设计领域,大量的精力都被投入评估设计中。作为设计中的重要一环,评估设计用来降低系统的错误率,提高系统的可操作性和可用性,包括发现在使用时出现的各种错误,如输入输出错误、显示错误、跳转路径错误等。也可以考虑对满足需求程度低的功能模块进行修改,让系统更吸引人,提高用户对当前系统的黏性。当评估交互系统的设计案例时,可以考虑以下几个方面。

1) 用户体验

以系统的用户界面和交互流程是否直观、易用、符合用户期望为标准,通过进行用户测试、问卷调查或使用专家评审等方法来收集用户反馈。

2) 功能完整性

评估系统是否提供了所需的功能,并且这些功能是否能够满足用户的需求,可以通过功能测试、功能需求分析等方法来评估。

3) 性能和效率

交互产品的性能和效率也是影响使用体验的重要一环。性能和效率包括响应时间、加载速度、数据处理能力等,可以通过性能测试、负载测试等方法来评估。

4) 安全性和隐私保护

评估系统的安全性和隐私保护措施是否得当,以保护用户的数据和个人信息,可以通过安全性测试、隐私风险评估等方法来评估。

5) 可维护性

评估系统的可维护性包括代码结构、文档、扩展等方面。可维护性好的系统易于修改和拓展,利于产品中后期的开发,同时能提升交互产品的重复使用率。

任何交互产品的设计都包括一系列活动,由一系列流程组成,包括明确需求、开发候选设计方案、构建交互式设计版本、进行评估设计等步骤。这些步骤没有强制的顺序,而是在系统设计过程中不断进行重复,以最大限度地提升系统的可用性。这是一种通用的交互设计生命周期模型,易于学习和掌握,因而在交互设计领域有较为广泛的应用。

3.2.2 关键特征

在设计过程中,设计者应时刻把握交互设计的关键特征,才能使设计最大程度地满足项目前期获取的用户需求。三个关键特征包括以用户为中心、制定稳定的可用性标准、进行迭代。

在设计过程中,用户无法直接参与系统的开发,但设计者应当让用户尽可能多地参与到反馈和评估中。这不仅能强调用户的需求和感受在设计中的重要性,更能发现设计中偏离需求的部分,及时进行修改和调整。

此外,制定稳定的可用性标准,有利于设计人员选择当前最合适的候选方案,并在产品开发过程中及时发现问题。可用性标准应当尽量准确、具体,可以从可用性和用户体验目标的角度进行考虑。在制定标准时可以参考以下几项内容。

1. 参考行业标准和最佳实践

了解相关行业的可用性标准和最佳实践,例如,ISO 标准、Web Content Accessibility Guidelines (WCAG)等。这些标准可以提供一个基准,帮助设计者制定自己的可用性标准。

2. 确定关键指标

确定关键的可用性指标,以衡量产品或服务的用户体验。这些指标可以包括任务完成时间、错误率、用户满意度等。设计者需要确保这些指标与服务的特定目标和用户的需求相一致。

3. 设定可接受的阈值

为每个可用性指标设定可接受的阈值或目标值。这些阈值应该基于用户需求、行业标

准和最佳实践。例如,任务完成时间应该在一定范围内,错误率应该低于某个百分比等。

4. 进行用户测试和评估

可以通过用户测试和评估来验证和调整可用性标准。收集用户的反馈和数据,了解用户的体验和需求,并根据实际情况对可用性标准进行修订和改进。

5. 持续监测和改进

持续监测产品或服务的可用性指标,并进行定期的评估和改进。

没有设计者能一次性设计出不存在问题的系统,因此迭代是不可避免的。针对设计过程中已经发现的问题,设计者应对交互系统进行迭代。例如,创建的想要解决某个问题的思路并不总是十分完善且可以实施的,它需要通过不断反复地发现问题、进行迭代来发现问题的最优解决路径和形式。成熟的设计者应该做到不逃避必要的修改。

现在以一个电子商务网站的购物车设计为例,说明如何开发一个符合交互设计关键特征的界面。购物车界面应该简单明了,用户可以轻松地将商品添加到购物车中,并查看已添加的商品列表。购物车界面应该有明确的按钮和标签,让用户能够直观地执行添加、删除和修改商品数量等操作。在视觉上,购物车界面应该与整个网站的设计风格保持一致,包括颜色、字体和布局等方面。用户在不同页面之间切换时,购物车的界面应该保持一致,让用户感到熟悉和舒适。当用户将商品添加到购物车或执行其他操作时,购物车界面应该及时给予反馈,让用户知道操作是否成功。例如,添加商品后,购物车图标上显示添加的商品数量,或者弹出提示框显示成功添加的信息。不仅如此,购物车界面应该考虑到不同用户的需求,包括残障人士。例如,提供辅助功能,如屏幕阅读器支持键盘导航,以确保所有用户都能够方便地使用它。购物车的功能也应该具备可扩展性,能够适应未来的需求和变化。例如,允许用户添加多个商品到购物车,支持优惠券和促销活动等功能的扩展。此外,购物车界面应该提供适当的引导和帮助,帮助用户了解如何使用购物车功能。例如,提供购物车使用指南、常见问题解答和联系客服的方式,以解决用户可能遇到的问题。将问题考虑全面后,设计的购物车界面将能提供非常好的用户体验

3.2.3 以用户为中心

华为的任正非曾经说过,“我们要建立一系列以客户为中心、以生存为底线的管理体系,而不是依赖于企业家个人的决策制度。在这个管理体系规范运作时,企业之魂就不再是企业家,而变成了客户需求。牢记客户永远是企业之魂。”这也是华为成功的奥秘所在。

事实上,不管是企业文化还是产品设计,只有真正做到“以用户为中心”,才能受到用户支持,有蓬勃的生命力。

亚马逊真正把客户至上作为最高准则,围绕“客户体验”进行战略的制定和业务的发展,给客户id提供绝佳的综合体验。其创始人贝索斯在开会的时候经常会在他的旁边放一把空椅子,这把空椅子并不为某位特邀嘉宾或者部门主管设置,而是专为消费者设计。在亚马逊的内部讨论中,贝索斯告诉经理们:如果工作有分歧,最终的评判标准是从“客户的角度出发”,有利于客户的一方胜出。不仅如此,贝索斯在给股东的一封信中曾写道:在亚马逊刚刚成立的时候,我们就给用户开通了评论商品的功能,结果就收到了一些供应商的抱怨:这样做有没有考虑过对自身的影响,商品有了负面的评论,不会影响销售和营利吗?亚马逊也做过一些市场调研,通过很多用户的反馈,的确负面的消息会影响用户对商品的购买。但贝

索斯说,“我们不是靠卖东西赚钱的,我们是靠帮用户做好的消费决策来赚钱的”。商品的负面评论虽然意味着亚马逊要承受更多压力与质疑,但毫无疑问的是,这一决策体现了“以用户为中心”的思想,在经受时间的考验后成为广受用户称赞的一项制度并为亚马逊带来了更多的收益。

