

# 产品设计必读书籍推荐

By 设计湿了没

[www.idesign123.cn](http://www.idesign123.cn)

经常有朋友要我推荐下做产品设计尤其是移动产品设计的书籍，结合自己的一些经验及其他一些 PM 的推荐，整理了如下书单给大家参考，并提供了下载地址，希望对大家有帮助。

1. 各个平台的设计指南
2. 《简约至上一交互设计四策略》
3. 《Tapworthy（触动人心）》
4. 《启示录：打造用户喜爱的产品》
5. 《用户体验的要素》
6. 《社会心理学》
7. 《情感化设计》
8. 《至关重要的设计》
9. 《赢在用户》
10. 《About Face 3 交互设计精髓》

以后有修订的版本，或者其他设计资料，会在微博@设计湿了没网站 和微信公众号（ID：[idesign123](https://www.idesign123.cn)）里通知，大家可以扫描二维码在微信里和我交流，或访问网站（[www.idesign123.cn](http://www.idesign123.cn)）了解更多互联网产品视觉、交互、用研及设计思想趋势和观点。



扫一扫或搜索：[idesign123](https://www.idesign123.cn)



New  
Riders

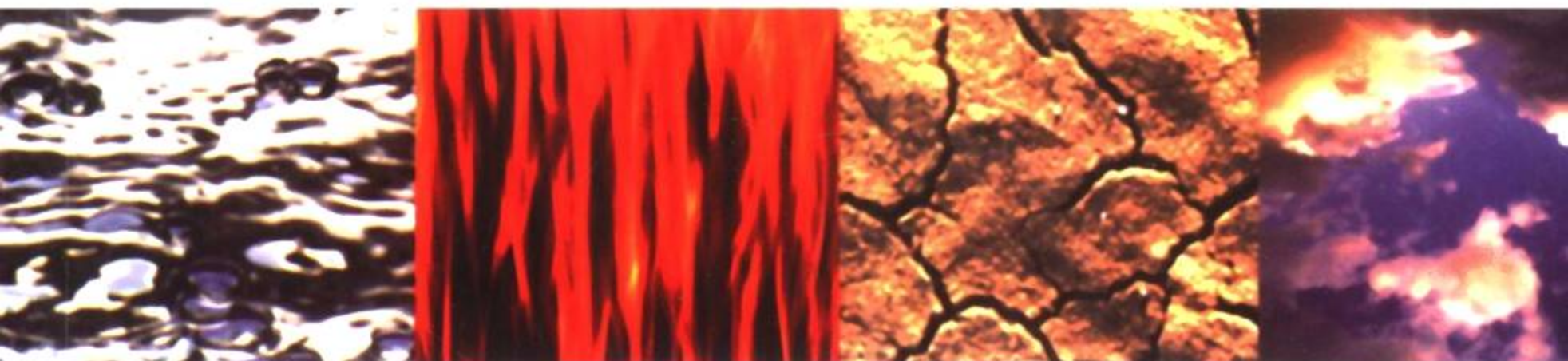
Ajax之父的预言之作 体现Web 2.0思想精髓

# 用户体验的要素

以用户为中心的Web设计

## The Elements of User Experience

### User-Centered Design for the Web



(美) Jesse James Garrett 著  
范晓燕 译



机械工业出版社  
China Machine Press



# 用户体验的要素

## 以用户为中心的Web设计

### The Element of User Experience User-Centered Design for the Web

聪明的企业意识到网页设计远远比创建条理清晰的代码和鲜明的图形重要得多。一个网站在满足您的战略目标的同时，还应该满足您的用户需求。如果没有一个“有凝聚力的、统一的用户体验”来支持的话，即使最好的内容和最精密的技术也不能帮助您平衡这些目标。

但是创建用户体验却是出乎意料的复杂。有很多方面都要考虑——可用性、品牌识别、信息架构以及交互设计等——这使得“建立一个成功的网站的唯一方法”似乎就是要花大价钱请那些完全了解这些细节的专家来做。

本书用清晰的阐述和生动的图形分析了“以用户为中心的设计方法（UCD）”来进行网站设计的复杂内涵，并关注于思路而不是工具或技术。Jesse James Garrett给了读者一个关于“网站用户体验开发的总体概念”，即从战略和信息架构需求到视觉设计。这种简单易学的介绍方法有助于任何一个网站开发团队（大的或小的）去创建一个成功的用户体验。

**Jesse James Garrett** 是用户体验咨询公司 Adaptive Path 的创始人之一。从用户体验要素在2000年3月初次发布到网上以来，Jesse 的所绘制的这个模型已经被下载了2万多次。Jesse 的互联网从业经验包括 AT&T、Intel、Boeing、Motorola、Hewlett-Packard 以及美国国家公众广播等。他在用户体验领域的贡献包括“视觉词典（the Visual Vocabulary）”，一个为规范信息架构文档而建立的开放符号系统，现在这个系统在全球各个企业中得到广泛的应用。他的个人网站 [www.jjg.net](http://www.jjg.net) 是提供信息架构资源的网站中最受欢迎的一个。

“这是一个能给使用您的网站的人们带来高质量体验的流程，而 Jesse James Garrett 使这个复杂的设计流程变得清晰明了。他解析并模型化了原因和概念的问题，并揭露出那些常常是带有深奥的技术外衣的问题的本质。”

——Alan Cooper (《About Face》和《The Inmates Are Running the Asylum》的作者)

“Jesse James Garrett 最终把他著名的图表扩展成了一本书，这本书使整个混乱的用户体验设计领域变得明晰。同时，由于他是一个非常聪明的家伙，因此这本书非常简短，结果就是几乎每一页都有非常有用的见解。”

——Steve Krug (《点石成金：访客至上的网页设计秘笈（原书第2版）》作者)

“终于，一个结合了许多不同部分的、用户体验的说明出炉了。本书条理清晰、可读性强，而且是必读的。”

——Louis Rosenfeld (《Information Architecture for the World Wide Web》的合著者)

New Riders

投稿热线: (010) 88379604  
购书热线: (010) 68995259, 68995264  
读者信箱: [hzjsj@hzbook.com](mailto:hzjsj@hzbook.com)

华章网站 <http://www.hzbook.com>

网上购书: [www.china-pub.com](http://www.china-pub.com)

封面设计: 王建敏



上架指导: 计算机/Web设计

ISBN 978-7-111-22310-8



9 787111 223108

ISBN 978-7-111-22310-8

定价: 25.00 元



好设计, 更有爱, 关注微信 [idesign123](https://www.idesign123.com)

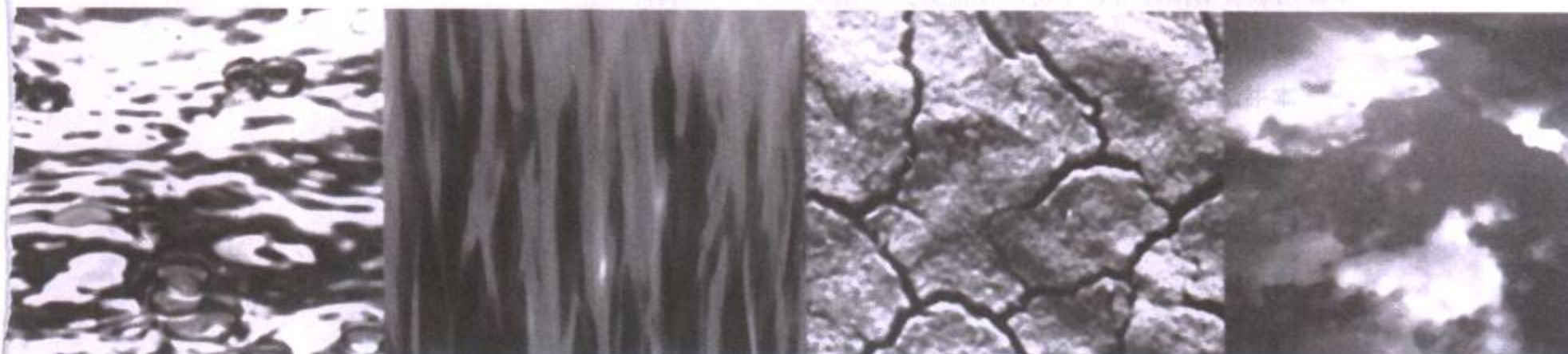


# 用户体验的要素

以用户为中心的Web设计

**The Elements of User Experience**

User-Centered Design for the Web



(美) Jesse James Garrett 著  
范晓燕 译



机械工业出版社  
China Machine Press



好设计,更有爱,关注微信



idesign123



本书是AJAX之父的经典之作。本书用简洁的语言系统化地诠释了设计、技术和商业融合是最重要的发展趋势。全书共8章，包括关于用户体验以及为什么它如此重要、认识这些要素、战略层、范围层、结构层、框架层、表现层以及要素的应用。

全书用清晰的说明和生动的图形分析了以用户为中心的设计方法（UCD）来进行网站设计的复杂内涵，并关注于思路而不是工具或技术，从而使你的网站具备高质量体验的流程。

Authorized translation from the English language edition entitled The Elements of user Experience: User-Centered Design for the Web by Jesses James Garrett, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2007 by Jesses James Garrett.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2007 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2007-2876

### 图书在版编目（CIP）数据

用户体验的要素：以用户为中心的Web：英文/（美）加瑞特（Garrett, J. J.）著. —北京：机械工业出版社，2007.9

书名原文：The Elements of User Experience: User-Centered Design for the Web

ISBN 978-7-111-22310-8

I. 用… II. 加… III. 主页制作—程序设计—英文 IV. TP393.092

中国版本图书馆CIP数据核字（2007）第139841号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李南丰

中国电影出版社印刷厂印刷·新华书店北京发行所发行

2008年1月第1版第1次印刷

186mm×240mm · 11.5印张

定价：25.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换  
本社购书热线（010）68326294



- 上市一周荣登China-pub网销售排行榜榜首。
- 上市一月后即重印。
- 入选美国著名图书频道Book Pool评选的近10年来计算机专业图书50强。
- 被Web设计人员奉为圭臬的经典之作。
- 全球销量超过10万，Amazon网站的网页设计类图书的销量排名佼佼者。



点石成金：访客至上的网页设计秘笈（原书第2版）

作者：Steve Krug

译者：De Dream

书号：7-111-18482-3

定价：39.00元

- 显示设计理论奠基之作。
- 国外畅销UI设计图书。



一目了然 Web软件显性设计之路

作者：Robert Hoekman

译者：何潇

书号：978-7-111-22362-7

定价：39.00元

- 继《点石成金》后，人物角色方面的力作。
- UCDchina成员范晓燕（Angela）翻译。



赢在用户：Web人物角色创建和应用实践指南

作者：Steve Mulder Ziv Yaar

译者：范晓燕

技术审校：De Dream

书号：978-7-111-21888-3

定价：29.00元



专业成就人生

立体服务大众



好设计，更有爱，关注微信

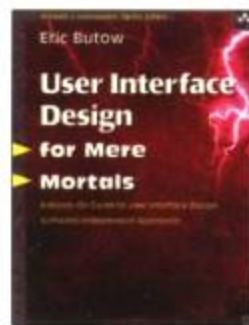


idesign123



## 尽在华章出版

■ UI设计的入门佳作。



User Interface Design for  
Mere Mortals

作者: Eric Butow

译者: 孙志超

预计出版时间: 2007年12月

■ 不可多得的GUI设计优秀资源。



GUI设计禁忌

作者: Jeff Johnson

译者: 王 蔓 刘耀明 等

书号: 7-111-15666-8

定价: 49.00元

■ 帮您摆脱困境的完美方案。



Web设计禁忌

作者: Jeff Johnson


译者: 张 颖

书号: 7-111-18008-9

定价: 38.00元

敬请读者关注 <http://www.hzbook.com>  
更多华章好书



好设计,更有爱,关注微信  idesign123

---

---

## 译者序

---

---

“the elements of user experience”的图示已经在网上流传了很长时间，许许多多的设计师都在争相转载，并表示这张图给了自己很大的启示。但是，由于国内和国外的设计师在环境、文化、经验方面的种种差异，再加上一张简单的图示也很难把这个复杂的问题说清楚，所以真正能看懂这幅图并将其应用到实际工作中的人几乎是零。

所幸的是，Jesse James Garrett本人也意识到了这个问题，正如他在“背后的故事”中所描述的，他最终决定把这个著名的概念扩展成一本书，并在其中加入了更详细的描述和更多的例子，这样就能很好地帮助有志于从事用户体验设计的国内从业者去理解和应用这些概念。

尽管这本书已经讨论了很多实例，也提供了一些方法，但是我仍然认为这是一本阐述设计思想的书。我们常说：“授人以鱼不如授人以渔”。如果说“鱼”是实例，“渔”是方法的话，那么这本书最大的作用，应该是阐明了“什么是渔”——这个至今仍然被争论不休的议题，也是很多交互设计师、信息架构师、产品设计人员等相关从业人员没有想清楚的问题。

作为一个从业者，我很高兴有机会翻译这本书，在反复揣摩作者文字和意图的过程中，我对这个行业的理解和认识也在不断地提炼和清晰。Jesse James Garrett认为自己将它们汇集到这本书中得到了很多的启迪和收益，于我也是如此。

借用作者的一句话：“一切就像是有幸运之神眷顾一样”，我希望你也能如我一般幸运。





《用户体验的要素》用简洁的语言系统化地诠释了设计、技术和商业融合是最重要的发展趋势。它的潜力不仅在于是能衡量“哪些是可能的”，更重要的在于我们“如何把技术的可能性转变成人类体验的提升”。

这不是提升那种“难以把握的用户体验”的概念。这是一个挑战，在你选择问题解决方式的同时，还要尊重用户的体验。这门学科实在是太年轻了，以致于大多数的设计师都缺乏方向的指导，而不得不靠自己摸索着前进。Jesse James Garrett适时地出现了。他引入了一个强有力的观点，并且清楚地描述了它是如何通过一系列分析方法，使得问题更容易控制和解决、并与交互设计产生联系的。



——Richard Grefe（执行总监，AIGA）

AIGA（American Institute of Graphic Arts：美国设计师协会）联合出版公司所选择的，都是在设计领域中最先进的思想家的工作成果——他们的作品定义了新的设计界限，AIGA同时也支持那些追求卓越和开拓相关领域的设计从业者。AIGA是美国最大和历史最悠久的设计师专业协会，它的成员都是来自各个设计领域的专家，其研究领域涵盖了设计过程中的各个环节。



## 关于作者

Jesse James Garrett是Adaptive Path——一个位于San Francisco的一个用户体验咨询公司——的创始人之一。从1995年开始，Jesse帮助一些企业改进他们的网站，其中包括AT&T、Intel、Boeing、Motorola、Hewlett-Packard和美国国家公众广播等。他在用户体验领域的贡献包括“视觉辞典”（the Visual Vocabulary）——一个为规范信息架构文档而建立的开放符号系统——现在这个系统在全球各个企业中得到了广泛的应用。他的个人网站www.jjg.net是提供信息架构资源的网站中最受欢迎的一个，他同时也是信息架构和用户体验的积极倡导者。2005年2月，Jesse James Garrett发表了《Ajax: A New Approach to Web Applications》一文，标志着Ajax的诞生。由此，他也被公认为“Ajax之父”。



### 关于技术评审人

在《用户体验的要素》的整个写作过程中，这些评审人提供了相当实用的专家意见。在撰写本书期间，他们评价了所有与技术有关的内容、组织和流程。在确保本书提供高质量的技术信息，并切合读者的需求方面，他们的意见是举足轻重的。

David Hoffer在信息架构和交互设计领域有着丰富而广泛的经验。他现在是CTB/McGraw Hill的市场推广部门的高级用户界面设计师，负责架构、界面和可用性。在加盟McGraw Hill之前，David作为高级信息架构设计师为Hill and Knowlton公共关系公司工作——这是美国最大的一家公共关系企业，同时也是全球第







二大的广告巨头WPP Group的一分子。David在Amazon.com的Alexa互联网子公司担任了两年的高级设计师，在那里他为Amazon的Alexa浏览器产品开发了客户端界面。David还曾为很多高科技企业提供过咨询服务，包括像NERDS和ActiveBuddy这样的小型创业型企业，也包括一些行业中更大、更知名的公司，例如Motorola和DEC。David拥有罗切斯特理工学院工业设计的学士学位，他热爱这门专业以及他所从事的工作。



Molly Wright Steenson从1994年开始从事与互联网行业相关的工作。她曾领导了60多个网站的架构设计和开发工作，服务过的公司包括Netscape、Reuters、Wrigley、Nike、Genentech等等。她经常撰写与互联网和设计有关的文章，并在一些国际会议上发表演讲。Molly支持“以用户为中心”来进行设计和开发，同时也是AIGA体验设计类资讯的网站编辑。她作为项目经理服务于Razorfish，一家位于San Francisco的公司，旗下拥有一个名叫Girlwonder.com的网站。她还可以讲流利的德语、法语和荷兰语。



范晓燕，UCDChina发起人，从1997年开始从事互联网相关工作，拥有超过10年的互联网从业经验；现从事用户体验研究、分析，以及互联网产品的设计和管理工作。推崇“以用户为中心（UCD）”的设计思想，是用户体验设计的积极推广者和实践者。个人博客：<http://ucdchina.com/angela>，电子邮箱：[angela.fan@msn.com](mailto:angela.fan@msn.com)。



---

---

## 致 谢

---

---

不要被封面上的名字数目欺骗了——这本书聚集了很多人的智慧和努力。

首先，我必须要感谢我在Adaptive Path的搭档：Lane Becker、Janice Fraser、Mike Kuniavsky、Peter Merholz、Jeffrey Veen和Indi Young，我能完成这个工作完全是由于他们对我的信任和支持。