人机交互领域的以用户为中心是一种设计和开发技术的理念,旨在将用户的需求、行为和体验置于首位,以提供更好的交互体验来满足用户的期望。在这种交互方式中,设计师和开发者将关注用户的需求、习惯、能力和偏好,并将这些因素融入系统的设计和函数中。

在人机交互领域,如果想真正做到“以用户为中心”,需要考虑如下内容。

1. 用户需求分析

通过调研和用户研究,了解用户的需求、目标和期望,以便设计出符合用户期望的系统。在进行分析时应当紧密与用户结合,除了运用严谨的需求获取方法如访谈法、调查问卷法等,还可以构建场景和人物角色,这有助于设计师更好地理解目标用户,并在设计过程中保持用户的视角。

创建场景首先需要确定设计产品服务的目标用户群体,这可以是特定的人群、用户类型。然后设计者需要选择一个具体的使用场景或情境,这个场景应该与产品或服务密切相关。例如,如果正在设计一个旅行预订应用,场景可以是用户计划一次旅行。接着要为场景中的用户角色创建一个简要的描述,参考后文即将介绍的人物角色,这有助于设计者更好地理解用户的行为和期望。详细描述用户在该场景中的行为流程,包括他们的起始点、目标和所需的步骤。例如,在旅行预订应用的场景中,用户可能会搜索目的地、选择日期、选择酒店、预订机票等。了解了上述需求就可以开始设计相应的解决方案。这包括界面设计、功能设计、交互设计等,以满足用户在该场景中的需求。最后将设计方案应用于实际场景中,并进行用户测试和反馈收集。根据用户的反馈和需求,不断迭代和改进设计。

通过创建场景,设计者可以更好地理解用户在特定情境中的需求和行为,从而设计出更符合用户期望的解决方案。这有助于确保设计出的产品或服务能够满足用户的实际需求,并提供良好的用户体验。

创建人物角色需要描述用户的特征、目标、需求和行为。为了达到这个目标,设计者必须获取角色的一些信息,以此为参考来推断他们如何做出决策。对于规模不太大的交互系统可以构建一个主要人物角色、一个次要人物角色。在构建时可以考虑:影响人物角色愿望的态度、经历、渴望,以及他对社会、文化、环境的认知因素。通过这些推断人物角色在与产品交互时对产品行为的期待和愿望。另外,设计者需要构建人物角色与当前界面进行交互的心智模型,这有利于在设计过程中为数据选择合适的组织形式。

需要注意的是,了解并迎合用户需求并不意味着无条件地满足用户的要求,要始终把握产品的目标来判断用户的提议是否可以采纳。

2. 用户界面设计

用户界面设计是指设计者通过布局、视觉元素、交互方式等手段,为用户提供友好、直观、高效的界面,使用户能够轻松地与软件、应用程序或网站进行交互。

一个好的页面设计应当具有清晰的布局、对重要内容的强调、一致的风格、简化的输入和易于理解的反馈等。

在具体设计时可以从以下几个方面入手。例如,设计一个直观和易于导航的布局,使用

户能够快速找到他们需要的功能和信息；使用明确的标题、分组和层次结构来组织内容；使用色彩、大小和对比度等视觉元素来突出显示重要的功能和信息，这可以帮助用户快速识别和理解界面的关键部分；确保整个界面在视觉上保持一致，包括颜色、字体、图标和按钮样式等，这可以帮助用户建立对界面的熟悉感，并提供一致的用户体验；尽量减少用户的输入工作量，通过使用自动填充、下拉菜单和默认选项等方式来简化表单及输入过程；为用户提供明确和及时的反馈，以确保他们知道他们的操作是否成功，这可以包括弹出消息、状态指示器和动画效果等。

总的来讲，为了使页面易于使用，用户能够快速理解和操作，设计师需要考虑用户的需求和行为模式，确保界面的布局和导航方式符合用户的直觉；界面的各个部分应该保持一致，包括颜色、字体、图标等，以提高用户的学习效率，使用户更容易理解和使用界面；界面上的元素应该清晰可见，用户能够快速找到所需的功能和信息。

3. 用户参与和个性化

设计者应该鼓励用户参与界面的设计过程。这可以通过用户调研、用户测试和用户反馈等方式实现。用户参与可以帮助设计者了解用户需求和偏好，从而更好地满足用户的期望。在参与的同时要为用户提供反馈机制，让用户知道他们的操作是否成功，以及如何纠正错误。例如，当用户提交表单时，显示一个成功的提示消息或错误的提示消息，帮助用户了解他们的操作结果。

设计者还应为用户提供个性化的界面体验，根据用户的偏好和习惯进行定制。这可以通过用户配置选项、主题选择、语言选择等方式来实现。

4. 用户体验评估

通过用户调研、用户测试等方法，评估用户对系统的满意度和体验，并根据反馈不断改进和优化系统。

通过完成上述工作，交互产品能提供用户友好的交互方式，使用户能够更轻松、高效地使用系统，并获得满意的体验。也就是说，关注用户需求和体验，可以提高系统的可用性、用户满意度和用户忠诚度。

3.2.4 候选设计方案

想要得到合适的候选方案，需要先进行设计，再对已有的设计进行筛选和修改。设计方案主要来源于三个途径：其他相似的设计，设计人员的积累以及其他人的构思。

在设计时可以参考其他类似产品的构思，尤其是对于已经获得大众认可的产品的的设计。使用与之前已有作品相似设计的优点在于：相似页面的页面布局和处理逻辑比较固定，这也意味着用户在使用时将花费更少的时间成本，更容易从新手用户变为稳定的中间用户，也就更容易对当前的交互产品产生信任和依赖。已有设计的另一优点是：经受过时间的考验。留存下来可以被看到的必然是以往很多个版本的设计中相对优秀的，是符合用户的期望的。从一定程度上可以说，已经获得广泛推广的交互产品都是“不太差”的。但不太差并不等于最优，这也就需要有新鲜的想法和创造，不断对产品的设计进行丰富和升级。

总有人能在其他人还在冥思苦想、不得其要点的时候就提出非常好的想法，业界也不否认这些人的存在，鼓励设计人员在交互的各个方面进行创新。在创新后，可以邀请其他设计

人员和涉众对新的设计进行评价和测试。如果能获得认可,不但可以为当前交互产品增添新意,也能推动整个交互设计领域的发展和变革。

当然,仅有创意是不够的,设计是严谨的、有约束的,不能随意将任何想法都盲目地添加到其中,而需要遵守一定的规则。例如,在系统的设计中,不仅需要保证每一处设计的风格一致,在面向用户时,还需要做到使产品的各个部分保持一致。

针对前期设计出的方案,需要进行筛选,即从两类指标入手来对方案进行设计决策。一是可以被直观感知和测量的特征,称其为外部特征;二是不可以被直观感知的,需要对系统进行解构和分析才可见的特征,称其为内部特征。

在筛选时,要继续遵循以用户为中心的设计原则,尊重用户需求,同时考虑技术的可行性。例如,在考虑文件保存功能时,设计者可以思考以下几个问题:是否只能手动对文档进行保存;是否为文档提供自动保存功能;如果提供自动保存的功能,时间间隔又该设定为多少。

交互式产品的外部因素可以通俗地理解为直接面向用户的设计,所以外部因素是否有优良的性能是筛选交互产品候选方案的重要标准。例如,打开一个网站,加载页面中的信息需要10s,后台的工作可能是对各种各样的图片信息进行缓冲,对大量数据进行处理。但在用户角度,关注的仅仅是此网站加载时间为10s,所以设计者可以把开发设计的时间精力适当集中在可见、可测量的外部行为上,让用户与产品的交互推动产品的设计和开发的进行。在选择候选方案时,仍然需要让产品与涉众交互,并听取用户在体验过程中的感受和意见。所以在表示设计的过程中,应当尽量采用易于用户理解的方式,尽量使用通俗直观的语言,避免专业性术语等用户无法理解的语言。一个应用广泛的表示设计的方法是原型,原型的直观是语言描述无法相比的。

另一个选择和衡量候选设计方案的重要标准是交互产品的质量。人们在使用交互系统时,通常不会有成文的标准规定某项功能的执行标准,但却在有意识无意识间将不符合自己期望的产品淘汰。例如,在网络状况相同时,一个视频网站可以清晰流畅地播放,而另一个却出现了不清晰或频繁卡顿的情况。又例如,某个设备连接外设时只需打开功能键,而另一个设备在连接外设时需要使用复杂的组合键,人们会理所应当地选择前者。

3.2.5 交互设计生命周期

软件工程生命周期和交互设计生命周期都包含一个产品从开始研制到最终被废弃的全部流程,二者互为参考。同时,软件工程生命周期中的设计和开发阶段与交互设计生命周期密切相关,交互设计团队需要与开发团队紧密合作,确保设计的可行性。所以,一个优秀的交互设计师不仅应当深入学习交互设计生命周期,对软件工程领域常见的生命周期模型也应有所了解。

1. 软件工程生命周期

一个软件产品或软件系统要经历孕育、诞生、成熟、衰亡等阶段。软件生命周期是指人们为更好地开发软件而归纳总结的软件生命周期的典型实践参考。一般来说,软件生命周期包括以下几个阶段。

1) 需求分析阶段

在这个阶段,开发团队与客户合作,收集和理解用户需求,确定软件系统的功能和特性。

2) 设计阶段

在需求分析的基础上,开发团队开始设计软件系统的架构和组件。这包括确定系统的模块划分,选择数据结构,进行算法、界面设计等。

3) 编码阶段

在设计完成后,开发团队开始实现软件系统的代码。开发人员将设计文档转换为可执行的软件程序,并进行单元测试以验证代码的正确性。

4) 测试阶段

在编码完成后,软件系统需要进行更全面的测试,包括单元测试、集成测试和系统测试等,以发现和修复潜在的问题和错误。

5) 部署和维护阶段

在测试通过后,软件系统可以部署到生产环境中供用户使用。同时,开发团队需要继续监控和维护系统,及时修复漏洞和错误,并根据用户反馈进行功能的更新和改进。

这些阶段通常被认为是软件开发的基本生命周期,不同的开发方法和模型可能会有一些变化和调整,但总体上遵循类似的过程。软件生命周期的目标是确保软件开发过程的可控性和质量,以提供满足用户需求的可靠软件系统。常见的软件工程领域的生命周期模型有瀑布模型、螺旋模型、快速应用开发、原型法等。

(1) 瀑布模型。

瀑布模型是软件开发中最经典的开发模型之一,它以线性顺序推进软件的开发。模型包括需求分析、设计、编码、测试和维护等阶段。

在瀑布模型中,每个阶段都是严格按照顺序进行的,前一阶段的输出作为后一阶段的输入。首先,需求分析阶段通过与客户沟通和收集需求,明确软件系统的功能和特性。然后,设计阶段根据需求分析的结果,制定软件系统的架构和组件设计。接下来,编码阶段将设计文档转换为可执行的代码,并进行单元测试。测试阶段对编码完成的软件进行各种测试,包括单元测试、集成测试和系统测试,以确保软件的质量和稳定性。最后,部署和维护阶段将测试通过的软件部署到生产环境中,同时继续监控和维护系统,修复错误和漏洞,并根据用户反馈进行功能的更新和改进。

瀑布模型的优点是结构清晰,易于理解和管理,适用于需求变化较少的项目,其结构如图 3.2 所示。然而,它也存在缺点,如刚性的阶段顺序导致需求变更困难等。因此,在实际开发中,可以根据项目需求选择适合的开发模型,如敏捷开发等。

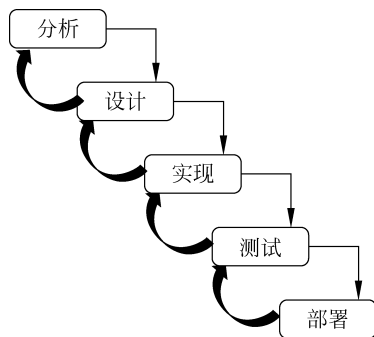


图 3.2 瀑布模型

(2) 螺旋模型。

螺旋模型是一种迭代式的软件开发模型,它结合了瀑布模型和迭代开发的特点,如图 3.3 所示。螺旋模型强调风险管理和迭代开发的概念,适用于大型、复杂且风险较高的项目。

螺旋模型的开发过程由多个循环组成,每个循环包括 4 个主要阶段:计划、风险分析、工程开发和评估。在计划阶段,确定项目目标、约束条件和可行性分析。在风险分析阶段,评估项目的风险,并制定相应的风险管理策略。在工程开发阶段,根据计划和风险分析的结

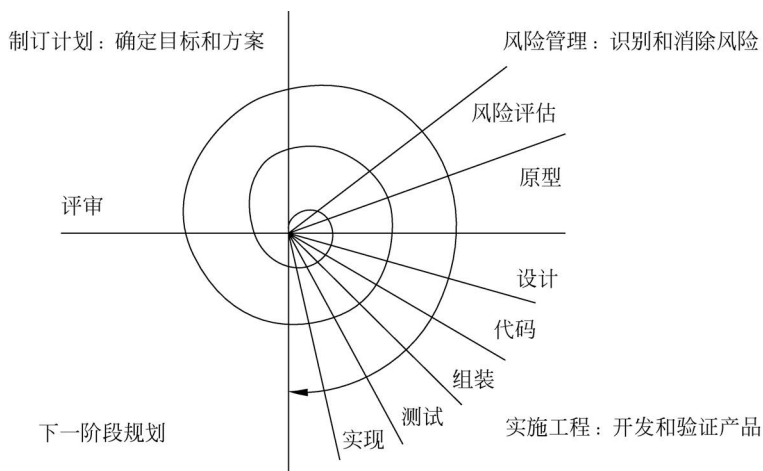


图 3.3 螺旋模型

果,进行需求分析、设计、编码和测试等开发活动。在评估阶段,评估前一个循环的结果,并决定是否继续下一个循环。

螺旋模型的优点是强调风险管理,能够及时识别和解决问题,降低项目失败的风险。同时,迭代开发的特点使得项目可以根据用户反馈和需求变化进行调整和改进。然而,螺旋模型也存在一些挑战,如需要更多的资源和时间、管理复杂度较高等。