然后是在New Riders的每一个人，尤其是Michael Nolan、Karen Whitehouse、Victoria Elzey、Deborah Hitte-Shoaf、John Rahm和Jake McFarlan。他们在撰写过程中起到了关键的指导作用。

Kim Scott和Aren Howell用他们敏锐的眼光来关注这本书所有设计的细节。他们对作者本人的建议所付出的耐心尤其值得称赞。

Molly Wright Steenson和David Hoffer在对我的手稿的评审中提出了很多有价值的见解。每一个作者都应该如此幸运。

Jess McMullin在许多方面都是我最尖锐的批评家，这本书在他的影响下得到了大大的改善。

同样感谢那些在撰写书籍方面富有经验的作者，他们的金玉良言帮助我在完成这个项目的同时，还能保持头脑清醒：Jeffrey Veen（再一次）、Mike Kuniavsky（再一次）、Steve Krug、June Cohen、Nathan Shedroff、Louis Rosenfeld、Peter Morville





和（特别是）Steve Champeon。

其他给我提出过有价值的建议或精神支持的朋友包括：Lisa Chan、George Olsen、Chritina Wodtke、Jessamyn West、Samantha Bailey、Eric Scheid、Michael Angeles、Javier Velasco、Antonio Volpon、Vuk Cosic、Thierry Goulet和Dennis Woudt。他们帮助处理了那些被我忽略的事情，正是因为这样，他们成为了我最好的同事。

本书写作过程中的音乐伴奏由Man（也许应该是Astro-man）、Pell Mell、Mermen、Dirty Three、Trans Am、Tortoise、Turing Machine、Don Caballero、Mogwai、Ui、Shadowy Men on a Shadowy Planet、Do Make Say Think和（尤其是）Godspeed You Black Emperor提供。

最后，还有三个人。如果没有他们，这本书就不可能完成：Dinah Sanders，在一个温暖的Texas晚上的聚会中，是她坚持要我认识某个人；我的妻子，Rebecca Blood，她使我在各方面都变得更强大、更聪明；以及Daniel Grassam，没有他的友谊、鼓励和支持，我可能还没有找到在这个领域中的方向。谢谢你们。



---

---

# 前言

---

---

这不是一本关于“怎样做 (How-to)”的书。有很多很多讨论如何建设网站的书，这本不是。

这不是一本关于技术的书。在这里你找不到一行代码。

这不是一本有答案的书。相反，这本书说的是“如何提出正确的问题”。

这本书将告诉你，在你阅读其他书籍之前，你需要提前了解什么。如果你需要一个大的概念，如果你需要了解用户体验设计师所做出的决策的环境，这本书很适合你。

这本书经过精心设计，使你可以在一两个小时之内读完。如果你是一个刚刚进入用户体验领域的新手——可能你是一个负责组建用户体验团队的管理人员，或者你是一个碰巧进入这个领域的作家或设计师——那么这本书将给你一些基础的概念。如果你已经对这些方法和用户体验领域的关注点很熟悉了，那么这本书将帮助你更有效地把这些概念传达给你合作的人们。

## 背后的故事

由于被询问得太多，所以我决定把本书的诞生过程写下来。

在1999年下半年，我作为第一个信息架构设计师加盟了一个从事多年网页设计顾问的公司。我通过很多种方式来明确我的职位职责并向人们不停地讲述我所做的事情是什么、这些事





情如何与其他人所做的工作融合到一起等等。一开始，他们都十分小心而且还有一点警惕，但是很快他们开始意识到我的存在是为了让他们的工作更容易，而不是更困难。我的出现并不表示他们的权威被降低了。

与此同时，我正在编写一个与我工作相关的、我个人用于收藏网上资源的网站（它最终作为我在[www.jjg.net/ia/](http://www.jjg.net/ia/)中信息架构资源的页面被发布在互联网上）。在做这些研究的时候，我总是不断地被这个领域中看上去很相似而实际上被随意和胡乱使用的一些基础概念的词汇所困扰。某个资料中称为“信息设计”的东西很显然和另一个资料所称的“信息架构”完全一样，而第三份资料中把所有的这些放在一起称为“界面设计”。

在1999年底到2000年的一月期间，我强撑着完成了一系列对这些关键议题的一致定义，并找到一种方式来表达它们之间的关系。但是我当时非常地忙，被一些正在进行中工作缠住脱不开身，我试着去阐述和说明的那种模型没有真正地在工作中产生效果，所以在这个一月快结束的时候，我已经放弃了整个的念头。

同年3月，我到Texas的Austin参加一年一度的South by Southwest 交互节会议。这是一个忙碌又发人深省的星期，在此期间我几乎没睡多少觉——大会的日程安排和晚上的活动就像一场耗时两三天的马拉松比赛。

那个星期快结束的时候，我穿过Austin机场的安检口准备登上返回San Francisco的飞机，这时一个三维的矩阵突然一下就跳进了我的脑海里，并完全占据了我整个思绪。在登上飞机之前我都一直耐心地等待着。而在我坐下的同时，我就掏出记事



本把它画了出来。

回到San Francisco之后，由于伤风我几乎立刻就倒下了。我经历了大约一个星期的高烧和谵语。在刚刚感到特别清醒的时候，我马上就把记事本上的这幅草图变成能整洁地展现在一张纸上的完整图示。我把它称为“用户体验的要素”。后来我听说，这个称呼是如何唤起了大多数人对于“元素周期表”和“Strunk and White<sup>①</sup>”的回忆的。不过让大家失望的是，在选择这个标题的时候，我的脑海中完全没有这种联想——之所以从辞典中把“要素 (element)”挑选出来，只是为了代替蹩脚的、听上去很技术化的“组件 (component)”这个词。

在3月30日，我把最终的图发布到了网上（你现在仍然可以在[www.jjg.net/ia/elements.pdf](http://www.jjg.net/ia/elements.pdf)中找到最初的这个图示）。它开始得到一些关注，首先是Peter Merholz和Jeffrey Veen，他们后来成为我在Adaptive Path的搭档。接着，我在信息架构峰会（Information Architecture Summit）和更多的人有了一定的交流。最后，我开始听到来自全世界各地的人们讲述他们如何使用这个图示去教授他们的同事，以及在讨论与用户体验相关的议题时，把这个作为通用的词汇表在企业内使用的。

在这张图示初次发表之后的一年间，“用户体验的要素”在我的网站上的下载数量超过了2万次。我听说它在一些大型企业或小型的网站开发团队中，被用于帮助大家更高效地合作和沟通。到这个时候，我开始认为在书中阐述这些想法，这会比用

---

① Strunk and White: William Strunk Jr.和E.B.White是两位文体专家，提倡简洁干净的文风。他们认为，好语言的标准是应该使读者念起来不觉得有累赘和障碍。Strunk和Whitew合著了《The Elements of Style》一书。





一张纸的表格能更好地解决这类需求。

又一个3月来到了，我又一次来到Austin的South by Southwest，在这里我认识了New Rider出版社的Michael Nolan，并向他讲述了我的想法。他对此非常感兴趣，同时很幸运的是，他的老板也同样感兴趣。

于是，一切就像是有幸运之神眷顾一样，这本书最终到达了你的手中。我希望这里所提到的这些想法能对你有所启迪和收益，就如同我将它们汇集到这本书里所得到的启迪和收益一样。

Jesse James Garrett

2002年7月

[www.jjg.net/elements/](http://www.jjg.net/elements/)



---

---

# 目 录

---

---

译者序

关于作者

致谢

前言

## 第1章

### 用户体验为什么如此重要

日常生活中的遭遇	1
什么是用户体验	2
用户体验和网站	5
竞争优势和投资回报率	7
记住你的用户	13

## 第2章

### 认识这些要素

五个层面	15
自下而上地建设	17
基本的双重性	21
用户体验的要素	25
应用这些要素	28





### 第3章

#### 战略层：网站目标和用户需求

战略层定义	34
网站目标	35
用户需求	40
团队角色和流程	50
推荐阅读	53

### 第4章

#### 范围层：功能规格和内容需求

范围层定义	56
功能和内容	59
收集需求	63
功能规格	66
内容需求	69
确定需求优先级	72
推荐阅读	76

### 第5章

#### 结构层：交互设计与信息架构

结构层定义	80
交互设计	81
信息架构	88
团队角色和流程	100
推荐阅读	105



## 第6章



## 框架层：界面设计、导航设计和信息设计

框架层定义	108
习惯和比喻	110
界面设计	114
导航设计	119
信息设计	125
线框图	129
推荐阅读	133

## 第7章



## 表现层：视觉设计

表现层定义	136
忠于眼睛	138
对比和一致性	140
内部和外部的—致性	143
配色方案和排版	146
设计合成品和风格指南	150
推荐阅读	153

## 第8章

## 要素的应用

实例：搜索引擎的实施	159
提出正确的问题	161
马拉松和短跑	163





长久以来，技术已经成为我们日常生活环境的一部分。它时而令我们备受鼓舞，时而令我们感到沮丧；它使生活变得简单，又使生活变得复杂；它使我们变得疏远，又使我们更加亲近。即使这样，我们还是不得不每天都和技术打交道，我们几乎忘记了，技术产品也是由人们制造的。当产品满足了人们的需要时，它的制造者应该受到赞扬；反之，则会受到指责。

## 日常生活中的遭遇

每个人都曾经历过这样的日子。

你知道我说的是什么样的日子：当你醒来时阳光已经照进了你的窗户，而奇怪的是，你设置好的闹钟竟然没有响。你看了一眼那个闹钟，发现它上面的时间是凌晨3点43分。你跌跌撞撞地爬下床找出另一只钟，它告诉你，如果你想按时上班，就必须在10分钟之内出门。



你一边打开咖啡机，一边匆忙地穿衣服，但当你去取那杯对你至关重要的咖啡时，却发现壶里没有咖啡。没有时间考虑为什么——因为你必须去上班了！

你大概走了一个街区那么远的时候，发现车快没油了。在加油站，你本来准备使用那台可以插ATM卡的加油机，但不知为什么，机器说什么也不能识别你的卡。于是你不得不跑进去付现金，这样你就得排队，等待那个收银员慢吞吞地逐一接待排在你前面的人们。终于，你加好了油，驶离了加油站，但与此同时，听到你的油箱盖从车顶滚落，蹦蹦跳跳地滚到了街道中央。

由于发生了交通事故，你不得不绕了条远路，所以你花在路上的时间超过了预想的时间。结果就是：尽管你已经很努力，你还是迟到了。终于，你来到你的办公桌旁。你感到焦躁不安，满腹委屈，筋疲力尽，怒气冲天——实际上你的一天还没有真正开始呢，你甚至连一口咖啡都还没有喝到。

## 什么是用户体验

这看起来真是一连串的倒霉事——这一天只是这类事件的一个缩影。但是让我们来分析一下，看是不是所有这些倒霉事本来是可以避免的：

**交通事故：**事故发生的原因是开车的那个司机为了调低收音机的音量，而暂时把视线从路面上离开。他不得不这么做，因为他用手摸不出哪一个是音量控制按钮。





**油箱盖：**你丢掉油箱盖是因为在加油时把它放到了车顶上，加完油以后，又因为太着急而忘记了把它盖回去。如果加油时根本不需要把油箱盖取下来放在别处——比如油箱盖是用某种方法拴在车上的，你就不可能丢掉它。

**收银机：**加油站收银机前排的长队之所以移动缓慢，是因为收银机操作起来过于复杂，很容易把人搞糊涂。收银员在登记款项的时候，必须集中全部注意力来操作，否则就会很容易出错，然后不得不重新来过。如果收银机设计得更简单一些，按钮的布局 and 颜色有所不同，那也许就根本不会有那么长的一个队伍了。

**加油机：**如果不是加油机拒收你的ATM卡，那你也就不用去排队付钱。其实你只要把卡掉个个儿插进去，加油机就可以读出了。但是，在加油机上却没有任何提示告诉你如何插卡，而你当时太着急了，没有想到把卡换几个方向插进去试试。

**咖啡机：**咖啡机之所以没有煮出咖啡，是因为你匆忙之中没有把“开”的按钮按到底。咖啡机在被打开时，没有任何现象表明它确实被打开了：没有指示灯，没有声音，哪怕是“滴答”的一声也没有。你以为你打开了咖啡机，其实没有。如果你会设置的话，也可以让咖啡机每天早上自动开启，这个问题还是可以避免的。但你从来没有学会使用这项功能——当然啦，前提是你幸运地知道有这么一个功能。



**闹钟：**现在，我们来看看这一大串事情的始作俑者——那只闹钟。闹钟没有响是因为钟上的时间错了，钟上的时间错了是因为你的猫半夜时分踩了它一脚，使时间被更改了（如果你觉得这事难以置信，先不要笑——这事就曾经发生在我身上。我为找到一只不怕猫踩的闹钟花费了很长时间）。只要在按钮的结构上做点细微的改动，就可以避免闹钟的时间被猫改掉，那么你就可以按时起床，根本不需要这么匆忙。

总而言之，假如人们在设计产品的时候多花点心思，上文所说的每件倒霉事都是可以避免的。这些例子都表明，我们对**用户体验（user experience）**的关注实在是太少了：产品在现实世界中是如何发挥作用的，人们又是如何使用它们的。在产品开发过程中，人们更多地注意产品将被用来做什么。用户体验是经常被忽略的另一个因素——即产品如何工作——而这一因素恰恰是决定产品成败的关键因素。

用户体验并不是指一件产品本身是如何工作的（虽然这有时对用户体验具有很大的影响）。用户体验是指“产品如何与外界发生联系并发挥作用”的，也就是人们如何“接触”和“使用”它。这种交互通常包括各种各样的按钮，它们体现在以上案例中的一些技术产品上，比如闹钟、咖啡机和收银机。有时，交互只是体现在一个简单的物理装置上，比如汽车的油箱盖。不管怎么样，人们使用的每一件产品都具有用户体验：报纸、调味罐、活动靠背椅、开襟羊毛衫。





无论什么产品，用户体验总是体现在细微之处，但却非常重要。按下按钮时的“滴答”声似乎无关紧要，但如果这个声音决定了你是否能喝到咖啡，那么它就变得很关键了。即使你从来没有意识到是这个按钮的失败设计给你带来了麻烦，但你可以想一想，你对一个总是用不好的咖啡机印象如何？你对生产这个咖啡机的厂家印象如何？你还会再购买他们生产的产品吗？也许不会了吧。这样，由于按钮不能发出声音，厂家失去了一个顾客。

## 用户体验和网站

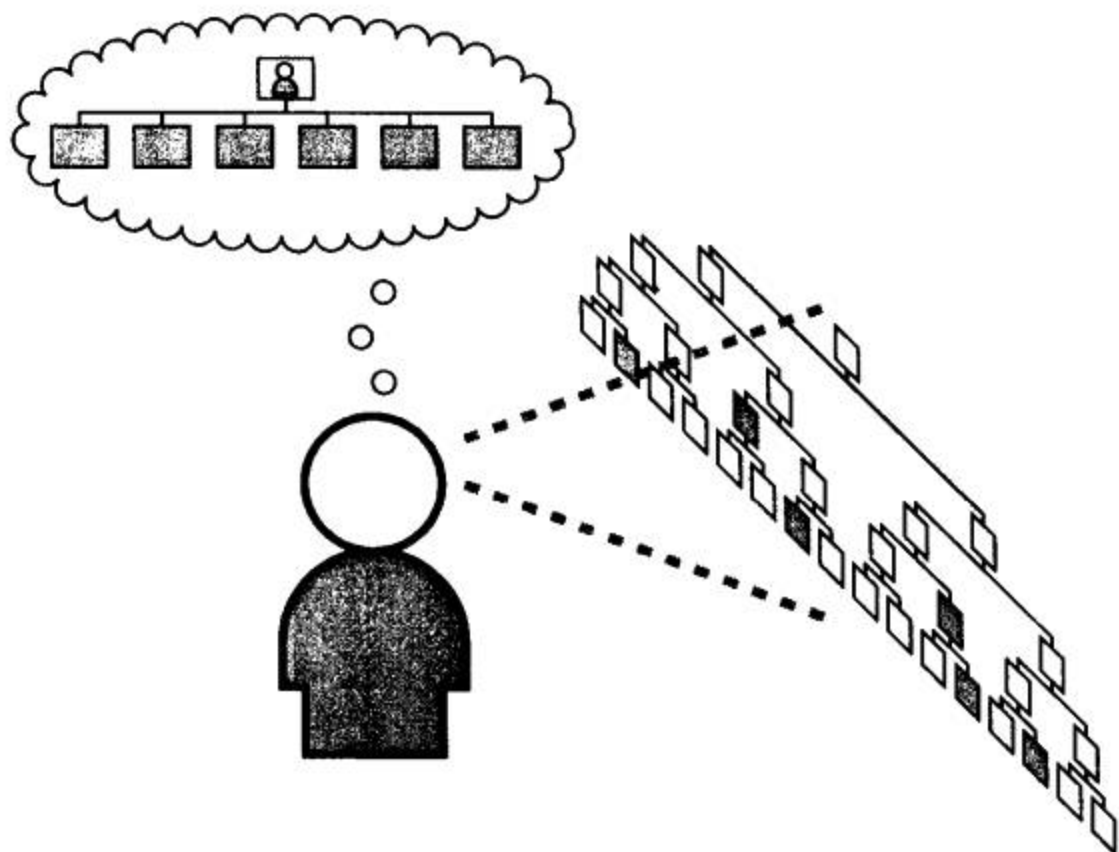
这本书讨论的是一种特殊产品的用户体验：网站。在网站上，用户体验比任何一个其他产品都显得更重要。