总的来说,螺旋模型适用于大型、复杂且风险较高的项目,可以帮助团队更好地管理和控制项目开发过程。

(3) 快速应用开发。

快速应用开发(Rational Application Develop,RAD)是一种软件开发方法,旨在通过迅速构建原型和迭代开发来快速交付软件应用程序,如图 3.4 所示。RAD 的目标是加快开发周期、提高开发效率和灵活性。RAD 的关键特点包括以下几点。

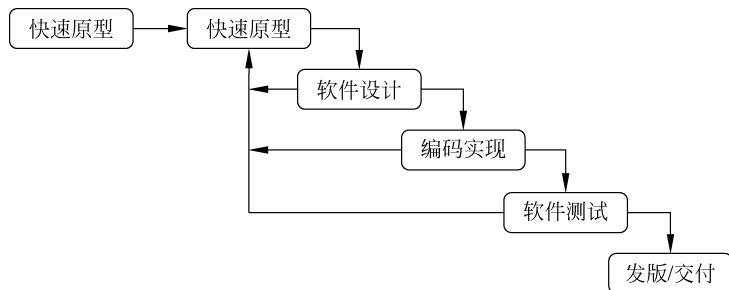


图 3.4 快速应用开发

① 快速原型开发。RAD 强调快速构建原型,以便快速获取用户反馈和验证需求。原型可以是简化的、功能有限的版本,用于演示和验证设计概念。

② 迭代开发。RAD 采用迭代的方式进行开发,每个迭代周期都会增加新的功能和特性。通过迭代,开发团队可以快速响应需求变化,并逐步完善软件系统。

③ 可视化开发工具。RAD 使用可视化开发工具和技术,如图形化界面设计工具、代码生成器等,以提高开发效率和降低编码工作量。

④ 高度协作。RAD 鼓励开发团队和利益相关者之间的紧密合作和沟通,以确保需求的准确理解和及时反馈。

RAD 的优点是快速交付、灵活性高和用户参与度高。它适用于需求变化频繁、时间紧迫的项目,以及小型和中型的软件开发。然而,RAD 也面临一些挑战,如需求管理和控制、可维护性和系统性能等方面的问题。

总体而言,RAD 是一种适应快速变化和迅速交付的软件开发方法,可以帮助开发团队更快速地构建和交付高质量的软件应用程序。

(4) 原型法。

随着计算机软件技术的发展,人们在 20 世纪 80 年代正式提出了原型法这个概念。它是一种全新的系统开发方法,与一步步进行分析、逐步整理出文字档案、进行整合后再让用户看到结果的方法相比,原型法相对来讲并不烦琐且非常直观,因而也被广泛地应用到设计开发的过程中。原型法能从视觉上直接将产品展示给用户,使用户可以直接参与到系统开发中,最终的产品自然更贴近用户期望,有较好的用户满意度。同时,原型法相对方便快捷,能缩短开发周期,控制开发成本。图 3.5 所示是一个视频播放平台的原型。



图 3.5 原型实例

但原型法的缺点也是固有的,例如,它不适合大规模开发,难以模拟运算量大、逻辑性强的系统,在遇到这一类系统时,使用原型法不仅会花费大量的时间精力,最终的原型也不能很好地向用户还原系统的使用体验。

2. 交互设计生命周期

1) 可用性工程生命周期

可用性工程是以提高产品可用性为目标的多学科交叉领域,综合了技术学科、心理学、工业设计、人类工效学、社会学等学科的方法,强调在产品开发过程中以用户需求作为出发

点,提高用户的参与度,以便能及时获得用户的反馈并据此反复改进设计。“可用性工程生命周期”这个概念,是将可用性工程与软件生命周期模型结合起来而提出的。它详细描述了如何执行可用性任务,并且说明了如何把可用性任务集成到传统软件开发生命周期中,如图 3.6 所示。在缺乏经验时使用快速原型开发法可以将可用性任务与传统软件的开发步骤相联系。可用性工程生命周期模型包含几个基本任务:需求分析、设计/测试/开发、安装,在中间的设计开发和测试阶段涉及较多的子过程。此模型需要首先提出一组可用性目标,并按照这些目标进行开发,以确保开发不偏离预设的轨道,防止在大部分工作都已经完成后才发现不符合标准或预期的情况。

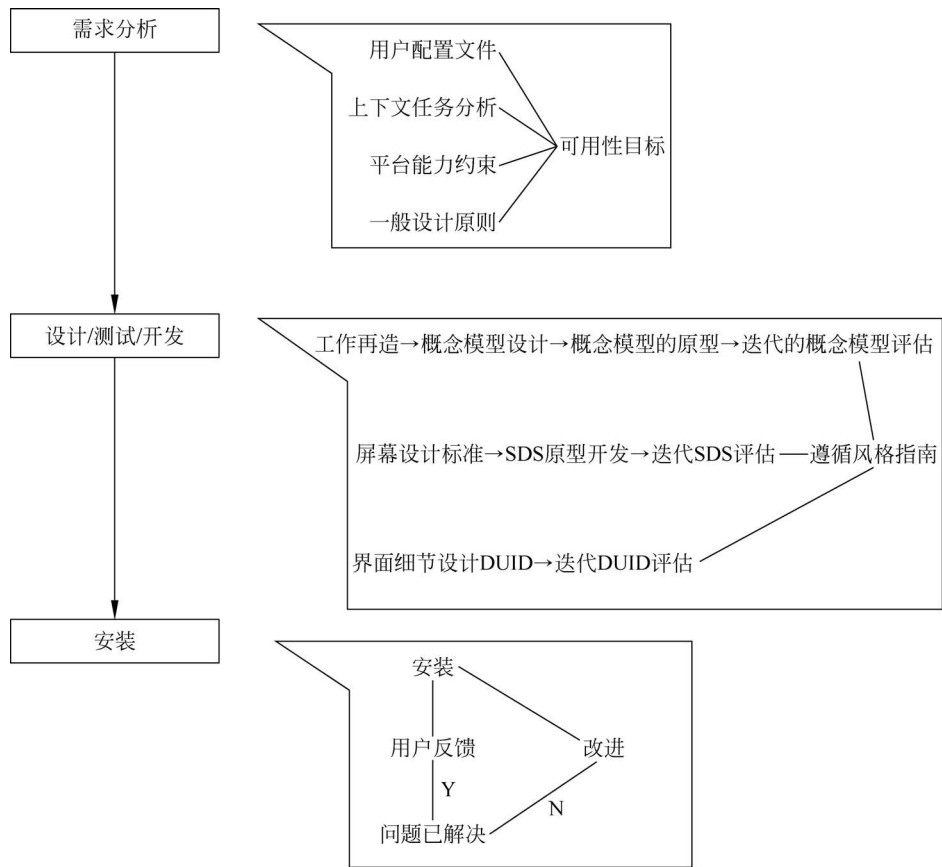


图 3.6 可用性工程生命周期

2) 星状生命周期

星状生命周期以评估为核心,同时反复进行实践、任务分析/功能分析、制作原型、需求规范、概念设计/正式设计表示等。值得注意的是,星状生命周期模型没有为任何活动指定次序,使用者可以选择合适的活动来开启设计工作,也可以自由地从一个活动切换到另一个活动,但在一项任务完成后必须对其结果进行评估之后才能进行下一项,如图 3.7 所示。一般来讲,星状生命周期有两种活动模式:分析模式和合成模式。分析模式是从系统到用户的方法,是自上而下的,在执行过程中强调正式性和规范性。合成模式是从用户到系统的方法,是自下而上的,在执行过程中强调创造性。设计人员在开发过程中通常让两种模式交替进行。

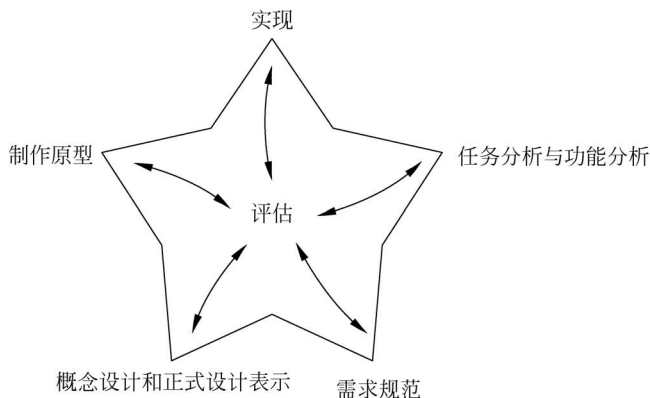


图 3.7 星状生命周期模型

3. 软件工程生命周期与交互设计生命周期

尽管两种生命周期都是人们为开发更好的产品而归纳总结的典型实践参考，软件工程生命周期与交互设计生命周期仍存在一些不同之处。软件工程生命周期的目标是开发和交付高质量的软件产品，而交互设计生命周期的目标是设计和开发用户友好的界面和交互体验；软件工程生命周期通常包括需求分析、设计、编码、测试和维护等阶段，而交互设计生命周期通常包括用户研究、需求分析、设计、评估和迭代等阶段；软件工程生命周期中的参与者通常包括开发人员、测试人员和项目经理等，而交互设计生命周期中的参与者则包含用户。

需要注意的是，交互设计领域的生命周期模型较少。已经介绍的两个模型：可用性工程生命周期模型、星状生命周期模型都以评估为核心。其中，可用性生命周期模型流程规范，星状设计模型没有指定任何活动次序。两种模型在应用时都需要把握“自下而上”分析，“自上而下”合成的原则。

3.3 人机交互设计框架

3.3.1 创建设计框架

像海报的设计一样，往往是先拥有设计的版式，再向其中填充内容。在设计的最初阶段，设计者应该先关注海报在整体上给读者带来的效果。例如，各个位置的功能、布局 and 预留的空间，而不必考虑每个位置具体的内容和情节。交互设计以目标为导向，在具体环节的实现了与其他设计类似，应从大至小进行设计而非在最初就进入细节设计。

设计框架包括交互框架和视觉设计框架、工业设计框架、服务设计框架等。交互框架要求设计者能对用户的需求和使用场景进行充分考量，并在此基础上进行屏幕和行为设计。视觉设计框架则需要利用视觉语言研究开发视觉设计框架。此外，在复杂的工业设计中，也可以提前设计工业设计框架和服务设计框架，这些框架将在开发后期填充细节时起到指导作用。通过上述方式设计的设计结构可以起到规范用户体验结构的作用，也能使系统在功能性、主要业务流程、产品信息、传递的信息、语言等方面有可以参照的标准。

了解了框架开发的基本情况，还需要处理好框架开发中抽象和细节的关系，学习与“修

改”有关的知识和技巧,理解如何通过可用性测试指导框架研究。

过于关注细节会导致设计过程中各个阶段的投入失衡,造成本末倒置和不必要的重复工作。应当采用先考虑“抽象”,再考虑细节的设计方法,自上而下地进行设计。遵循这种科学的设计顺序,不但能从初始阶段就开始以目标为导向设计,还能大大降低出错率,提升工作效率,减少开发成本。因此,越是规范的工业产业开发,框架设计显得越重要。

修改是设计的一部分,一个好的设计师应当不惧怕修改,因为修改是无法避免且有益于设计本身的。但对必要细节进行锤炼和修改,并不等于花时间打磨所有细节,其中,冗余的部分和在迭代中不断变更的细节在框架设计阶段完全可以不考虑。在不确定如何对系统进行有效的修改时,可以通过反复演示设计方案中涉及的过程,让参与其中的开发者或用户获取新的灵感,从而实现对系统的整改。此外,夹杂叙事场景的故事本草图也可以作为反复演示的替代方案,让开发和体验在场景中进行,避免了过于细致的要求对于设计的阻滞作用。

多项研究表明,在开发过程中演示图的绘制能协助开发者和涉众对系统有更好的了解,更直观生动地理解系统的功能模块,进一步明确开发者的规划。用于演示的图的保真程度的选择则依赖于当前所处的开发阶段,在初期可以选择耗时少、易修改的低保真原型图进行设计,随着设计与开发不断深入,逐步提高演示图的保真程度,如按从先到后的开发阶段分别选择草图、线框图和高保真原型图。此外,在必要时及时地进行用户反馈和可用性测试,也能大幅提高开发质量。

下面以具体的 6 个步骤来描述交互框架的定义方法。

1. 定义形式要素、姿态和输入方法

设计者可以将自己代入用户的使用场景,以更好地理解形式要素对产品设计产生的影响。例如,在地铁站的场景,则大致可以推断周围环境嘈杂,无稳定的网络信号,环境开放,对用户个人隐私保护程度低;如果场景是在办公室,则可以认为用户对于设备或系统有安静的操作环境和稳定的操作条件。在有具体场景时,设计者会很容易明白产品应当更需要哪种品质,是灵敏、快捷还是方便易携带。产品姿态是指用户将会投入多大的注意力和产品互动,以及产品的行为将会对用户投入的注意力做出何种反应。主要用户和次要用户可以通过键盘、鼠标、遥控器等与产品进行互动,这些互动称为输入方法。如果某种情况下用户的需求不同,需要使用多种输入方法进行操作,设计者应区分主要输入方法和次要输入方法。

2. 定义功能性和数据元素

数据元素代表着界面中要展示给用户的数据,如图片、声音、使用记录、账号信息等,是用户可以访问、反应、操作的基本个体。数据元素的设计过程中有几点注意事项:需要对人物模型关心的数据模型有所了解,这样可以使创建的数据模型更符合人物的心理预期,降低用户的使用难度,节约学习成本;一些显著的属性应当被重点关注,如文档的最新修改时间、文档的创建者等,这涉及产品功能的定义和使用;关注数据之间的联系也是必不可少的一个环节,如根文件夹下有子文件夹和其他文件,如何处理这些数据元素之间的结构和联系,都是需要提前考虑的地方。功能性代表着页面为用户提供的功能和交互,具体地讲,是对已定义的数据元素进行显示和操作,如视频的播放功能、文档的保存功能等,设计功能时应紧密结合需求调研的成果。通过功能元素的编写,能让场景剧本中的设想在系统中得以实现。需要注意的是,往往不能通过一个功能元素实现一个完整的需求,而需要多个功能元