不管用户访问的是什么类型的网站，它都是一个“自助式”的产品。没有可以事先阅读的说明书，没有任何操作培训或讨论会，也没有客户服务代表来帮助她了解这个网站。她所能依靠的只有自己的智慧和经验，来独自面对这个网站。

她被困在了某个地方，必须要靠自己找到出路，这已经够糟糕了；而大部分网站甚至都没有意识到她的困境，这只会让事态变得更糟。无论用户体验对网站的成功具有多么重大的战略意义，在大多数网站发展的过程中，仅仅是“去理解人们所想和所需”这样一件简单的事，都从来没有得到过重视。



面对大量的选择，用户只能自己想办法，去决定哪一个网站功能会符合她的需求。



这些是如何发生的呢？在网站最初出现的时候，成功的关键是第一时间把各种想法推向市场。就像Yahoo!一样，早早地建立起很多个“第一”，使得后来的竞争对手不得不努力追赶。接着一些公司竞相建起了网站，希望不再被视为落后于时代。但是在大多数情况下，企业只是把拥有网站看成是一个杰出的成就，至于它是否真正能为人们所用，充其量是后来才想到的一件事情。

为了赢得市场份额，对付那些“捷足先登”的网站，企业开始强调“产品特性”，向他们的网站中加入越来越多的内容和功能，希望可以吸引一些刚开始上网的人（也许还能从竞争对手那里偷过来一些客户）。





无论如何，拥有更多的“产品特性”，被证明只能保持短时间的竞争优势。随着功能的不断膨胀，网站变得越来越复杂、越来越笨重、越来越难以使用，最后就失去了对初次访问者应有的吸引力。同时，企业仍然很少去关心用户真正喜欢什么，很少去发现有价值、或真正可以使用的东西。

企业已经开始意识到，提供优质的用户体验是一个重要的、可持续的竞争优势。用户体验形成了客户对企业的整体印象，界定了企业和竞争对手的差异，并且决定了客户是否还会再次光顾。

## 竞争优势和投资回报率

也许你在网站上并不销售任何东西。你所提供的全部内容都是关于企业的。看上去你对这些信息拥有垄断权——如果人们想得到它，那么必须从你那儿得到。你不像一个在线书店那样有很多的竞争对手。尽管如此，也不能忽视用户在你的网站上的体验。

如果你的网站主要是由我们称为“内容”的网站类型组成——也就是说，信息——那么网站的主要目标之一，就是尽可能有效地传达那些信息。仅仅把它们放在那儿是远远不够的。它必须用一种能帮助人们理解和接受的方式呈现出来。否则，用户永远不会发现你所提供的服务或产品正是他们在寻找的。即使他们能找到这些信息，用户也很可能得出一个结论：你的



网站很难使用，恐怕你也一样。

即使你的网站包括了一些交互的小工具，从而用来帮助人们完成某个任务（比如购买机票或管理银行账户），高效的沟通仍然是决定产品是否成功的关键因素。如果用户弄不明白它是怎样工作的，你所提供的“世界上最强大的功能”将摇摇欲坠，并以失败而告终。

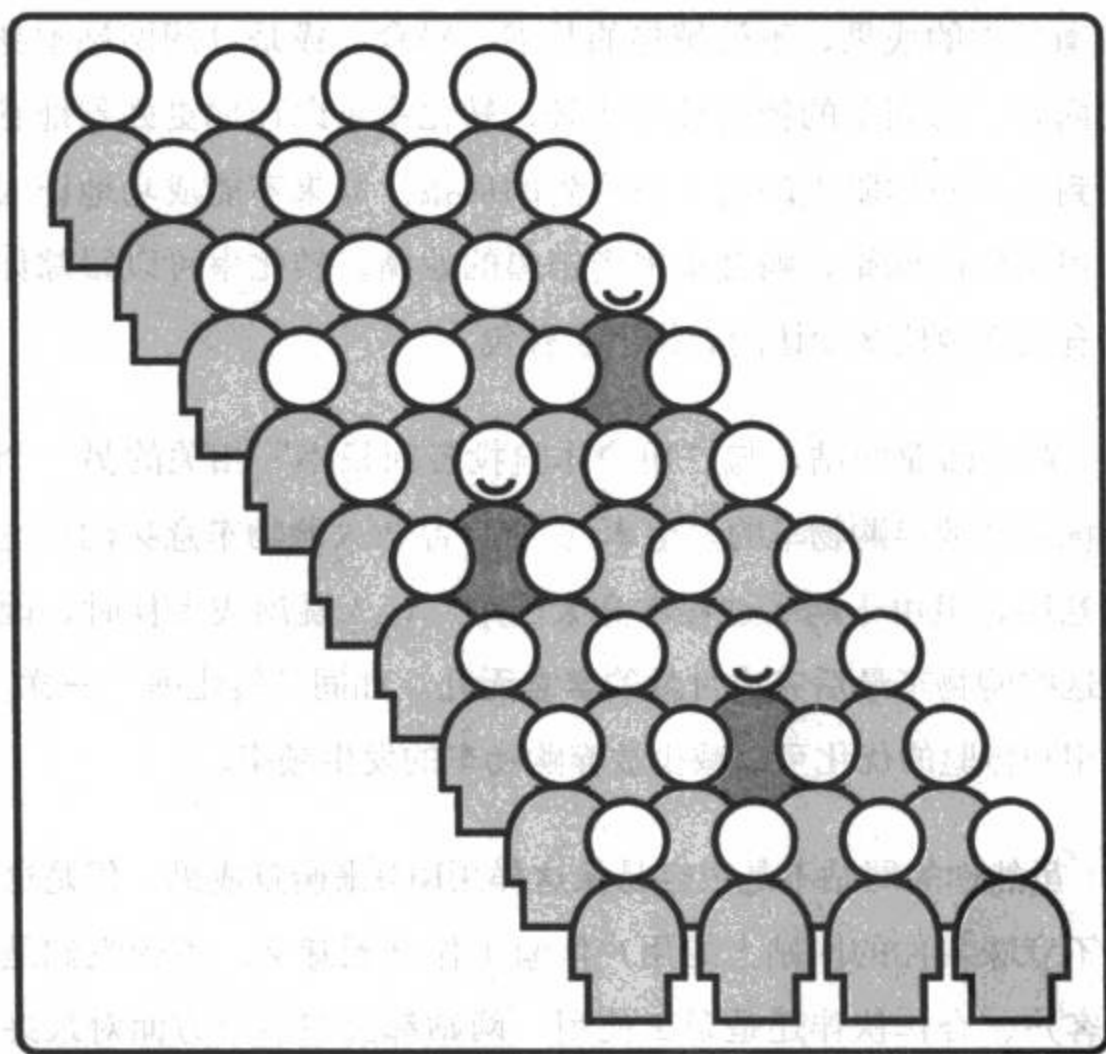
简而言之，如果用户得到一次不好的体验，那么他们将不再回来。如果用户在网站上体验尚好，但是，在你的竞争对手那儿感觉更好，那么他们下次将访问竞争对手的网站，而不是你的。“特性”和“功能”总是重要的，但是用户体验对于客户的忠诚度有着更大的影响。所有的尖端技术和企业营销邮件都无法让这些客户再访问第二次。一次良好的用户体验就可以——而且你不会得到太多的“第二次机会”去纠正它。

“客户忠诚度”并不是关注网站的用户体验所带来的唯一的回报。企业关注财务盈收的情况，希望知道投资所得到的回报（return on investment），或投资回报率（ROI）。投资回报率通常是用金钱来衡量的：所花出去的每一元钱，能收回多少元的等值收益？这就是投资回报率。但是投资所得的收益并不是一定要用严格的货币术语来进行计算。你所需要的是一种衡量的方法，用于计算“花出去的钱”转化成了多少“企业的价值”。





一个最常用的投资收益的度量标准是转化率 (conversion rate)。任何时候，你都希望鼓励你的客户在和你建立某种关系的时候采取更进一步的行动——不管是和“用户定制个人偏好网站”一样复杂、还是和“登录并接受Email简讯”一样简单的行动——你总是可以用转化率来衡量这个结果。通过跟踪有百分之多少的用户被“转化”到了下一个步骤，就能衡量你的网站在达到“商业目的”方面的效率有多高。



转化率是一种常用的方式，来衡量用户体验的效果。

$$\begin{array}{r} 3 \text{ 个注册并订阅邮件的用户} \\ \div \\ 36 \text{ 个访问者} \\ = \\ 8.33\% \text{ 的转化率} \end{array}$$



转化率对于商务网站来讲尤为重要。浏览商务网站的人数远远多于从它那里购买产品的人数。一次优质的用户体验是将这些“偶然浏览者”转化成“实际购买人”的关键因素。即使只增加了很少的一点转化率，它也能使你的财政收入得到一次重大的飞跃。将1%的转化率提高十分之一，从而导致财政收入增加10%甚至更多，这并不是很罕见的现象。

在任何一个用户有可能掏腰包的网站上，都可以用转化率来衡量你的成果，不管销售的是书、猫食，或是订阅网站本身的内容。与简单的销售数字比较，转化率可以让你更强烈地感受到在用户体验上的投入所得到的回报。如果不能成功地让人们理解你的网站，则会损害到销售的业绩。转化率可以跟踪你究竟成功地使多少访问者掏出了腰包。

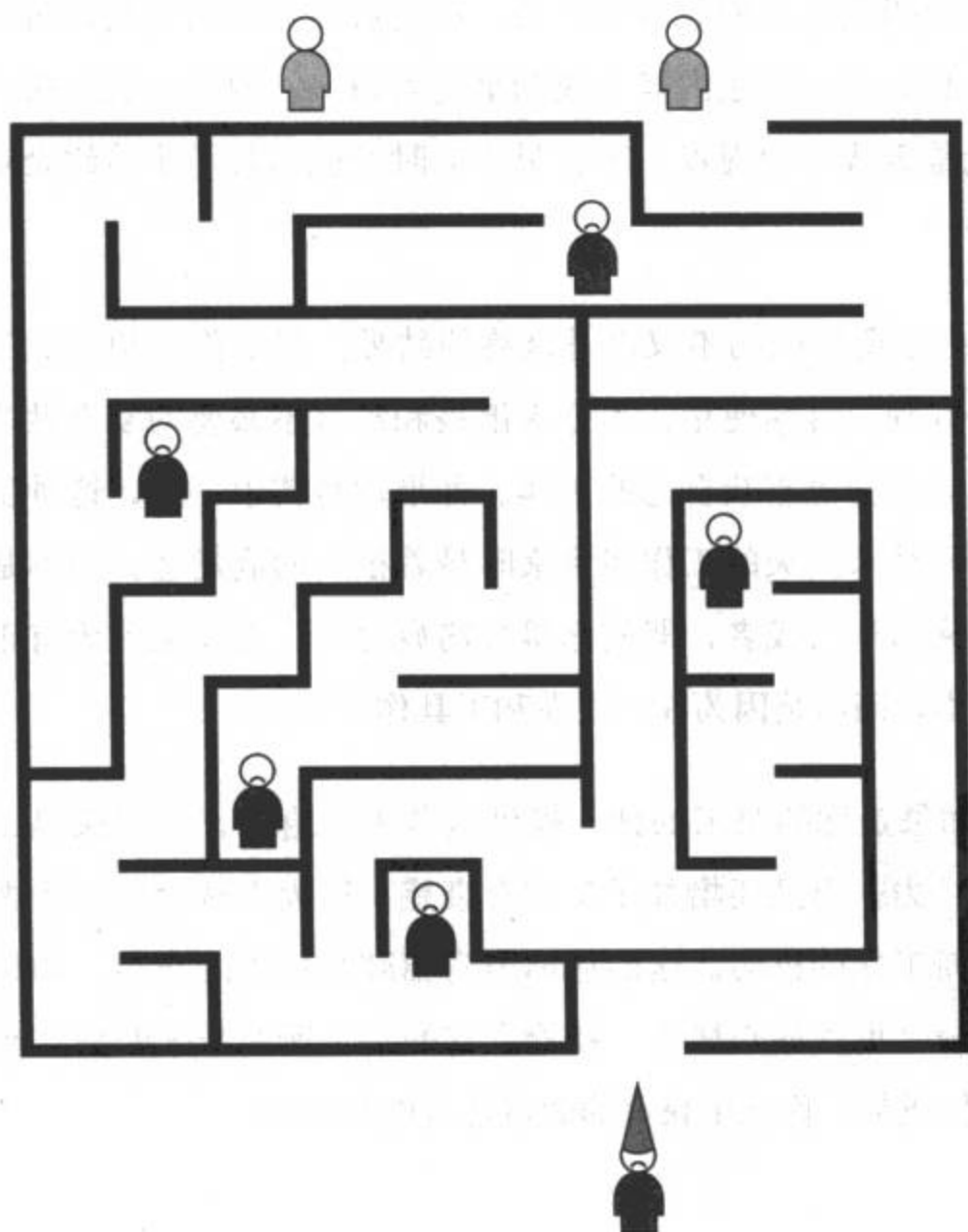
对于商业网站，与“用户体验投资回报率”相关的另一个指标是“放弃购物车的发生率”。把产品放入购物车意味购买它的意愿，但由于购买过程本身太困难、让人疑惑或太耗时，因此这些购物车最后完成付款的寥寥无几。如同“转化率”一样，对用户体验的优化可以减少放弃购物车的发生频率。

虽然你的网站不是很容易像这样用ROI来衡量成果，但是这并不意味着你的网站上的用户体验工作毫无意义。不管它们是被客户、合作伙伴还是员工使用，网站都会在各个方面对最终结果起到间接的影响。





网站是一门错综复杂的技术，而当人们使用这些错综复杂的技术遇到困难时，有趣的事情就发生了：他们总是责备自己。他们认为自己一定做错了什么。他们觉得自己很愚蠢。可以肯定地说，这是不理性的。毕竟，网站没有按照他们所期望的那样运作，这不能算用户的错。但是用户仍然觉得自己很笨。如果想把人们从你的网站上赶跑，那就在他们使用网站的时候让人们觉得自己愚蠢吧，你很难再找出比这个更有效的办法了。



科技产品没有按照人们期望的那样工作，这让他们觉得自己很笨——即使他们最终完成了自己想做的事情。



即使你网站不会被企业之外的人看到（比如企业的内部网站），但是用户体验仍然会造成很大的差异。这是一个“为企业创造价值的项目”和另一个“变成资源消耗噩梦的项目”之间的差异。

任何在用户体验上所做的努力，目的都是为了提高效率。这基本上是以两种主要形式体现出来的：“帮助人们工作得更快”和“减少他们犯错的几率”。所使用的工具效率的改进，会直接提高企业的整体生产力。在完成任务时所花费的时间越少，那么在一天之内能完成的事情就越多。根据一直以来“时间就是金钱”的观点，节省员工的时间就直接等于节约企业的金钱。

效率所影响的不仅仅是最终的结果。当工作中用到的工具合乎规则、容易使用、不令人沮丧和没有不必要的复杂性的时候，人们会更喜欢自己的工作。如果你是其中一员，这种工具会在你结束一天的工作回到家时带着很大的满足感，而不是累得筋疲力尽（或者，即使你累得筋疲力尽，至少是因为有正当的理由，而不是因为你一直在和工具作斗争）。

如果是你的员工的话，提供这种工具给他，不仅仅提高了的生产力，而且还增加了工作自豪感，这大大减少他们考虑换一个新工作的机会。这能让你节省招聘和培训的费用，如果再算上一个更加尽心尽力、经验丰富的员工所能带来的更高质量的工作成果，你从中获得的效益是显而易见的。





## 记住你的用户

创建吸引人的、高效的用户体验的方法被称为以用户为中心的设计 (user-centered design)。以用户为中心的设计思想非常简单：在开发产品的每一个步骤中，都要把用户列入考虑范围。这个简单思想之下的含义，却出乎意料的复杂。

用户所体验的每一件事，对你来讲都应该是经过慎重考虑和论证以后的决定。实际上，设计出一个更好的解决方案需要更多的时间和费用，你可能不得不在各个方面做出妥协。但是，一个“以用户为中心”的设计流程保证了这些妥协不是随机决定的。考虑用户的体验、把它分解成各个组成要素、从不同的角度来了解它——通过这些才能确保你控制了决策所造成的全部结果。

用户体验对你很重要，其中一个最大的理由是：它对你的用户很重要。如果没有给他们一个积极的体验，他们不会使用你的网站。如果没有用户，你能得到的只是一台藏在某个角落里、布满了灰尘的网络服务器，无聊地等待着去完成永远不会到来的请求。对于那些来造访的用户，你必须为他们规划一个协调一致、直观明了、甚至还让人愉快的体验——一次“每件事都按照正确的方式在工作”的体验。不管他们这一天的其他时间是如何度过的。





# 第2章

## 认识这些要素





用户体验的整个开发流程，都是为了确保用户在你的网站上的所有体验不会发生在你“明确的、有意识的意图”之外。这就是说，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。这听上去像是一个很庞大的工作，而且在某种程度上来讲也的确是。但是，我们可以把设计用户体验的工作分解成各个组成要素，以帮助我们更好地了解整个问题。

## 五个层面

大多数人都曾经通过网站购买过一本书。这种经历几乎每一次是一样的——你到达网站，寻找你想买的书（也许使用搜索引擎，也许使用分类目录），把你的信用卡号和邮寄地址告诉网站，然后网站则保证这本书将递送到你的手中。



这个清晰、有条不紊的体验，事实上由一系列完整的决策——大大小小的决策——组成的：网站看起来是什么样、它如何运转、它能让你做什么。这些决策彼此依赖，告知并影响用户体验的各个方面。如果我們去掉这些体验的外壳，就可以理解这些决策是如何做出来的了。