素配合来完成。以发送电子邮件为例,需要的功能元素包括但不限于:使用与“联系人”有关的数据元素作为收件人或手动输入收件人邮箱号,发送电子邮件按钮,电子邮件内容选择区(可以通过外设输入,也可以导入计算机内部的数据和文件)等。

在数据元素和功能元素的设计出现问题时,可以从以下两个方面入手进行改进和设计。

1) 将交互系统想象成一位得力助手

澳大利亚南昆士兰大学研究的 Echo 被瑞典斯德哥尔摩的阿玛兰登号角酒店使用,并充当管家,该机器具有的功能能很好地协助酒店管理,如帮助客人请求客房服务、提供在线信息援助等。一个好的系统应当像 Echo 一样,能为使用该交互产品的用户带来便利。假如一位教师在使用系统,设计者就可以把系统当成一位班助,为了协助教师更好地进行工作,班助会在教师发布通知时帮忙@所有人;在教师需要统一上交材料时,班助会统计班级中没有按时上交的人的名单并反馈给教师。与此相对应,在进行系统开发时,设计者将系统想象成一位得力助手,不仅可以帮设计者厘清设计思路,还能提升用户使用系统的满意度。

2) 采用已有的设计原则和模式

这些原则和模式都是前人不断在错误中改进和提炼的,用户也已经对现有的设计模式有了自己的心智模型,贸然改动只会使设计者的设计成本和用户的学习成本增加。

3. 确定功能组和层级

确定功能组和层级是设计交互框架的重要一步,在功能元素和数据元素设计得相对比较完善时,就可以进行层级的划分和功能组的确定。最初建立这些关系的时候,可以用缩略图或韦恩图进行简单的记录,在绘制时应当尽可能采用易于修改和重做的方法,如用铅笔绘图或用可擦笔在白板上绘图。在对功能和数据元素进行分组时,可以根据产品大小、姿态、屏幕大小、输入输出方法等条件考虑如何组织和安排。例如,在设计过程中,可以思考哪些元素能够容纳其他元素,如何组织这些元素才能优化 workflow,各种元素需要规划多大位置等。和前面提到过的各种设计一样,功能组和层级也会随着设计的进行不断变化,设计者需要密切关注设计的推进情况,及时对变化的部分做出相应的调整。以下是一个示例的功能组和层级结构。

1) 主要功能组

- (1) 信息查询。提供用户所需的信息查询功能,例如,搜索、查找、浏览等。
- (2) 问题解答。回答用户提出的问题,并提供相关的解释和信息。
- (3) 任务执行。执行特定的任务,例如,预订、购买、预约等。
- (4) 娱乐和休闲。提供娱乐和休闲相关的功能,例如,游戏、笑话、音乐等。

2) 信息查询功能组的层级

- (1) 基本信息查询。提供基本的事实性信息查询,例如,定义、解释、统计数据等。
- (2) 实时信息查询。提供实时的信息查询,例如,天气、股票行情、新闻等。
- (3) 深入信息查询。提供更深入的信息查询,例如,学术研究、历史事件、文化知识等。

3) 问题解答功能组的层级

- (1) 常见问题解答。回答常见的问题,例如,常识问题、技术问题等。
- (2) 领域专家问题解答。回答特定领域的问题,例如,医学、法律、科学等。
- (3) 推理和解释。提供推理和解释,帮助用户理解复杂问题和概念。

4) 任务执行功能组的层级

- (1) 基本任务执行。执行基本的任务,例如,预订酒店、订购商品等。
- (2) 复杂任务执行。执行更复杂的任务,例如,旅行规划、项目管理等。
- (3) 个性化任务执行。根据用户的个性化需求执行任务,例如,定制化服务、个人助理等。

5) 娱乐和休闲功能组的层级

- (1) 游戏。提供各种类型的游戏,例如,文字游戏、智力游戏等。
- (2) 娱乐内容。提供娱乐内容,例如,笑话、趣闻、音乐等。
- (3) 休闲活动。提供休闲活动建议,例如,旅行推荐、电影推荐等。

以上只是一个示例的功能组和层级结构,具体的功能组和层级结构可以根据项目需求和用户需求进行调整和扩展,确保功能组和层级结构的设计能够满足用户的需求,并使用户能够方便地找到所需的功能。

4. 勾画交互框架

“勾画”交互框架,即无须给出高保真的演示模型,只需利用手头现存的工具简要地绘制一下,如图 3.8 所示。可以为当前窗口(又称为视图)划分区域,并用线条勾勒出的方形区域,标识某个图标、控制部件等。在完成以上工作后,需要为绘制的“方块图”添加适当的注释和标签介绍,并描述每个分组或元素之间的联系,在最后还需要用箭头进行标注,以表示状态或者流程的改变。

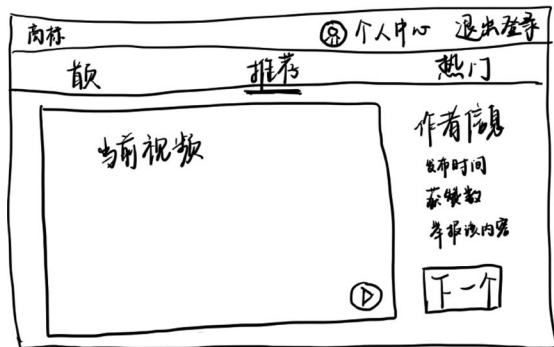


图 3.8 视频播放网站框架草图

5. 构建关键线路情景剧本

关键线路的使用频率高,且其往往是系统的核心功能。关键业务的流程和交互是否合理很大程度上影响着用户对于系统的满意度,所以关键线路对于一个交互设计来说是十分重要的。当使用视频播放软件时,更倾向于其播放视频的流程是否易用、流畅,而关注系统附带的社交功能比较少。与目标导向情景场景不同,关键路线场景以任务为导向。设计者可以将人物模型和已经设计好的交互框架代入具体场景中删除不必要的任务,优化必要任务。但关键线路情景剧本必须严格地描述每项活动的交互行为,不可以粗略地一笔带过。

具体进行构建时,设计者无须制作高保真可以交互的系统,而可以只用几个静态的图片或者页面进行演示,辅以一定的解说来评估已有功能并设计其他功能。当构建关键线路情景剧本时,一般遵循以下交互设计的要点和步骤。

1) 确定用户目标

首先,明确用户在该情景中的目标是什么。例如,用户可能想要查询信息、执行任务、解决问题等。理解用户的目标,才能设计一个符合用户期望的剧本。

2) 确定系统角色

确定系统在剧本中的角色和功能。系统可以扮演一个信息提供者、问题解答者、任务执行者等。根据用户目标和系统角色,决定系统需要提供的功能和交互方式。

3) 设计剧本流程

根据用户目标和系统角色,设计剧本流程。综合考虑用户可能的输入、行为和系统的回应,确保在具体场景下的剧本流程清晰、连贯,并能够满足用户的需求。可以使用流程图或类似的工具来可视化对话流程。

4) 定义用户输入

参考之前定义的输入方式,判断在当前情景下用户可能选择的输入方式和格式。用户可以使用文本输入、语音输入、按钮选择等方式与系统进行交互。交互系统的设计需要确保用户输入的方式符合用户习惯,并能够方便地表达他们的意图。

5) 定义系统回应

确定系统对用户输入的回应方式和内容。系统可以提供文本回复、语音回复、图形界面等,确保系统的回应能够清晰地传达信息,并满足用户的期望。

6) 考虑错误处理

考虑用户可能的错误输入或系统无法处理的情况,使用户在遇到问题时能够得到适当的引导和支持。设计者可以设计相应的错误处理机制,例如,提示用户重新输入、提供帮助信息等。

7) 用户反馈和迭代改进

在实际使用中,收集用户反馈并不断改进剧本,即通过用户反馈,可以了解用户的体验和需求,并进行相应的优化和改进。

以上是一个基本的交互设计构建关键线路情景剧本的过程。具体的设计步骤和方法可能因项目而异。在设计过程中,要始终关注用户体验和目标,并确保剧本能够满足用户的需求并提供良好的交互体验。

6. 运用验证性场景来检查设计

对于关键线路采用构建场景剧本的方法进行演示和评估,读者也许会有疑问,其他没这么重要的功能应该如何进行评估呢? 场景剧本太过费时费力,而对功能的验证与演示几乎又是必不可少的,针对一些不被频繁使用或非核心任务,利用“验证性场景”来检查设计。验证性场景无须像场景剧本一样准确严格,可以进行较为粗略的记录,必要时还可以进行一些假设,以提高检查效率。那么如何从验证性的角度出发进行工作呢? 以下进行简单介绍: 首先可以考虑一些场景的替代场景,例如,在周围环境噪声较大时,用户更倾向于利用文字发送信息交流而不是用电话交流,此时发送文字信息就是语音通话的替代操作。其次,可以对一些必须要完成但频率不是很高的场景进行验证,如绑定用户手机号、清空历史记录及其他特别的请求。最后应该考虑边缘情形场景,边缘场景虽然并不会被频繁使用,也不涉及关键功能,但正因为如此,边缘场景容易被忽略导致系统漏洞。例如,当两文件重名时,系统应该如何解决这些冲突。需要注意的是,虽然设计时致力于解决边缘场景可能出现的问题以

减少系统的漏洞,但工作的重心始终应该放在上述关键场景的完善上,设计者应该把握好在整个项目过程中为边缘场景分配的时间。

3.3.2 细化外形和行为

为了开发一个高质量的系统,首先需要开发出坚实的框架,在框架设计完成后再进行细节的填充。这样会使设计过程清晰、有逻辑,把大的不易完成的目标转换为小的清晰可行的目标。只要将每个阶段的过程分别完成,就能得到完整的系统产品。这个阶段,设计者已经可以看到设计的雏形,要做的是进行进一步的转换和提炼。此时要对用户界面有比较清晰的描绘,最好采用全分辨率的屏幕,从像素级进行描述,如图 3.9 所示。



图 3.9 华为商城像素级演示界面

在创建交互设计框架时,需要利用前面介绍过的 6 个步骤来指导设计者的设计,即定义形式要素、姿态和输入方法,定义功能性和数据元素,确定功能组和层级,勾画交互框架,构建关键线路情景剧本,运用验证性场景来检查设计。在对外形和行为进行完善和细化时,也可以使用与创建框架相似的步骤。需要注意的是,如果外设不存在意料之外的问题,如成本过高,在细化阶段将不再对外形因素和输入方法做出调整。在这个过程中,比起对于整体风格和布局的考虑,设计者更需要将注意力放在细节上,进一步对视觉、工业等设计进行填充。现在应该主要关注页面和窗格所在层次,利用场景进行模拟,在模拟过程中可以发现较为深层的容易被忽略的问题,而这些问题的发现往往能推动设计者对外形和行为的细化,提高交互产品的可用性,降低其出错率。

在细化过程中往往会出现一些问题,例如,众多的场景再加上详尽的对于外形和行为的检查与设计会使设计者的工作量陡增。如果这些工作全权由设计者完成,不但会使设计人员身心俱疲,还可能影响到最后交付的交互产品的质量,所以比起事事亲力亲为,设计者更应该做的是根据已有的设计,提炼出一套符合当前产品要求的风格指南。对于一些边缘化的、次要的部分,程序员可以直接参考这份指南来进行视觉设计元素的运用。

需要注意的是,不仅在框架开发阶段需要遵守各项设计原则,在进行细节的填充时也要注意以用户为中心、遵循交互设计的4项基本原则,把握关键特征。此外,在细节填充阶段需要与程序开发团队密切合作。因为需要把前期提出的概念和想法变为现实,只有与开发团队密切沟通,才能验证设计是否可行。例如,预设的想法能否呈现在程序中,怎样设计才能使已有的技术条件对交互设计的效果有加强作用。在确实出现问题时,可以与设计团队协调进行适当的修改,必要时可以邀请产品涉众参与其中,以确保修改不偏离设计目标,满足在用户体验方面的要求。

产品的交互设计理念和细节上的考量不能每次都依靠交互设计者口述。为了方便设计开发团队更好地进行交流与总结,需要提炼出一份规范的文档。在文档中应当给出底层的模式风格、原则和基本原理,包含整个产品的外形和行为规格。为了更好地起到解释作用,对于复杂的交互系统,在文档中还应添加描述行为的详细故事板和可交互的原型。此外,设计团队应注意与开发团队保持密切的沟通,按开发团队需要,对确实存在疑问的部分填充解释性的内容。

3.3.3 验证与测试设计

在这一阶段,设计框架已经完成,并且填充了一定的细节,系统设计逐渐趋于完善。这一阶段的目标是对已有内容进行评估与改进而非进行新的创造。对于系统可用性的测试已经不限于用人物模型和场景剧本进行演示,而是可以用更有效的方式进行。例如,向用户展示某个流程或界面,并要求用户就此提出意见和建议,这也是在设计后期仍要征求用户意见的原因——及时修改不恰当的设计,防止在设计后期没有时间发现的错误让设计开发团队陷入困境。

在此阶段仍要进行用户反馈和可用性测试,但与项目前期的反馈和测试不同的是,此时进行测试能发现细节问题并进行微调。例如,不合理的操作顺序,不符合用户心智模型的交互设计,过长的响应时间,不易用的输入输出方式等。无论如何,此时的验证仍存在一定的局限性,通过一定的试用和演示,虽然能发现新手用户在操作中的问题,但不能发现更深层次的问题。如果验证与测试仅停留在当前阶段,可能会导致中间用户和专家用户在使用时发现很多问题。为了发现这些问题,还需要继续进行进一步的检测以提高系统精度。