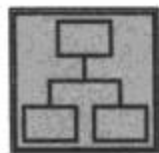
### 表现层

在表现层 (surface)，你看到的是一系列的网页，由图片和文字组成。一些图片是可以点击的，从而执行某种功能，例如把你带到购物车里去。一些图片就只是图片，比如一本书的封面或网站自己的标志。



### 框架层

在表现层之下是网站的框架层 (skeleton)：按钮、表格、照片和文本区域的位置。框架层用于优化设计布局，以达到这些元素的最大的效果和效率——使你在需要的时候，能记得标识并找到购物车的按钮。



### 结构层

与框架层相比更抽象的是结构层 (structure)，框架则是结构的具体表达方式。框架层确定了我们的结账页面上交互元素的位置；而结构层则用来设计用户如何到达某个页面，并且在他们做完事情之后能去什么地方。框架层定义了导航条上各项的排列方式，允许用户可以浏览书籍的不同类别；结构层则确定哪些类别应该出现在那里。





## 范围层

结构层确定网站各种特性和功能的最合适的组合方式，而这些特性和功能就构成了网站的范围层（scope）。有些卖书的网站提供了一个功能，使用户可以保存之前的邮寄地址，这样他们可以再次使用它。这个功能——或任何一个功能——是否应该成为网站的功能之一，就属于范围层要解决的问题。



## 战略层

网站的范围基本上是由网站战略层（strategy）所决定的。这些战略不仅仅包括了经营者想从网站得到什么，还包括了用户想从网站得到什么。就我们的网上书店的例子而言，一些战略目标是显而易见的：用户想要买书，我们想要卖出它们。另一些目标可能并不是那么容易说清楚的。

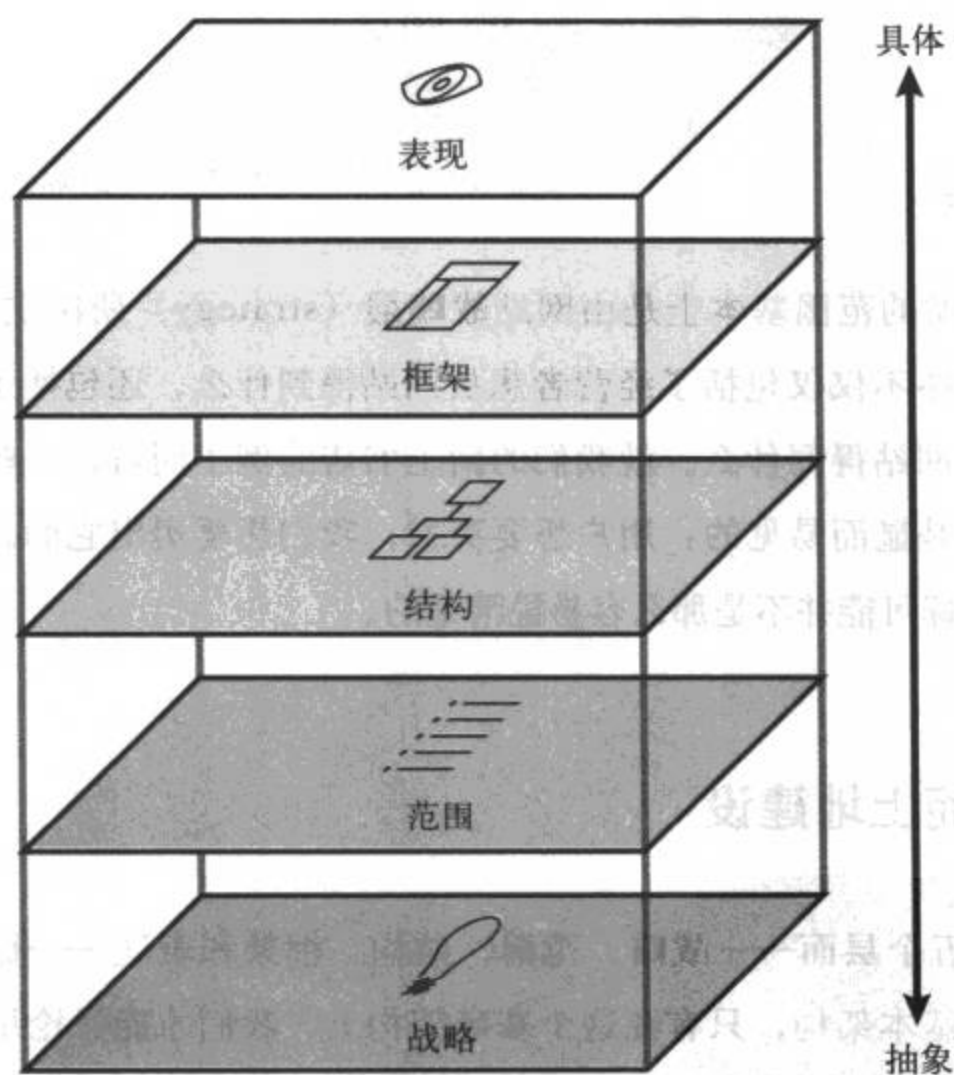


## 自下而上地建设

这五个层面——战略、范围、结构、框架和表现——提供了一个基本架构，只有在这个基础架构上，我们才能讨论用户体验的问题，以及用什么工具来解决用户的体验。

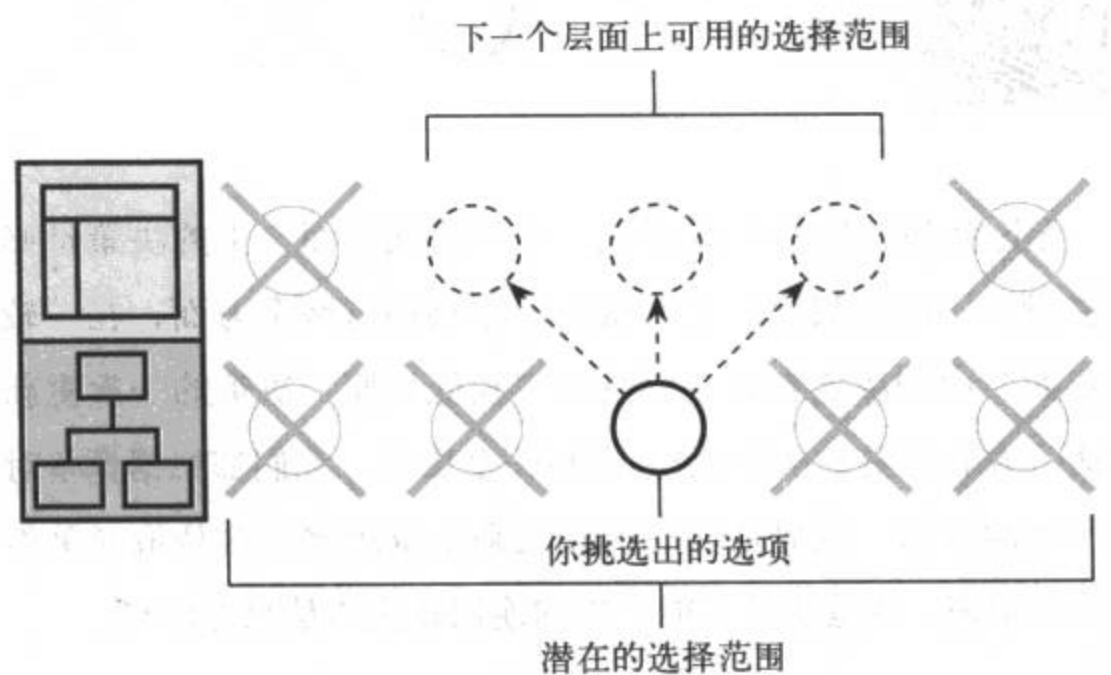


在每一个层面中，我们要处理的那些问题不是抽象的，它们会变得更加具体。在最低的层面，我们完全不用考虑网站最终的外观——我们只关心网站如何满足我们的战略（同时也满足用户的需求）。在最高的层面，我们只关心网站所呈现的最具体的细节。随着层面的上升，我们要做的决策就一点一点地变得具体，并关系到越来越精细的细节。



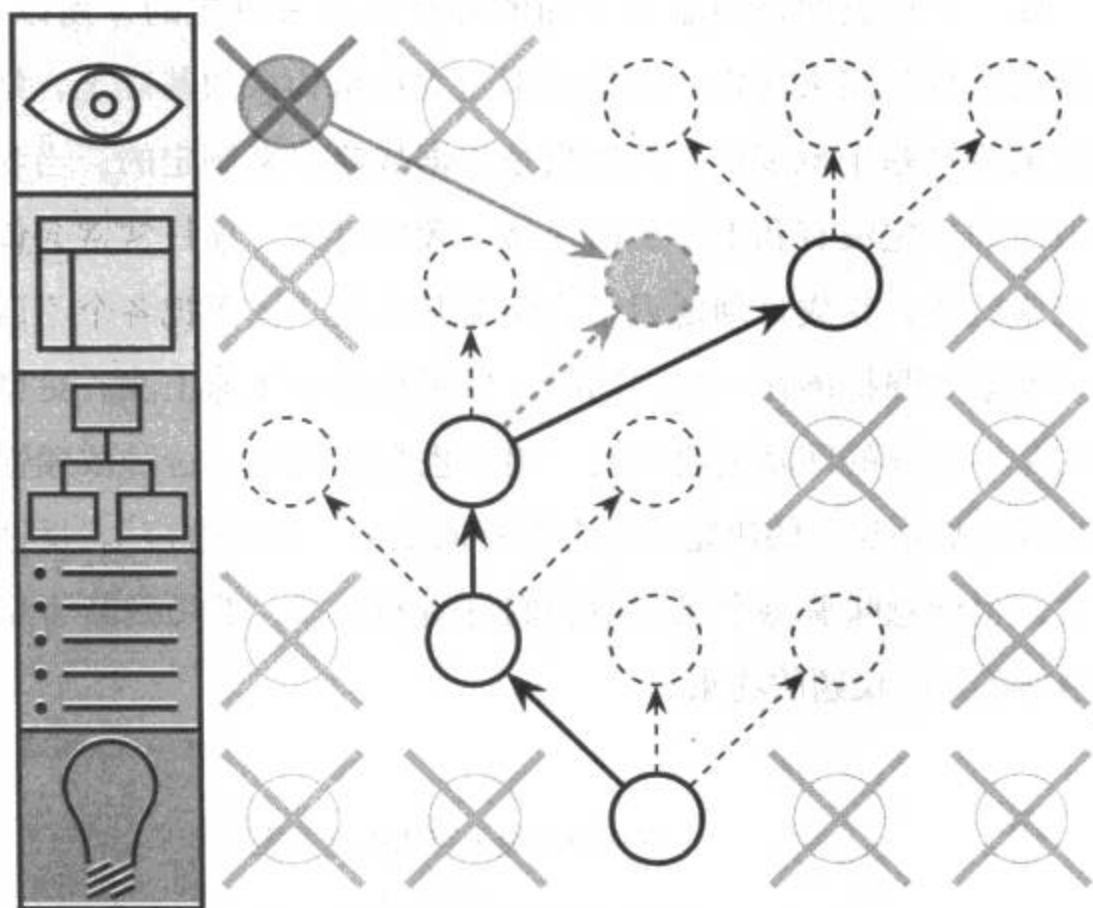


每一个层面都是根据它下面的那个层面来决定的。所以，表现层由框架层来决定，框架层则建立在结构层的基础上，结构层的设计基于范围层，范围层是根据战略层来制定的。当我们做出的决定没有和上下层面保持一致的时候，项目常常会偏离正常轨道，完成日期延迟了，而在开发团队试图把各个不匹配的要素勉强拼凑在一起的同时，费用也开始飞速上涨。更糟糕的是，这样的网站上线以后，用户也会痛恨它。这种依赖性意味着在战略层上的决定将具有某种向上的“连锁效应”。反过来讲，也就意味着每个层面中我们可用的选择，都受到其下层面中所确定的议题的约束。



在每一个层面的决定都会影响到它之上层面的可用选项。

这种连锁效应意味着在“较高层面”中选择一个界限之外的选项将需要重新考虑“较低层面”中所做出的决策。



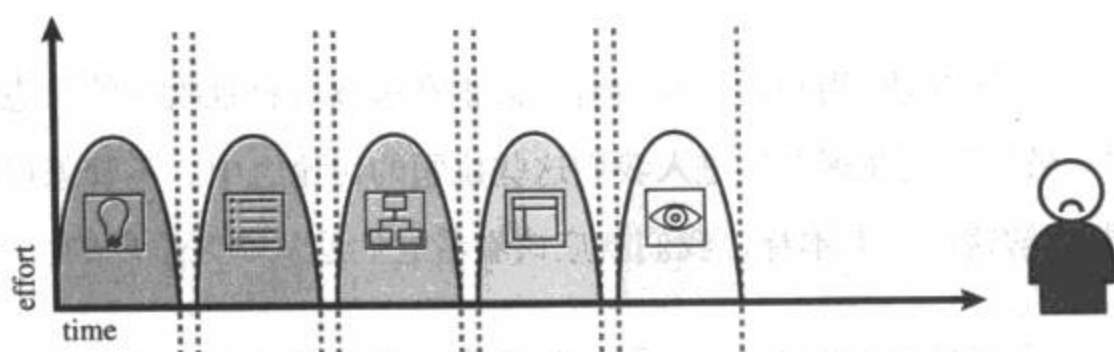
无论如何，这并不是说每一个“较低层面”上的决策都必须在设计“较高层面”之前做出。事物都有两个方面，在“较高层面”中的决定有时会促成对“较低层面”决策的一次重新评估（或者是第一次评估）。在每一个层面，我们都根据竞争对手所做的事情、业界最佳的实践成果来做决定，这是最简单不过的老常识。这些决策可能产生的连锁效应该是双方向的。

如果想在开始“较高层面”设计之前，要完全确定“较低层面”的话，几乎可以肯定的是，你已经把你的项目日程——也许是你最终产品的成功——扔进了一个危险的境地之中。

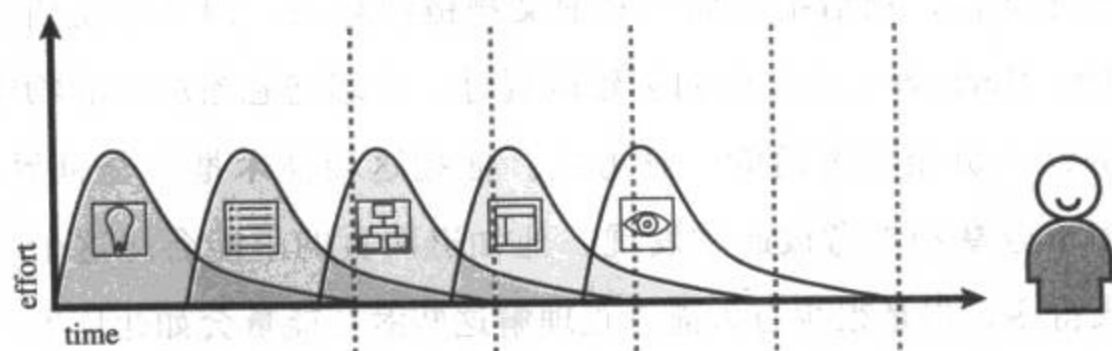




相反地，应该计划好你的项目，让任何一个层面中的工作都不能在其下层面的工作完成之前结束。这里最重要的一条是，在我们知道基本形状之前，不能为房屋加上屋顶。



要求每个层面的工作在下一个层面可以开始之前完成，会导致你和你的用户都不满意的结果。



一个更好的方法是让每一个层面的工作在下一个层面可以结束之前完成。

## 基本的双重性

当然，用户体验的要素肯定不止这五个，与任何专业领域一样，这个行业也有它自己的专用术语。对于任何一个偶然进入这个领域的人来讲，用户体验可能显得非常复杂。所有看上去很相似的词汇有：交互设计、信息设计、信息架构。它们都是什么意思？做的是什​​么？或者它们仅仅是一些毫无意义的行业术语吗？

为了进一步把事情复杂化，人们会用不同的方式来使用同样的词汇。某个人也许用“信息设计”来描述别人口中说的“信息架构”，而“界面设计”和“交互设计”有什么不同呢？它们是一件事情吗？

幸运的是，用户体验领域看上去正在远离这种乱哄哄的状态。“一致性”正在逐渐地进入我们这些议题的讨论之中。不管怎样，要理解这些词汇本身，我们就应该看看它们是从哪儿出现的。

当网站刚刚兴起的时候，它只是一种超级文本。人们可以创建文档，然后把它们与其他文档链接起来。网页的发明者 Tim Berners-Lee 在最初创建网页时，是想把它当成高能物理研究人员相互沟通的一种方式。他把这项技术推向全世界，用于分享和参考彼此的发现。他知道网页的潜能会比这个要大得多，但是很少有人能真正理解这些潜在能量会如此巨大。

人们最初把网页当成一种新的出版媒介，但随着先进技术和新特性不断被加入到网页浏览器和服务端之中，网页开始拥有更多的新能力。在网页开始变得流行，并成为更大的互联网社区之后，它发展了更复杂和强大的一些功能，这些功能使网站不仅仅能传达信息，还能收集和控制这些信息。有了这些功能，网站变得更加互动，响应用户输入的方式也更象传统的客户端应用程序。

随着商业利益在网站上的出现，这些功能找到了一个更大的应用范围，比如电子商务、交流论坛，其中还包括了网上银



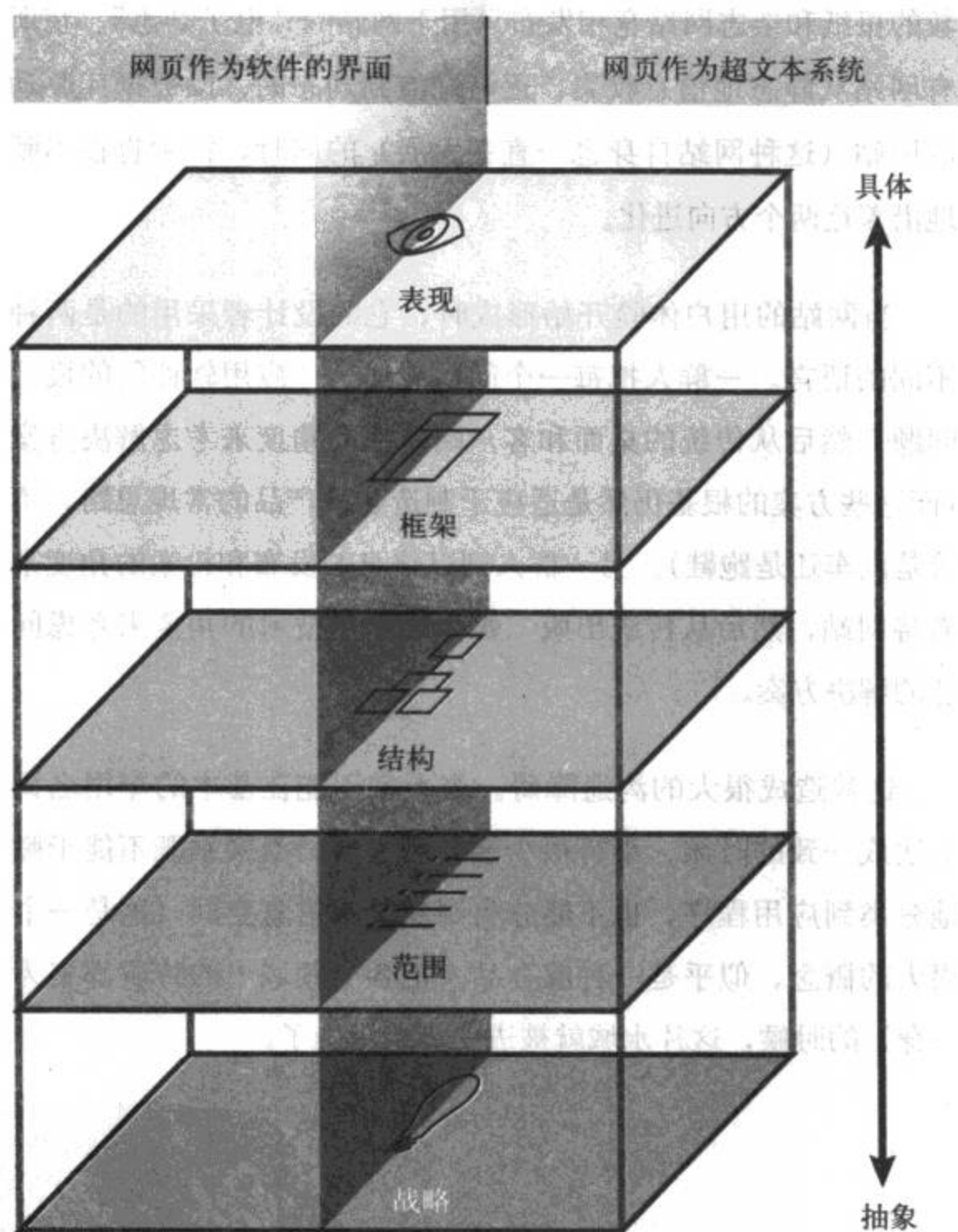


行。与此同时，网页作为一种出版媒介继续蓬勃发展，不计其数的报纸和杂志网站竞相发布只用于网站的“电子杂志”。在所有网站从静态地信息收集，逐渐过渡到动态的、以数据库驱动的网站（这种网站自身也一直在发展）的同时，技术也在不断地沿着这两个方向进化。

当网站的用户体验开始形成时，它的设计者采用的是两种不同的语言。一群人把每一个问题看成是“应用软件”的设计问题，然后从传统的桌面和客户端软件的角度来考虑解决方案（而这些方案的根基仍然是适应于制造各种产品的常规思路，不管是汽车还是跑鞋）。另一群人则以信息的发布和检索的角度来看待网站，然后从传统出版、媒体和信息技术的角度来考虑问题的解决方案。

这就造成很大的沟通障碍。当大家不能在基本的专用名词上达成一致的时候，事情很少能顺利进展。当网站既不能干脆地分类到应用程序，也不能分类到超文本信息空间（这是一个很大的概念，似乎是一种混合品，把各个领域中的特质都集为一身）的时候，这片水域就被进一步地污染了。







为了解决网页这种基本的双重性质，让我们从中间把这五个层面分开。在左边，这些要素仅用于描述软件界面类型的网页。在右边，我们将把这些要素用于超文本信息空间。

在软件的一边，我们主要关注的是任务——所有的操作都被纳入一个过程，去思考人们如何完成这个过程。在这里，我们把网站看成用户用于完成一个或多个任务的一个或一组工具。

在超文本的一边，我们的关注点是信息——网站应该提供哪些信息，这些信息对用户的意义是什么。超文本就是创建一个“用户可以穿越的信息空间”。

## 用户体验的要素

现在，我们可以把整个混乱的词汇集放到这个模型里了。把每一个层面分成各个组成部分，通过这个方法，我们就可以仔细地看看所有这些片段是如何组合在一起，从而形成整个用户体验的。



## 战略层

无论是在软件产品还是信息空间，战略层所关注的内容都是一样的。来自企业外部的**用户需求**（user need）是网站的目标——尤其是那些将要使用我们网站的用户。我们必须要了解这些观众想从我们这儿得到什么，还要知道他们想达到的这些目标将怎样满足他们所期待的其他目标。

与用户需求相对应的，是我们自己对网站的期望目标。这些**网站目标**（site objective）可以是商业目的（通过网站达到今年100万美元的销售收入）或者是其他类型的目标（让选民了解下一届候选人的情况）。在第3章，我们将了解更多这些要素的细节。

## 范围层

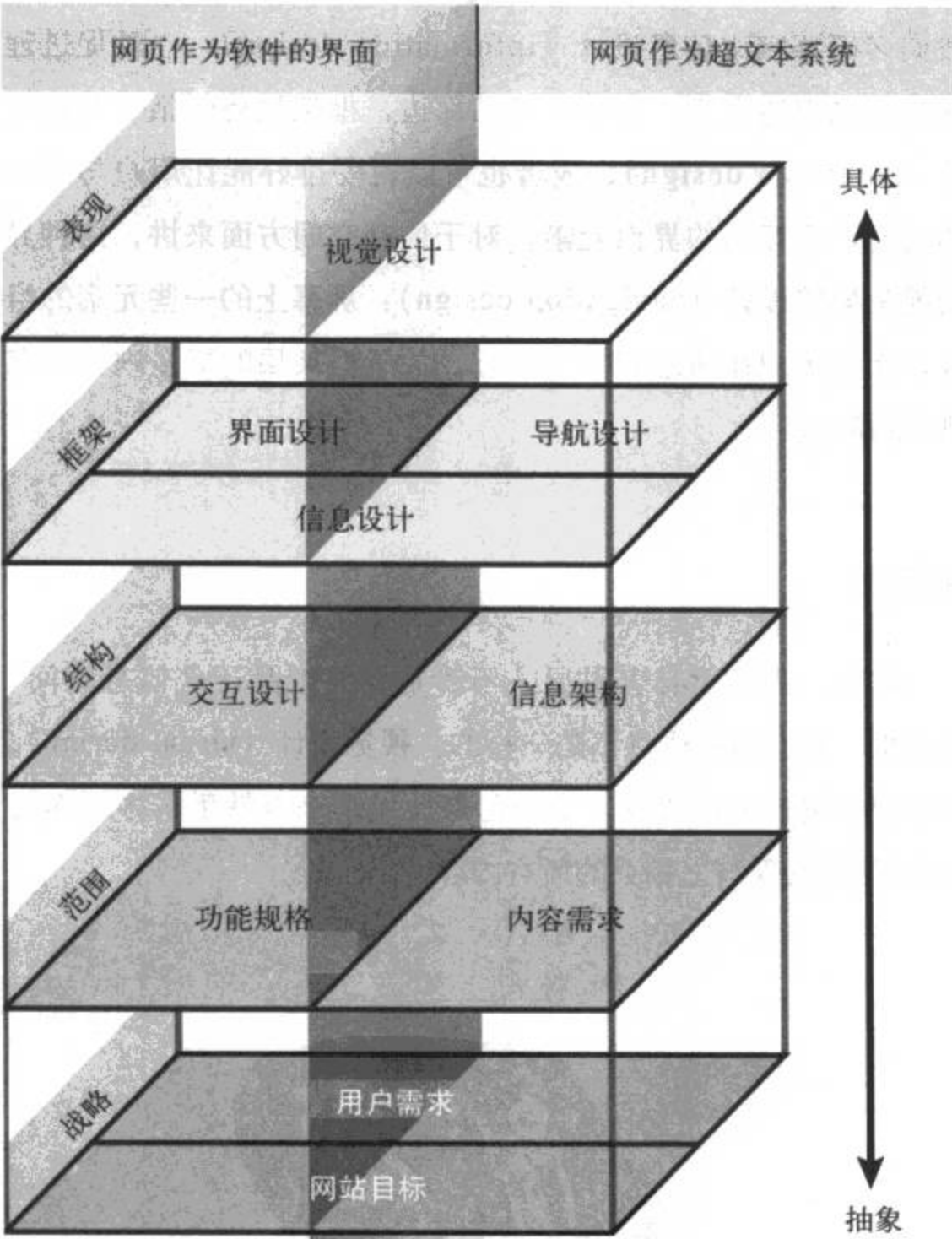
从战略层进入范围层以后，在软件方面它就转变成**创建功能规格**（functional specification）：对产品的“功能组合”的详细描述。而在信息空间方面，范围则是以**内容需求**（content requirement）的形式出现：对各种内容元素的要求的详细描述。第4章将讨论这些范围层的要素。

## 结构层

在软件方面，结构层将从范围转变成交互设计（interaction design），在这里我们可以定义系统如何响应用户的请求。在信息空间方面，结构层则是**信息架构**（information architecture）：在信息空间中内容元素的分布。关于这些你将在第5章了解更多细节。







## 框架层

框架层被分成了三个部分。不管是软件界面还是信息空间，我们必须完成**信息设计**（information design）：一种促进理解的信息表达方式。在软件产品那边，框架层还包括了**界面设计**（interface design），或者说安排好能让用户与系统的功能产生互动的界面元素。对于信息空间方面来讲，这种界面就是**导航设计**（navigation design）：屏幕上的一些元素的组合，允许用户在信息架构中穿行。关于框架层的更多内容将在第6章描述。

## 表现层

最后，我们还有表现层。不管是软件产品还是信息空间，在这里，我们的关注点都是一样的：**视觉设计**（visual design），或者说最终产品的外观。它做起来比说起来要棘手得多。你可以在第7章发现与之相关的所有内容。





## 应用这些要素

很少有网站只属于这两部分的任何一个。在每一个层面中，这些要素必须相互作用才能完成该层面的目标。比如，信息设计、导航设计以及界面设计，它们共同定义了网站的框架层。你在某个要素上所做的决定对同一个层面的其他要素是不可能没有影响的。所有处在同一个层面中的要素都具有相同的功能——在这个例子里，就是“定义网站的框架”——即使它们是通过不同的方式。

这种把用户体验划分成各个方块和层面的模式，非常有利于我们去考虑用户在体验中有可能遇到的麻烦。但是在现实世界中，这些区域之间的界限并没有那么明确。最常见的情形是，你很难鉴定某个用户体验的问题是否可以通过重视这个要素或那个要素去解决。是在视觉设计上玩一些小把戏就可以呢，还是要改造最基本的导航设计？某些问题要求同时重视多个区域，而另一些好像就横跨在这个模型中各个要素的边界上。

这样的组织方式常常使“设计用户体验”这件事进一步复杂化了。在一些企业中，你会遇到一些被称为“信息架构师”或“交互设计师”的人。不要被这个现象搞糊涂了。这些人一般都具有很多种专业技能，这些技能包括大多数与用户体验的要素有关的领域，而不仅仅是他们的职位名称所表明的那些内容。你的团队里，不一定非要一个了解各个领域的专家，你只需要保证有专人在负责考虑每一个议题就行了。



还有两个额外的因素，它们将会对最终的用户体验产生影响，但是在这里无法详细描述。首先就是内容（content）。古语说道（嗯，在网页刚出现的时候）：在网页里“内容至上”。这绝对是个真理——大多数网站能提供给他们的用户的最重要的一件东西，就是这些用户认为有价值的内容。

用户不会仅仅为了体验导航的乐趣而访问网站。你可以得到的内容（或你有资源去得到和管理的内容）将在你的网站中扮演一个非常重要的角色。在之前网上书店例子中，也许我们决定让用户看到被出售的所有书籍的封面。如果能得到这些封面的话，我们有某种方法可以给它们分类吗？可以跟踪它们的变化，以及保持更新吗？如果某本书的封面我们根本就拿不到手，这种情况要怎么办呢？这些内容问题，对用户在网站上的最终体验非常重要。





其次，技术也像内容一样，对于建立一个成功的用户体验很重要。在大多数案例中，你能提供给用户的体验状态主要是由技术来决定的。在网页刚出现的时候，把网站和数据库连接起来的工具相当原始，而且非常有限。但不管怎样，随着技术的发展，数据库被更加广泛地用于网站的驱动和控制。这反过来又使得越来越精细的用户体验方法变得可能，例如动态的导航系统，就是一种根据用户在网站中的移动来改变导航的技术应用。技术总在变化，用户体验的领域必须要适应这些变化。虽然如此，用户体验的要素始终是不变的。

这本书以下的内容将分层面讨论这些要素，非常详细。我们将仔细了解一些用于处理每一个要素的工具和技术。我们将知道每个层面中哪些要素是共同的，是什么让它们各不相同，以及它们是如何相互影响，然后创建一个总体的用户体验的。





# 第3章

## 战略层

网站目标和用户需求





设计过程中，设计师需要根据用户的需求，结合企业的实际情况，制定出一套完整的设计方案。在设计过程中，设计师需要不断地与用户进行沟通，了解用户的需求，并根据用户的需求进行设计。同时，设计师还需要不断地进行自我反思，不断改进自己的设计，以达到最佳的设计效果。

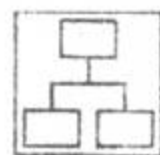
成功的用户体验，其基础是一个被明确表达的“战略”。知道企业与用户双方对网站的期许和目标，有助于确立用户体验各方面战略的制定。然而回答这些看似简单的问题却不如说起来那么容易。



表现层



框架层



结构层



范围层



战略层



## 战略层定义

网站失败的最常见原因不是技术，也不是用户体验。网站失败的最常见原因，是在开始写第一行程序、描第一个像素，或配置第一个服务器之前，没有人试图回答下面两个非常基本的问题：

- 我们要从这个网站得到什么？
- 我们的用户要从这个网站得到什么？

通过回答第一个问题，我们就能描述出企业的网站目标（site objective）。第二个问题则提出了用户需求（user need）的问题，这是来自企业外部的目标。结合内外两者，网站目标和用户需求就组成了战略层，也就成为我们设计用户体验过程中做出每一个决定的基础。然而，令人惊讶的是，许多用户体验项目在开始之初对于最基本的战略层并没有清楚与明确的认识。





此处的关键词是明确。当我们能越清楚明确地表达我们想要什么，以及确切地知道其他人想要从我们这里得到什么，我们能越精确地满足双方的需求。

## 网站目标

为了明确地理解战略，第一步就是检查我们自己的网站目标。网站目标经常以“只可意会不可言传”的状态存在于一小群建站的人当中。当网站目标无法用口头表达出来时，对于应该如何完成项目，不同的人就经常会有不同的想法。

## 商业目标

人们常使用像商业目标 (business goal) 或商业驱动因素 (business driver) 这样的词汇来描述内部的战略目标。我将在本文中使用网站目标，因为我认为其他的词汇不是太狭义就是太广义：太狭义是因为并不是每一个内部目标都是商业目标（毕竟不是每个企业都有像商业那样的同类型目标），太广义则因为这里我们最关心的是用尽可能具体的词汇来定义我们期望网站本身能完成的事情，其余的活动也是如此。

多数人一开始是用很宽泛的词汇来描述网站目标。基本上，企业网站的存在是为了满足两种意图其中的一个：替公司赚钱或替公司省钱。有时它同时满足这两个。但为了这两个目标，网站到底应该做些什么并不总是很清楚。



相反地，太具体的目标也无法充分地描述出在战略制定过程中可能发生的困难。例如，写明你的目标是“提供给用户一个基于Java的实时通信工具”，并不能解释这个工具要如何支持企业目标，或是它如何满足用户需求。

要想在太具体和太宽泛之间取得一个平衡，我们就应该避免在尚未充分了解问题之前就试图得出结论。为了创造成功的用户体验，我们必须保证决策不是随便决定的——也就是说，每一个我们做出的决定都应该是深思熟虑的结果。