在和用户一起进行验证时大概有几种方法可以选择,设计团队可以根据当前的需要选择合适的方法。选择非正式的环境,则测试进行得更轻松;选择正式的环境,则测试进行得更精准。较为轻松的方式,如在不经解释的情况下,让用户利用交互产品完成特定的功能,并且引导用户在使用系统时说出自己的疑问和想法,设计团队对用户的意见进行记录。较为正式的方式,如以热点图调查被试者的视觉数据,以样本为底图,上面叠加收集到的视觉数据,颜色越深,则表示该部位注视点越多、注视时长越长。可用性测试,可以用来测试设计在多大程度上能辅助用户完成特定的功能。测试的范围越大,数据越具有代表性,更能反映功能可用性的一般程度。但也不是参与测试的人越多越好,在人数达到一定量后,测试的精度不会随着人数的增加而增加,所以需要设计者根据项目的规模选择合适的测试人数。

为了设计出用户满意度高的交互产品,需要进行充分的用户研究。用户研究包括进行访谈、发放问卷、任务分析等。虽然一般认为用户研究应该在框架搭建之前完成,但可用性测试也是用户研究的一种,与其他研究方式配合能达到更好的分析效果。在对于设计和测

试所占比重进行安排时,设计者可以参考交互设计的一般规律:在项目花费的时间和投资一定的情况下,在产品初期,如框架搭建和设计中花费时间收获的效益要好于将这些时间投入测试中。也就是说测试是必不可少的,但具体安排多少时间、在项目中的占比如何,需要整个设计开发团队进行沟通和协调。

在选择研究方式时,设计者可以选择定性的研究方式或定量的研究方式。定性的研究方式是指根据社会现象或事物所具有的属性在运动中的矛盾和内在规律性来研究事物的一种方法或角度。例如,交互设计领域的观察法、访谈法等。定量的研究方式一般是为了对特定研究对象的总体得出统计结果而进行的。常见的如调查研究法和实验研究法,容易被忽略的如可用性测试,都是很好的定量测试方法。定性研究和定量研究各有优缺点。定性研究给出了问题大体的走势,方便快捷;定量研究给出了具体的统计量,准确,值得信赖。设计者可以根据需要选择合适的研究方式,接下来主要对可用性测试中的一些注意事项进行说明。

可用性测试能定量地给出数据,因此它十分适合进行不同方案之间的比较,从中选择一种或几种的情形。当需要对已有的交互设计进行验证和修改时,可用性测试给出的数据往往是用户使用感的真实反映,是很有价值且值得参考的。它作为一个广受认可的研究方法有诸多好处,但也存在其局限性:对于首次使用系统的用户的体验反映比较准确,而很难描述频繁使用系统的用户的体验。这也意味着,用这种方法进行测试更多地注重的是新手用户的感受,而在某种程度上忽略了群体最庞大的中间用户的需要,更是无法反映专家用户的感受。在测试中需要保证,所有标准都是可以表示和测量的,它应当是具体的、无歧义的。在无从下手时,可以从以下几点入手进行测试:表意——按钮、组件、标签等的内容或命名是否合理,是否能清晰地传达当前的功能,用户在使用时有没有歧义;布局——各个功能模块的布局位置是否合理,页面的分区是否能帮助用户找到并完成相应功能;易学——对于新用户是否友好,如果之前没有使用过同类交互产品的经验,需多长时间能掌握当前产品的使用;有效——功能模块能否完成对应的任务;精度——交互产品是否有错误发生,是否能检测用户可能发生的错误(如输入/输出错误)并给出一定的提示。

为了更好地掌握测试的时机,设计者应当对形成性评价和最终性评价有一定的概念。在开发过程中对交互设计的某个部分进行的测试,称为形成性评价;在产品已经完成后对产品整体进行的测试会产生最终性评价。

形成性评价是一种定性的测试,它一般在项目有一定基础要进一步深挖时进行,具有便捷且快速等特点。在设计时要注意将问题描述准确,通常就用户在完成任务中的感受进行提问并引导用户用语言将这些感受描述出来。通过形成性评价可以了解系统功能的实现程度。通俗地讲,如果用户把交互产品当作一种工具,形成性评价可以反映这个工具能否顺利地帮助用户解决特定的问题。

相比形成性评价,最终性评价更具有宏观性,它的评价结果甚至能为交互产品的生命周期提供规划和参考。但这种方法也有其局限性,它的流程比其他评价方式更烦琐且即使发现了问题,因为交互系统的设计和开发已经到了中后期阶段,进行修改的时间成本和经济成本都会比较大,设计者应当对这种方法的缺点有所了解。

测试过程中可能出现很多问题,如无法核实量化标准是否符合事实,也可能陷入在项目后期仍要进行较为大量修改的麻烦。但无论如何在进行测试时有些问题需要格外注意,接

下来给出一个案例,模拟在开发过程中进行测试的流程。首先,要选择合适的测试时机,过早会因为项目不够完整导致很多测试流程无法完成,影响测试效果;过晚会使后期的修改变得异常艰难,即使确实发现一些问题也没有时间进行调整。确定了合适的时机后设计者可以开始寻找用户,在寻找过程中可以以人物模型为参考,从产品目标群体中招募测试者。测试还需要协调者就过程中出现的问题进行调节和说明,为避免已有项目经历对工作立场的影响,最好选择当前项目无关人员进行测试。在测试前有关人员应该对产品需要完成的测试任务和用户的体验内容进行规划,在测试时引导用户描述在使用当前设备时的感受,并做好记录。在以上工作都完成后,可以组织观察者和设计者一起听取受试者操作的记录和使用时的感受,分析问题出现的位置和原因,从而找出对应的解决方案。

人在不同位置上看问题的角度不同,人和人之间也存在巨大的差异,所以不是所有用户与设计者都能进入互相信任的良性沟通状态。在与用户的沟通出现问题时,可以首先尝试用人物模型对用户的目标、期望和态度进行模拟,来辅助之后双方进行直接的交流。设计者作为测试活动的最大受益者,也应积极努力推动测试活动功能的进展。依据以往开发中的经验,设计者需要做到如下几点:将测试重点放到主要的功能和业务流程中,而非在细节问题上强调太多,花费较多时间;在过程中需要将设计阶段已有的人物模型代入主要场景中,指导用户进行测试;在用户受试过程中做好记录并与团队协商评价过程中出现的问题,方便后期回顾和解决这些问题,达到测试的最终目标。

思考与实践

1. 试解释“交互设计生命周期”的含义。
2. 试解释“交互设计的关键特征”的含义。
3. 交互设计的基本活动有哪些?请谈谈如何更好地获取用户需求。
4. 试将星状生命周期模型代入自身以往的交互设计中,并分析利用该模型的优点。

5. 交互设计生命周期模型与软件工程生命周期模型一个较为显著的区别是,在生命周期中用户的参与度。假设你正在开发一款跨平台的通信工具,这个产品支持单人、多人参与,能通过手机网络发送语音、图片、视频和文字。请谈谈怎样能让用户更好地参与到产品开发的整个生命周期中。