### 品牌识别

对于任何一个网站，它需要明确描述的基础目标之一就是**品牌识别**（brand identity）。我们中的大部分人看到“品牌”二字都会直接联想到商标、色调和字体设计。当然这些品牌的视觉组成很重要（我们将会在第7章表现层深入探讨品牌的视觉设计），不过品牌概念远远超越了视觉表现。品牌识别——可以是概念系统，也可以是情绪反应——它之所以重要是因为它无法不被用户注意。在用户与网站交互的同时，企业的品牌形象就不可避免地在用户的脑海中形成了。





你必须要决定品牌形象是无意之中形成，还是经过网站设计者有意精心安排的结果。大多数企业选择对他们的品牌形象施加一些控制，这就是为什么传递品牌识别是非常普遍的一种网站目标的原因。品牌不仅仅影响商业企业——它还影响每一个有网站的组织，从非盈利机构到政府部门，都是靠着用户体验来创造品牌形象。而将品牌形象具体且明确地写进目标，将会提高呈现出积极的品牌形象的机会。

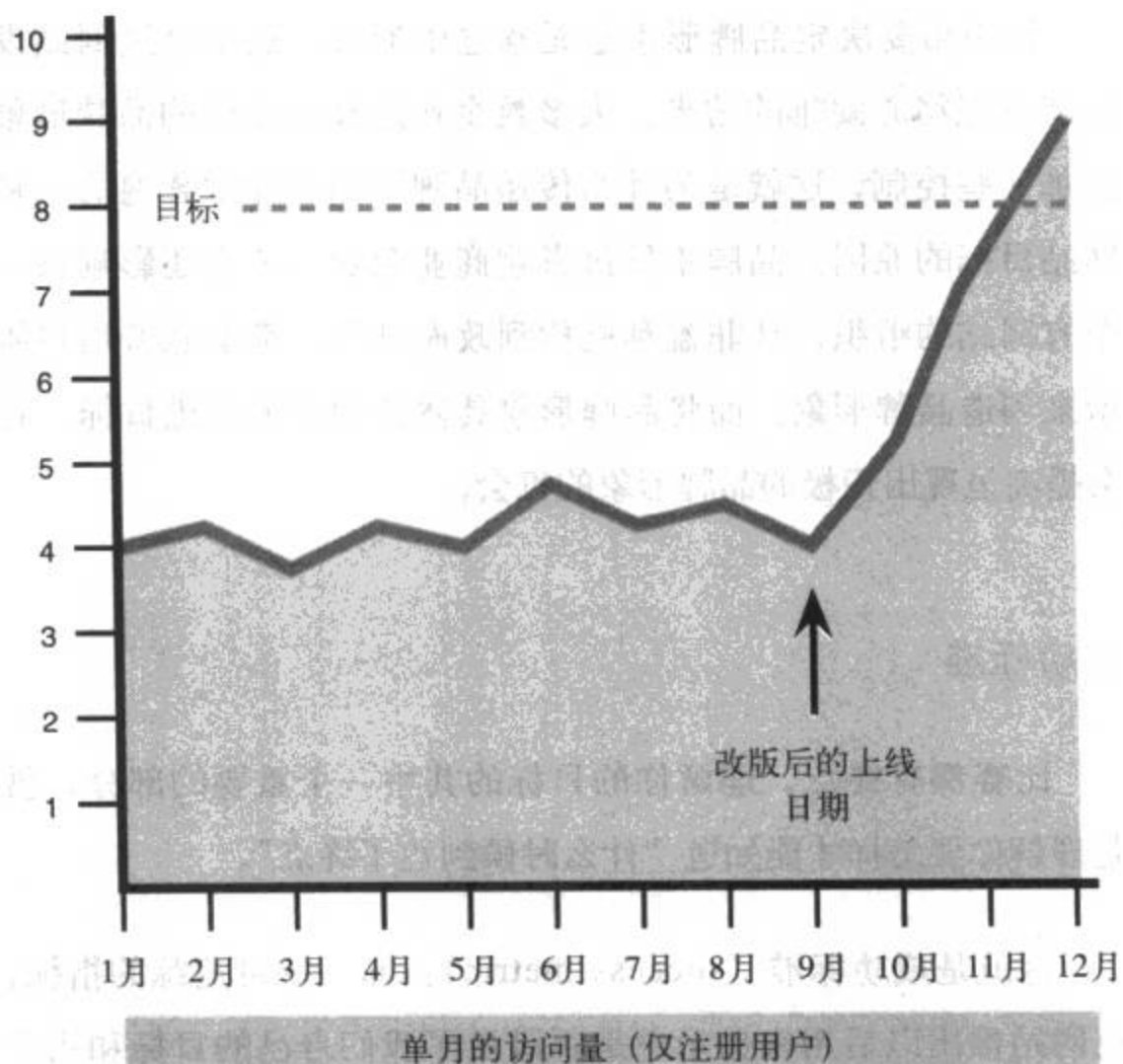
## 成功标准

比赛都有终点。理解你的目标的其中一个重要的部分，就是理解你要怎样才能知道“什么时候到达了终点”。

这就是**成功标准** (success metric)：即一些可追踪的指标，在网站推出以后用来显示它是否满足了我们自己的目标和用户的需求。好的成功标准不仅影响项目各阶段的决策，也提供了用户体验工作价值的具体依据。特别是当你正申请下一个用户体验项目的预算，却发现对方对用户体验的价值表示怀疑时。



成功标准具体显示了用户体验是否有效地达成了战略目标。在此例中,通过衡量每一个注册用户单月的访问次数表明了此网站对核心用户的价值。



有时这些成功标准与网站本身和用户如何使用该站有一定的关系。用户在每一次访问网站时的平均停留时间是多少(服务器记录可以帮助你确定这点)?如果想要鼓励用户随意轻松地发掘网站提供的服务,那么你一定希望看到单次访问的时间有所增加。相对的,如果想要提供快捷简便的信息服,则你或许希望单次访问的时间减少。





对于依赖广告收入的网站，**页面浏览量**（page view）——网站上每一个页面的每天浏览数量——是至关重要的统计信息。但你必须小心地平衡企业目标和用户需求。在主页和用户想看的内容页面之间多增加几层导航，无疑地会增加页面的浏览量，但这是用户需要的吗？可能不是。长期下来，你将看到：由于用户感到挫折与不便，他们决定不再回来，你的页面浏览量将会从升势开始下跌，甚至可能比原来跌得更低。

不是所有的成功标准都必须直接由网站获得。你也可以衡量网站的间接影响。如果你的网站为用户提供产品常见的疑难解答，那么你的客户服务专线的电话数量应该相应减少。一个有效的内部网站可以提供工具与资源的简便的途径，如此则能缩短销售人员的工作时间——也就是直接转化成增加的年收益。

最有效的标准是那些可以直接地归因于网站的用户体验改变的标准。当然，如果网站改版之后，网上交易的每日收益跃升至百分之四十，则很容易可看出其间的因果关联。但是如果这些改变发生在一个更长的时间段，那就很难判定这些改变是源于用户体验还是其他因素。



例如，网站的用户体验无法为站点带来新的用户——你必须依靠口碑或是市场营销来吸引潜在用户。然而用户体验能极大地影响访问者是否会再度访问。测量回访数据可能是一个很好的方法，它能分析你的网站是否满足了用户需求，但是要小心：有时候那些用户不回来是因为你的竞争对手展开了一场声势浩大的广告造势，或因为你的公司目前负面新闻缠身。因此，任何断章取义的标准都可能造成误导，请务必后退一步，看看除了网站之外还发生了什么事，以确定你了解到了事情的全貌。

## 用户需求

我们很容易落入这样的陷阱，即认为我们正在为理想用户而设计网站，理想用户就是“某些与我们完全一样的用户”。但我们并不是为自己设计，而是为其他人设计的，如果想要这些“其他人”喜欢并使用我们创建的东西，那么就必须要了解“他们是谁”以及“他们的需求是什么”。通过投入时间去研究这些需求，我们才能抛弃自己立场的局限，真正从用户的角度来重新审视网站。

确认用户需求是复杂的，因为用户群体之间存在着很大的差异性。即使我们设计的是一个仅供企业内部使用的网站，也仍然需要大范围地考察用户的需要。如果我们的网站用户定位为一般消费者，那需要考虑的各种可能就大大增加了。





## 用户细分

我们可以把大量的用户需求划分成几个可管理的部分，这将通过用户细分（user segmentation）来完成。我们将用户分成更小的群组（或细分用户群），每一群用户都是由具有某些共同关键特征的用户所组成。有多少用户类型几乎就有多少方式来细分用户群，但这里所提及的是几种最常见的方法。

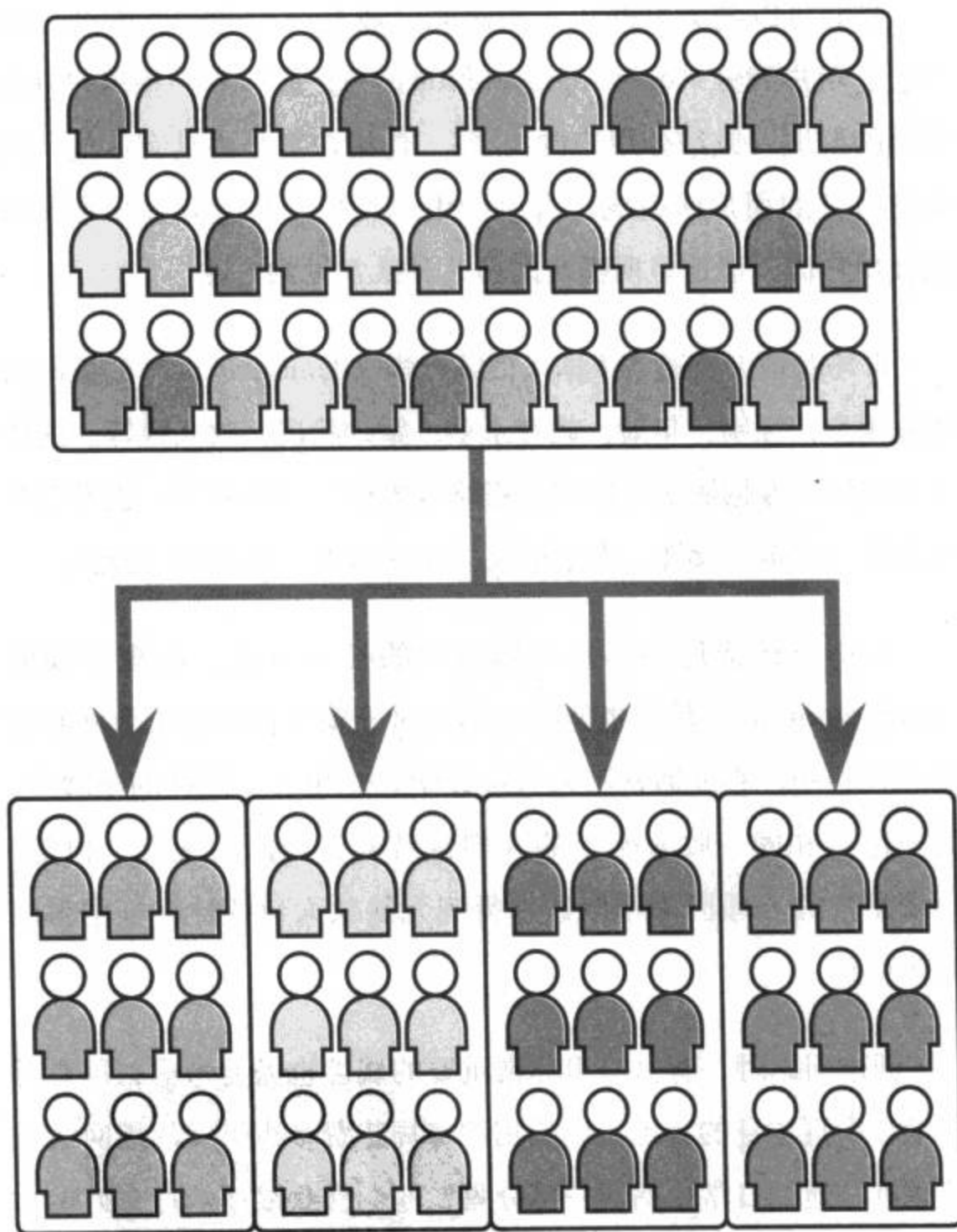
市场营销人员通常依据人口统计学（demographic）的标准来划分用户：性别、年龄、教育水平、婚姻状况、收入等等。这些人口统计的数据概况可以相当粗略（男性：18至49岁）也可以非常具体（未婚、女性、大学毕业、25~34岁、年薪5万美元）。

人口统计特征并不是了解用户的唯一方法。心理方面的（psychographic）数据也能描述用户对于整个世界或某个网站的观点和感知。心理通常与人口统计学息息相关：同样的年龄段、同一地点和同一收入水平的人们常常会有相似的观点。但是，记录用户的心理因素可以让你得到无法从人口统计特征中得出的新见解。

创建网站时，有另一组非常重要的属性也需要考虑：用户对技术和网页本身的观点。你的用户每周花费多少时间使用网络？计算机是他们日常生活的一部分吗？他们喜欢跟技术打交道吗？他们总有最新和最好的硬件，还是他们每5年才买一台新计算机？由于对技术有恐惧心理的用户和高级用户在使用网站的方式上非常不同，因此我们的设计必须要能容纳不同类型的用户群。以上这些问题的答案可以帮助我们设计出更符合用户需求的网站。



用户细分将全部的用户划分成较小的、有共同需求的小组，以此来帮助我们更好地了解用户的需求。





除了了解用户对于技术的熟悉程度和适应程度，我们还需要知道他们对于网站相关内容的知识有多少。卖厨具给一般人与卖厨具给专业厨师的处理方式必须非常不同。同样地，为那些不熟悉股市运作的新手设计股票交易软件和为经验丰富的投资者设计也必须有不同的考虑。

人们使用信息的方式经常取决于他们的社会或专业角色。例如学生家长和那些报考大学的学生对于信息的需求就不尽相同。因而定义网站用户的不同角色可以帮助你区别并分析他们各种需求。

在对用户群开展了一些研究之后，你也许需要调整细分用户群。举例来说，如果你正在研究25~34岁、大学毕业的女性，可能会发现30~34岁年龄段女性的需求与那些25~29岁年龄段的女性不一样。如果差异足够明显，那么你或许需要将这两个年龄段作为单独分开的两个用户群。另一方面，如果18~24岁年龄段的女性与25~34岁年龄段女性的需求似乎非常相似的话，则你或许可以将其合并到一个用户群中去。创建细分用户群只是一种用于“发现用户最终需求的手段”。你真正只需要得到的是和“用户需求数目”一样多的细分用户群。



创造细分用户群还有其他重要的原因。不仅仅是因为不同的用户群有不同的需求，还因为有时候这些需求是彼此矛盾的。用先前股票交易软件的例子来讲，最适合炒股新手的软件，可能是那种能自动将股票交易过程分解成简单步骤的软件。然而对炒股专家而言，这样的步骤可能会妨碍他进行快速的操作。专家需要将所有的功能都集中在一个界面上，而且能快速地进入操作。

很明显，我们无法提供一种方案可以同时满足这两种用户的需求。此时，我们要么选择针对单一用户群设计而排除其他用户群，要么为执行相同任务的不同用户群提供不同的方式。不论我们选择哪一种，这个决策将会影响日后与用户体验相关的每一个选择。

### 可用性和用户研究

如果你曾读过任何一本关于网站设计的书，那么大概看过这个词：**可用性**（usability）。这个概念对于不同的人来讲意味着不同的事。有些人将这个词用于表述“用典型用户来验证设计方案的实践过程”。对于其他人而言，它则意味着“采取一种非常具体的开发方式”。不论哪种解释，可用性都是寻求使产品更容易使用的一种方法。

许多不同的定义和设计规则都希望编撰出网站设计的可用性规范。有些甚至彼此赞同对方不同的观点。但他们全都具有同样的核心原则：用户需要可用、易用的产品。这的确是所有用户最普遍的需求。





明白用户需要什么，我们首先必须知道他们是谁。用户研究（user research）的领域致力于收集必要的信息来达成这样的认识。

一些研究工具——比如问卷调查、用户访谈，或焦点小组——最适合用于收集用户的普遍观点与感知。

至于其他研究工具——比如用户测试或现场调查——则更适用于理解具体的用户行为以及和网站的交互方面的信息。

一般说来，你在某个用户身上花费的时间越多，就能从这个用户研究中得到越详细的信息。同时，在一个用户身上所花费的时间越多，也就意味着你不可能在用户调查中接触太多的用户（如果网站最终一定会发布的话）。

像问卷调查和焦点小组这种市场调研方法（market research method）会是得知用户普遍信息的宝贵资源。当你能明确地表达你试图从用户身上获得什么信息时，这些方法才是最有效的。想要知道你的用户如何使用某个网站特定功能吗？或者已经知道这些，但是你需要了解他们为什么使用该功能。越是清楚地描述出你想要得到的信息，就能越具体并有效地公式化你的问题，这样才能确保你获得正确的答案。



**现场调查** (contextual inquiry) 是指一整套完整的、有效且全面的方法，用于了解在日常生活情境中的用户行为（因而得来此名）。这门技术是从考古学家在研究文化和社会学时采用的方法演变过来的。它通常应用于一个较小的范围，并且执行方法都是一样的，比如，了解某个游牧部落的运作，与了解购买飞机零件的人类行为，用的都是同一种方法。现场调查的唯一缺点是它有时候会非常费时而且昂贵。如果你的资源充足，并且你的问题要求对用户有更加深刻的理解，现场调查可以揭露一些无法通过其他方法获知的、细微的用户行为。

与现场调查密切相关的研究方法是**任务分析** (task analysis)。任务分析的概念是认为每一个用户与网站的交互行为都发生在执行某一任务的环境中。有时任务非常具体（譬如买电影票），而有时则比较广泛（譬如学习国际商务章程）。任务分析是一种仔细地了解用户完成任务的精确步骤的方法。这种了解可以通过用户访谈来做，让用户通过讲述故事来说出他们的经验，也可以通过现场调查来做，在用户的“日常生活环境”中直接观察他们的行为。

**用户测试** (user testing) 是最常见的用户调研方法。用户测试并不是测试你的用户；相反地，它是请用户来帮忙测试你的产品。有时用户测试用于一个已完成的网站，也可以用于测试改版效果，或者用于在网站发布之前发现可用性的问题。另外，用户还可以测试正在建设中的网站甚至一个粗略的低保真原型。





测试一个已完成的网站可以是在一个非常广泛或是非常狭窄的范围中进行。就调查问卷与焦点小组而言，在坐下来与用户面对面之前，最好对你想要了解的问题有一个清楚的概念。然而这并不意味着，用户测试必须严格地局限在“分析用户如何成功地完成某一项特定的任务”之中。用户测试也可以调查更加广泛而并不是那么具体的问题。例如，用户测试能用来发现网站设计的调整是否能加强或减弱公司的品牌气质。

另一个用户测试的方法是让用户测试原型。原型可以是各种形式，从纸上的草图，到低保真的HTML模拟网页，以及看上去像是一个已完成的网站和“可点击”的高保真原型。大型项目在不同的阶段使用不同的原型来搜集用户意见，这将贯穿到整个开发过程中。

有时用户测试根本不牵涉到网站。可以招募用户来参加各种不同的活动，只要这些活动可以让你洞察用户如何看待并使用你的网站。卡片排序法（card sorting）用于探索用户如何分类或组织各种信息元素。给用户一迭索引卡片，每一张卡片附有信息元素的名字、描述、一张图像或内容的类型。然后用户根据小组或类别，依照自己感到最自然的方式将卡片排列出来。分析几位用户的卡片排列结果，就可以帮助我们了解用户对网站信息的看法。



收集各种各样的用户数据将是非常有价值的，但有时候你会忽略统计数字背后所代表的真正人物。因此，通过创建人物角色（personas）——有时也叫做用户模型或用户简介——你可以让你的用户变得更加真实。人物角色是能代表整个真实用户需求的虚构人物。通过赋予一张人物的面孔和名字，你将用户调查及用户细分过程中得到的分散资料重新关联起来，人物角色可以帮助你确保在整个设计过程期间把用户始终放在心里。

让我们来看一个例子。假设我们的网站是一个为刚开始创业的人提供信息的网站。从研究中得知，我们的大部分用户将是30~45岁的人。总体来说，我们的用户普遍倾向于对互联网与计算机科技有相当高的适应程度。有些用户甚至在商业领域有很多经验；而另一些，则是第一次自己经营这样的商务活动。

在这种情况下，或许应该创建两个人物角色。我们把第一个人物角色称为Janet。她42岁，已婚并有两个孩子。她过去几年在一家大型会计师事务所担任副总裁。她不满于为他人工作，现在她想要建立自己的公司。

第二个角色是Frank。他37岁，已婚，有一个孩子。从事木材加工是Frank多年的周末爱好。他的一些朋友非常喜欢他制作的家具，因此他认为他可以尝试出售他的家具作品。为了开展他的新生意，他正在犹豫是不是必须要放弃校车司机的工作。







## Janet

“我没有时间去整理大量的信息，我需要快速找到答案。”

Janet对企业的环境感到很失望，想开一家自己的会议事务所。

年龄：42  
 职业：会议事务所的副总裁  
 家庭：已婚，两个孩子  
 家庭年收入：\$140,000  
 喜欢的网站：WSJ.com

技术概况：对技术相当熟悉，Dell笔记本（大约使用了一年），Windows XP，DSL接入，每周上网8-10小时  
 互联网使用：75%的时间在家使用，新闻和资讯，购物





WSJ.com      Salon.com      Travelocity.com

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可成为样例的虚构人物。

## Frank

“这些事情对我而言很新鲜。我想要一个能解释所有细节的网站。”

Frank想知道如何才能将他制作家具的个人爱好变成一门生财之道。



年龄：37  
 职业：校车司机  
 家庭：已婚，一个孩子  
 家庭年收入：\$60,000  
 喜欢的网站：ESPN.com

技术概念：多少有一点技术恐惧，Apple iMac（大约使用了两年），Mac OS 9，拨号接入，每周上网4-6小时  
 互联网使用：100%的时间在家使用，娱乐，购物





ESPN.com      movifone.com      eBey.com



这些信息来自哪里呢？很大程度上是我们编造的。我们希望人物角色可与我们从用户研究中了解的内容保持一致，但是为了使人物角色更加栩栩如生，他们的一些具体细节可以是虚构的。

当决定网站的用户体验设计时，我们必须要紧记Janet和Frank所代表的用户群的需求。为了帮助我们记住他们和他们的需求，我们要找出一些珍藏的照片来让Janet和Frank的形象更加鲜明，然后将他们的相关信息与这些照片合并到一起。这些人物角色档案可以印出来并且张贴在办公室周围，这样当我们要做决定时候，可以问自己：“那会对Janet有用吗？Frank会有什么反应呢？”人物角色能帮助我们在前进的每一步都时时记着用户。

## 团队角色和流程

战略问题影响参与用户体验设计过程的每一个人。但是，尽管事实是这样（或许就因为如此），明确这些目标的责任人常常是受到争议的。咨询公司有时会找一个**战略专家**（strategist）帮助客户处理项目中的这些问题——但由于聘请这类稀缺的专家通常非常昂贵，同时战略专家并不直接负责建设网站的任何一部分，因此这项支出就常常成为第一个被裁减的项目预算。





战略专家会与企业内部的许多人谈话，尽可能全面地收集大家对于网站目标与用户需求问题的不同看法。决策层（stakeholder）是一群资深的决策者，他们管理那些会影响网站决策的部门。例如，该网站是为了提供给顾客进入在线产品支持信息而设计的，决策层可能就会包括市场销售部门的代表、客服部门还有产品经理。它取决于企业在做决策制定时的正式流程（以及私下的政治现状）。

在拟定决策时有一群人经常被忽略，那就是普通员工——这些人负责让企业每天正常运作。然而这些人通常比他们的经理更知道什么行得通什么不可行——特别是在用户需求方面。没有人比那些每天跟客户交谈的人更明白他们遇到的困难是什么的了。我经常很惊讶地发现，用户的反馈很少能传递到需要这些信息的产品开发团队中去。

网站目标和用户需求经常被定义在一个正式的战略文档或可视的文档（vision document）中。用户需求有时被记录在一个独立的用户调研报告中（虽然将所有信息集中在一个地方有某些好处）。这个报告不仅仅是列出目标清单——它提供不同目标之间的关系分析，并且说明这些目标要如何融入更大的企业环境中去。这些目标和他们的分析经常由决策者、普通员工，和用户自己的直接意见来支持。这些意见生动地说明了项目中的战略制定问题。



撰写战略文档时，文档并不是越多越好。你没有必要把每一个数据来源和相关的意见都写出来以表达你的观点。让它简洁明了并切中要点。请记住，许多将要阅读此文档的人不会有时间或兴趣翻阅上百页的参考资料，与其让他们惊讶于你交付的文档的重量，还不如让他们直截了当地理解这些战略来得更重要。一个有效的战略文档不仅可以在用户体验开发团队中起到试金石作用；它还可以成为企业其他部门的项目支持文档。

不让你的团队阅读战略文档是你能做的最糟糕的一件事。战略文档并不是创建出来被藏到某个地方，或只与少数几个资深员工分享的——如果这些努力想要得到回报的话，在项目进行期间，此文档就必须被频繁地使用。所有参与者——设计师、程序员、信息架构师、项目经理——需要这份战略文档，以帮助他们在工作中做出正确的决定。战略文档通常包含敏感的资料，但仅仅因为这个就不告知应该理解战略目标的团队，只会破坏他们完成这些事情的能力。

战略应该是设计用户体验流程中的起点，但那不意味在项目开始之前你的战略需要完全确定下来。虽然设法击中一个移动的目标可能会浪费很多时间和资源（更不用说极大的内心挫折感），但是战略也应该是可以演变和改进的。当战略被系统地修改与校正时，这些工作就能成为贯穿整个过程的、持续的灵感泉源。





## 推荐阅读

Cooper, Alan. The Inmates Are Running the Asylum: Why High-Tech.

Products Drive Us Crazy and How to Restore the Sanity. Sams, 1999.

Krug, Steve. Don't Make Me Think: A Common Sense Approach to Web.

Usability. New Riders, 2000.

Spool, Jared M., et al. Web Site Usability: A Designer's Guide. Morgan.

Kaufmann, 1998.

Web resources: [www.jjg.net/elements/resources/](http://www.jjg.net/elements/resources/).



# 第4章

## 范围层

功能规格和内容需求



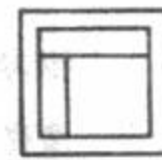


因此，在对创意进行评价时，设计师需要具备其相应的专业素养和综合能力。首先，设计师需要具备扎实的专业知识，包括设计理论、设计史、设计方法等。其次，设计师需要具备较强的创意思维能力，能够提出新颖、独特的设计创意。最后，设计师还需要具备良好的沟通能力、团队协作能力和项目管理能力，能够与客户、团队成员进行有效的沟通和协作，确保设计项目的顺利推进。

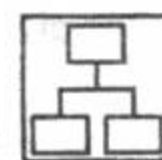
带着“我们想要什么”、“我们的用户想要什么”的明确认识，我们就能弄清楚如何去满足所有这些战略的目标。当你把用户需求和网站目标转变成网站应该提供给用户什么样的内容和功能时，战略就变成了范围。



表现层



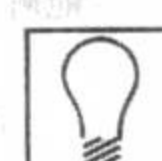
框架层



结构层



范围层



战略层



## 范围层定义

我们做的某些事是因为其过程具有价值，就像慢跑或练习钢琴一样。而我们做的另一些事情是因为其产品具有价值，就像做一块蛋糕或修理一辆汽车。定义项目范围就包括了这两方面：一个有价值的过程导致一个有价值的产品。

过程（process）的价值在于，当整个事情还处在假设阶段的时候，它能迫使你考虑潜在的冲突和产品中一些粗略的点。我们可以确定现在能解决哪些事情，而哪些必须要再迟一点才能解决。

产品（product）的价值在于，它给了整个团队一个参考点，关于在这个项目中要完成的全部工作，它也提供了一门共同的语言，用于讨论这方面的事情。定义好你的要求能保证在开发过程中不会出现模棱两可的情况。

我曾经设计过一个似乎永远是Beta版的网页应用系统：差不多，但还是没准备好向真正的用户推出。用我们的方法去做的许多事情都是不对的——技术不稳定、我们好像根本不了解用户，同时我是整个企业中唯一一个在开发网页方面有一些经验的人。

但是这些都不能解释为什么我们不能推出这个产品。最大的绊脚石是没有一个人愿意去撰写一个文档。毕竟，当我们所有的人都在同一个办公室里工作时，记下每一件事是很麻烦的。除此以外，产品经理也需要把他的精力集中在新功能的思考上。





这样做的结果，就产生了一个不断变化的、混合了完整产品的各个阶段功能的产品。只要有人读到的一篇新文章或一个新的想法，都会启发这个产品的负责人的灵感，然后考虑增加另一个功能特性。我们倒是有一个固定的工作流程，但是没有日程安排，没有里程碑，项目也看不到尽头。因为根本就没有人知道这个项目的范围，又怎么能知道它应该在什么时候结束呢？

用文档来说明项目要求，这件事很麻烦，但是你必须要做。这是由于以下两个主要原因：

### 原因#1：这样你才知道正在建设什么

这似乎很明显，但是它对于正在建设某个网页应用程序的团队却是一个惊喜。如果详细地记录下你正在建设的内容，每一个人就会知道这个项目的目标是什么，什么时候达到这个目标。最终产品不再是一个停留在产品经理头脑里的不定形的图像，它变成了一种在企业内部的每一个级别的每一个人都触手可及的东西，从高层管理人员到入门级的工程师，人人都能参与进来。

如果没有文档的要求，你的项目很可能会变成一个叫做“电话”的校园游戏——在团队中的每一个人都得到一个关于产品的印象，然后通过口口相传的方式传递出去，每个人的描述都略有不同。甚至更糟的可能是，每个人都认为别人肩负着管理产品关键环节的责任，但事实上这个人并不存在。

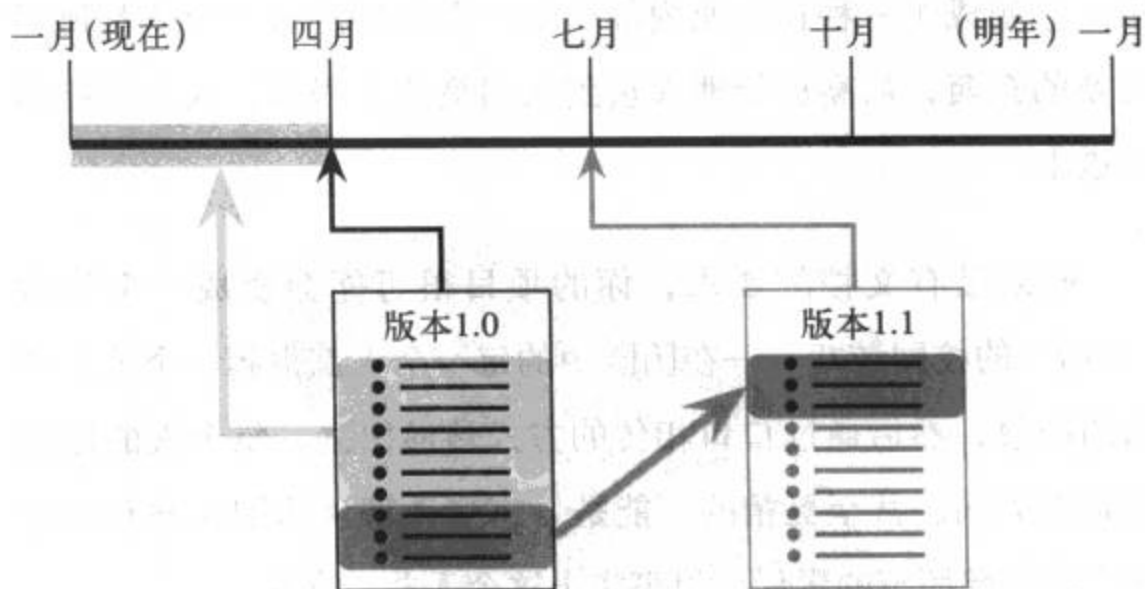


拥有一系列明确的要求，能让你把责任分配得更清晰，这可以大大提高协作的效率。在了解了一个详细策划的完整范围之后，你就可以看到各个相对独立、也不显著的要求之间的内在联系。在早期的讨论中，支持文档和产品规格表可能看起来像是独立的功能内容，但是把它们作为要求确定下来，会让这种有所交叠的事情更加明显，并且同一个小组的人可以把它们都负担起来。

#### 原因#2：这样你才知道不需要建设什么

许多功能听上去都相当诱人，但是它们对于项目的战略目标并不是必需的。此外，所有在项目开始如火如荼地进行时，关于功能的、各种各样的可能性都会浮现出来。当这些想法出现的时候，用一个文档来记录它们，可以为你提供一个评估这些想法的架构。

当前难以满足的需求，  
可以成为启动下一个版本的基础，这样就能形成一个不断循环的开发过程。





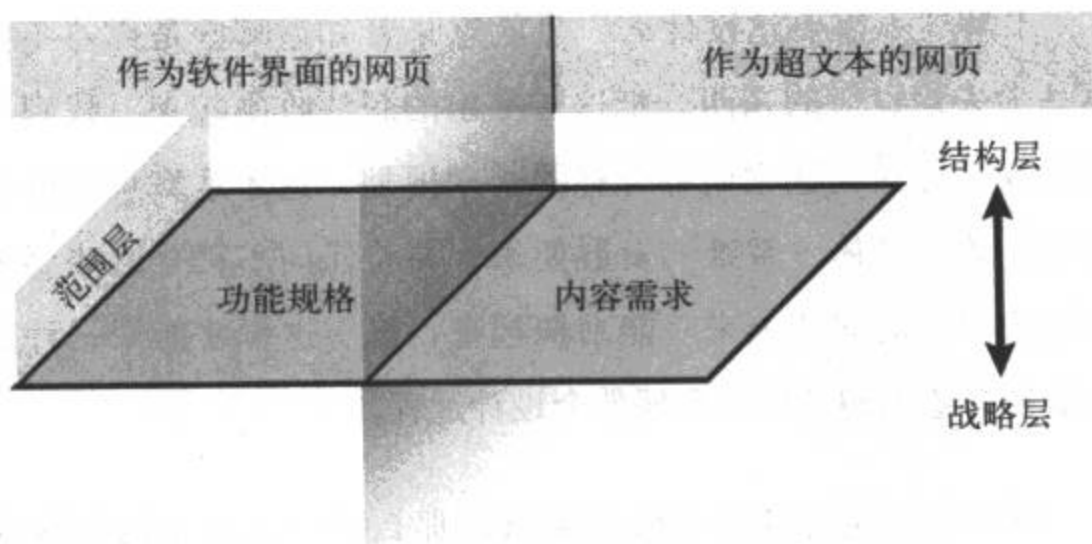
了解“不需要建设什么”也就意味着知道哪些是你“不需要马上去建设”的东西。把这些杰出的想法收集起来，找到一种适宜的方式，让它们符合你的长期规划，这才是真正的价值所在。确定具体、系统的发展要求，并将任何不符合这些要求的想法作为潜在的未来功能囤积起来，只有这样才能用一种更慎重和有意识的途径来管理整个设计过程。

如果你不能有意地管理要求，那么将陷入可怕的范围蠕变（scope creep）中，这个词总会让我想起一个景象：一个雪球向前滚了一英寸——接着又是一英寸——每一次滚动都会捎带上更多的雪，直到它冲到山脚下。在这个过程中，雪球变得越来越大，你也越来越难以阻止它。与这个景象相同的，每一个额外的要求看上去并没有增加太多的工作量，但是当它们汇集到一起的时候，整个项目就会失去控制地膨胀，结束时间遥遥无期，而费用预算正不可避免地朝着最终的分崩离析飞奔而去。

## 功能和内容

在范围层，我们从讨论战略层面的抽象问题——“我们为什么要建这个网站？”——转而面对一个新的问题：“我们要建设什么？”





在这里，范围层被软件界面的网页和超文本的网页分成两个部分。在软件方面，我们考虑的是功能规格——哪些应该被当成软件产品的“功能组合”。在超文本方面，我们考虑的是内容，这属于编辑和营销推广的传统领域。

内容和功能看上去很像是两个完全不同的事物，但是当它们在定义范围层的时候，它们所用的方式是非常相似的。在本章中，我将使用一个词特性（feature）来同时表示软件的功能和所提供的內容。

在软件开发中，范围层确定的是功能需求或功能规格（functional specification）文档。有些企业用这些术语来表示两种不同的文档：项目初期阶段的需求，描述系统应该做什么，以及在末期的规格说明，描述系统真正完成了什么。在这种定义中，功能规格在功能需求确定之后才开始撰写，同时将加入更加具体的实施细节。但是大部分的时候，这两个术语是可以互换的——事实上，有些人使用“功能需求规格”来表示他们的文档覆盖了包括以上两者的内容。我将使用功能规格来描述





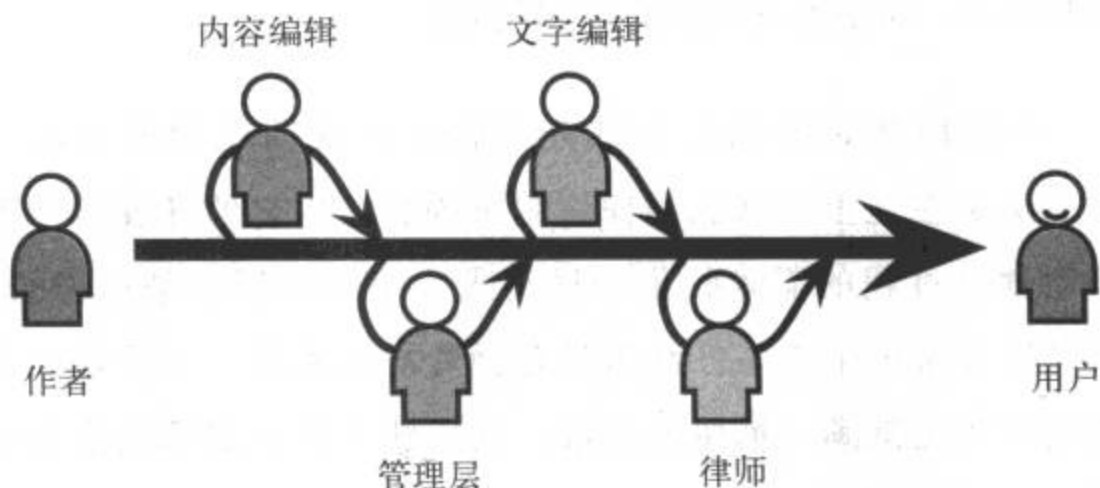
文档本身，而用需求来描述文档的内容。

本章所使用的词汇大部分和软件开发所使用的词汇一样。但是在这里的概念同样适用于内容。内容的开发常常不会像软件过程的需求收集一样正式，但基本原则是一样的。内容设计者要坐下来仔细考量各种资料的来源，不管是一个数据库还是满满一抽屉的剪报，然后她才能决定哪些信息必须纳入设计范围之内。这种不太正式的、收集内容需求（content requirement）的过程，实际上与技术专家和董事会集体商议功能需求，并回顾已有的文档记录没有本质上的区别。两者的意图和方法是一样的。

内容需求常常伴随着功能的需求。现在，内容常常是通过一个内容管理系统（content management system, CMS）来进行管理的。这些系统大小不一，大的系统能根据众多不同的数据来源动态生成页面，庞大而复杂；小的可以是一个很轻巧的工具，能以最高效的方式来优化某种类型的专题内容的管理。你也许决定去购买一套专用的管理系统，或从众多开放源代码的候选方案中选择一个，甚至从头开始建设一个管理系统。不管是哪一种方式，在大部分情况下，你都必须对这个系统进行一些简单的修补以适合你的企业和内容的需要。



一个内容管理系统可以实现自动化流程，能出示和交付内容给用户。



内容管理系统必备的功能取决于你将要管理的内容的性质。你是否需要维护多语言或具有数据格式的内容？CMS就需要具有处理这些类型的内容元素的能力。你的每一篇新闻稿是否必须要通过六个执行副总裁和一个律师的审核？CMS就需要在流程中支持这些类型的需求。你的内容元素是否要根据每一个用户的喜好动态地组合？那么CMS就必须要能完成这些级别的复杂输出。

类似地，功能需求也常常伴随着内容的需求。在个人喜好设置的页面中需要有使用说明吗？错误提示呢？必须要有个专门的人来写这些内容。每一次当看到网页上出类似“无效输入”的错误提示时，我就知道这种文字是出自开发工程师之手，并成为了最终产品，因为没有人把这些错误提示纳入内容需求中。而事实上，如果开发者能花一点点时间让某些人看一看应用程序中的内容的话，无数的技术项目会因此得到极大的改善。





## 收集需求

一些需求适用于整个网站。品牌需求是最常见的一种；某些技术需求，比如支持浏览器和操作系统，是另一种。

另一些需求只适用于特殊的特性。大多数的时候，当人们说到某种需求的时候，他们想的是产品必须拥有的、某种特性的一句简短描述。

最用之不竭的需求源泉总是来自用户本身。去了解“人们在想什么”的最佳途径就是直接询问他们。在第3章列出的用户研究技术都可以用来帮助你更好地了解用户，了解他们希望在你的网站上看到的特性的种类。

不管你是从企业内部的管理者，还是直接从用户处收集这些需求，这个过程中得到的需求将被分成三个主要类别。首先，最显而易见的是人们讲述的、他们想要的东西。这中间有一部分是非常清晰的好想法，会寻找各种途径进入最终产品。

有时候人们口中说出来的、所期望的特性并不是一个好主意，但是它们代表了一条通向下一个版本的路径：用户实际想要的东西。当人们在某个过程或某个产品中遭遇到一些困难时，想像有某种解决办法可以缓解这一困难，这对任何人来讲都是很正常的反应。有时这个解决办法是行不通的，或者仅仅是治标不治本的办法。通过与用户探讨这些建议，你有时候可以得出能真正解决问题的、完全不同的需求。



在这个阶段能得到的第三种类型的需求是人们不知道他们是否需要的特性。当你让人们讨论新的需求和战略目标时，他们有时会突然想起某个伟大的构思，而根本忘记了那个正在维护中的网站。这些通常会在头脑风暴讨论的时候出现，那正是与会者有机会参与和探讨项目的可能性的时候。

具有讽刺意味的是，那些很少去想像网站的新方向的人，恰恰是参与创建和设计网站最深入的人。出于这个原因，汇集企业各个部门的成员或不同类型的用户代表来进行的头脑风暴会议，可以是一种打开设计者思路、让他们考虑以前从未想到的可能性的、非常有效的工具。

让一个工程师、一个客服人员、一个营销人员坐到一间会议室中谈论同一个网站，这会对大家都有启发意义。听取各方从自己熟悉的角度出发来考虑的、对于网站的观点——并给予反馈——可以鼓励人们从不同的角度来思考开发中的网站遇到的问题以及解决办法。

通过这种方式得到的需求通过是得到如何去除某些障碍的方法。假设你有一个用户已经决定要购买了——他们只是没有决定是不是买你的产品，你的网站要怎样才能让这个过程——首先是选择你的产品，然后买下它——对他们来讲更容易呢？





在第3章，我们看到一种叫作人物角色的，创建虚拟人物，来帮助我们更好地理解用户需求的技术。在决定需求的时候，我们可以再次使用这些人物角色，把我们的虚拟人物放到一个简短的故事之中，我们称之为场景（scenarios）。一个场景是一个简短的故事，简单描述了一个人物角色会如何完成这些用户需求。通过“想像我们的用户将会经历的过程”，我们就可以得到有可能满足他们的需要的潜在需求。

我们也期望从竞争对手处得到一些启示。任何一个在做同一件事的企业基本上在试图满足同样的用户需求，同时也在试图完成相似的网站目标。竞争对手是否找到一种特别有效的特性，能达到其中的某个战略目标？他们是如何权衡和调整我们所面对的那些问题的？

即使不是网站的直接竞争对手也能提供丰富的潜在需求。比如大部分的企业网站，提供了关于工作机会的信息。了解我们所处行业之外的一些企业已经采用的内容类型，也许就能发现一条可以给我们一定的优势，能超越直接竞争对手的途径。

你的需求的详略程度常常取决于该项目的具体范围。如果项目的目标是完成一个非常复杂的系统，那么可能需要有一个非常详细的需求，即使这个项目的范围相对于某个更大的网站来讲非常小。相反地，一个大型项目，它的内容也许只是相似或相同性质的东西，那么内容需求只需要一般化就可以了。



## 功能规格

功能规格在某些方面的名声不太好。程序员们痛恨功能规格，因为它们非常枯燥，并且会占用大量编码的时间去阅读它们。结果，那些“没有人读的功能规格”反过来又强化了“撰写它们是一件浪费时间的工作”的印象。

关于功能规格的抱怨之一是它们没有反映实际的产品。“在实施过程中事情会产生变化”，每个人都理解这个——这是技术型工作的正常情况。有时候你考虑好的一些事情会行不通，或不大可能以你想像的那种方式来运作。无论如何，这并不是一个把撰写功能规格当成一件失败的工作而放弃的理由。相反，它强调了“维护功能规格”和“及时更新它”的重要性。当事情在实施过程中发生改变的时候，你不应该抬起双手宣称撰写功能规格是没有价值的，而是应该随时注意保持功能规格与开发的同步更新。

但是，无论这个项目有多么庞大或复杂，有几条规则适用于撰写任何类型的需求。

**乐观 (be positive)**。描述这个系统将要做什么事情去“防止”不好的情况发生，而不是描述这个系统“不应该”做什么不好的事情。比如，下面这句描述就不太好：

这个系统不允许用户购买没有风筝线的风筝。





替换成下面这句会更好：

如果用户想买一个没有线的风筝的话，这个系统应该引导用户到风筝线页面。

**具体 (be specific)**。尽可能详细地解释清楚状况，这是我们能决定一个需求是否被实现的最佳途径。

对比下面的例子：

1. 该网站要使残疾人可用。
2. 该网站要遵守美国残疾人法案的第508条。

第一句话看上去像是定义了一条明确的需求，但是不需要花费太大精力就能看出它有很多漏洞。什么叫做“可用”？如果这个网站提供所有图片的文字描述是否就“可用”了呢？哪些人可以算成残疾人？如果这个网站没有音频，它也必须使聋人“可用”——这是否就可以了呢？

幸运地是，美国国会已经为我们写出了所有的这些定义和区别，那就是第二句话所说的、非常详细地定义了我们的目标的某个法律文件。通过去除有不同释义的可能性，第二条需求完全避开了在实施过程之中或之后有可能出现的争议。

**避免主观的语气 (avoid subjective language)**。这是另外一种使需求“保持明确”和“避免歧义”的途径——因而也避免了误解的可能性。



这里有一个非常主观的需求：

这个网站的风格应该是时尚、闪耀的。

需求必须可验证——就是说，它必须要能证明“这个需求没有被满足”。你如何去验证这种被宣称为“时尚”和“闪耀”的产品品质？我对于时尚的定义也许并不符合你的，而CEO更可能对此有完全不同的看法。

这并不是说你不能要求你的网站时尚，只是必须找到某种方式来明确说出应该达到的标准：

这个网站应该符合邮递员Wayne所期望的时尚。

Wayne通常不会对这个项目说些什么，但是我们的项目发起人很显然会尊重他对于时尚的看法。而且这有可能和我们的用户的期望值是一样的。但是这样的需求仍然是很主观的，因为我们依赖的是Wayne对于风格的想法，而不是可以更客观界定的一系列的标准。所以，这个需求最好能像这样：

网站的外观应该符合企业的品牌指南文档。

时尚的概念已经完全从这个需求中消失了。相反地，我们得到了一个清晰的、毫不含糊的、已有的参考指南。为了确保品牌指南有足够的时尚性，负责市场的副总裁可能需要去了解邮递员Wayne，或者去和她十几岁的女儿交流，甚至她还可能做一些用户调研的工作。这取决于她。但是现在我们可以明确地说出“这个需求是否被满足了”。





我们也可以用量化的术语来定义一些需求，通过这样的手段来避免主观性。正如成功的标准能使战略目标得以量化一样，用量化的标准来定义需求也有助于我们清楚地知道我们是否满足了需求。比如，要求系统具有“高级别的执行能力”，我们可以用“要求这个系统的设计至少要支持1000个用户同时使用”来代替。如果最终产品只能支持千位数的用户，我们就知道这条需求没有被满足。

## 内容需求

很多时候我们说到的内容指的是文本。但是我们还应该记住图像、音频和视频也算是内容的类型。这些不同类型的内容可以一起协作去满足某一个需求。比如说，一个涵盖了运动赛事的专题内容，它可能是一篇附有照片的文章以及视频片段。定义出所有不同类型的内容，可以帮助你确定需要哪些资源来制作这些内容（或它们是否应该被提供）。



不要混淆某段内容的格式和它的目的。当我和管理层讨论内容需求的时候，所听到的第一件事往往是：“网站应该有FAQs（Frequently Asked Questions：常见问题）。”但是FAQ这个词仅仅指的是内容的格式：一个系列简短的问题和回答。一个FAQ给予用户的真正价值，在于它可以随时提供用户普遍需要的信息。其他的内容需求也可以满足同样的目的；但是当关注点是格式时，目的本身就可能被遗忘。多半的结果是，FAQs忽略了这个词汇中“常见”两个字，内容设计者总是用其他一些答案替代了能真正满足FAQ需求的答案。

内容特性的每一个预期的规模，都将对你必须做的用户体验决策产生极大的影响。内容需求应该提供每一个特性规模的大致预估：文本的字数、图片的像素大小、下载的文件字节、类似PDF的独立内容元素，或类似音频或视频一样的特性。这些大小的估计不一定要非常精确——大致相近即可。我们所需要的，只是收集我们在设计一个适宜的网站内容时必要的资料。设计一个提供小的缩略图图片的网站，与一个提供原始大小图片的网站是完全不一样的；事先知道我们必须考虑的内容元素的大小，能让我们在这个设计过程中做出最明智的决定。





尽可能早地确定某个人来负责每一个内容元素是非常重要的。一旦它被认为对我们的战略目标是有用的，任意一种内容特征听起来不可避免地会被当成一个好主意——只要是别人来负责建设和维护它。如果我们在没有确定谁将会负责每一个内容特性需求的情况下，过多地投入到开发流程中去，那么我们最后得到的很可能就是一个千疮百孔的网站，因为那些在假想阶段人人都喜欢的特性，将在实际执行的时候变得非常沉重。

而且这正是人们在确定需求的时候常常忘记的事情：内容是一件艰苦的工作。也许你可以找一些合作的资源来为网站的初次上线及时地准备好一些内容（或者，更有可能指定某个市场人员来完成这个工作），但是谁来更新它们呢？内容——嗯，应该说，有效的内容——需要日常维护工作。如果在做内容规划的时候，你认为它们只需要发布一次，之后再也不用更新的话，那么随着时间的推移，这个网站就越来越难以满足用户的需求。

这就是为什么你应该定义每一个内容特性的“更新频率”的原因。更新频率应该来自于你的网站的战略目标：从你的网站目标方面来看，你希望用户多长时间来访问一次？从你的用户需求方面来看，他们希望多长时间更新一次信息？无论如何，要记住，对于你的用户而言较为理想的更新频率（“我马上了解每一件事，24小时服务！”）也许对你的企业来说不切合实际。但你必须要确定一个频率，它是介于你的用户期望值和有效资源之间的一个合理的中间值。



如果你的网站是为各种用户服务的，那么确定哪些用户会更想要哪种内容特征是非常有益的。尤其是这些用户各有不同需求的时候，知道哪些用户想要什么样的内容，有助于我们在思考“如何呈现它们”的时候做出更好的决定。“为孩子们准备的信息”与“为他们的父母准备的信息”是两种完全不同的方式；而“为所有人准备的信息”则应该是第三种处理方式。

对于那些已有大量内容的项目而言（而不是那些从零开始建设的内容），很多关于内容的这些信息都记录在一个内容清单（content inventory）中。整理一个在现有网站中所有内容的清单看上去像是一件枯燥无味的过程——确实也是这样。但是得到这个清单（它通常采用一种简单的格式，即使是一个非常庞大的电子表格）是很重要的，原因和得到具体需求一样重要：这样团队中的每个人才能确切地知道他们设计用户体验需要做哪些工作。





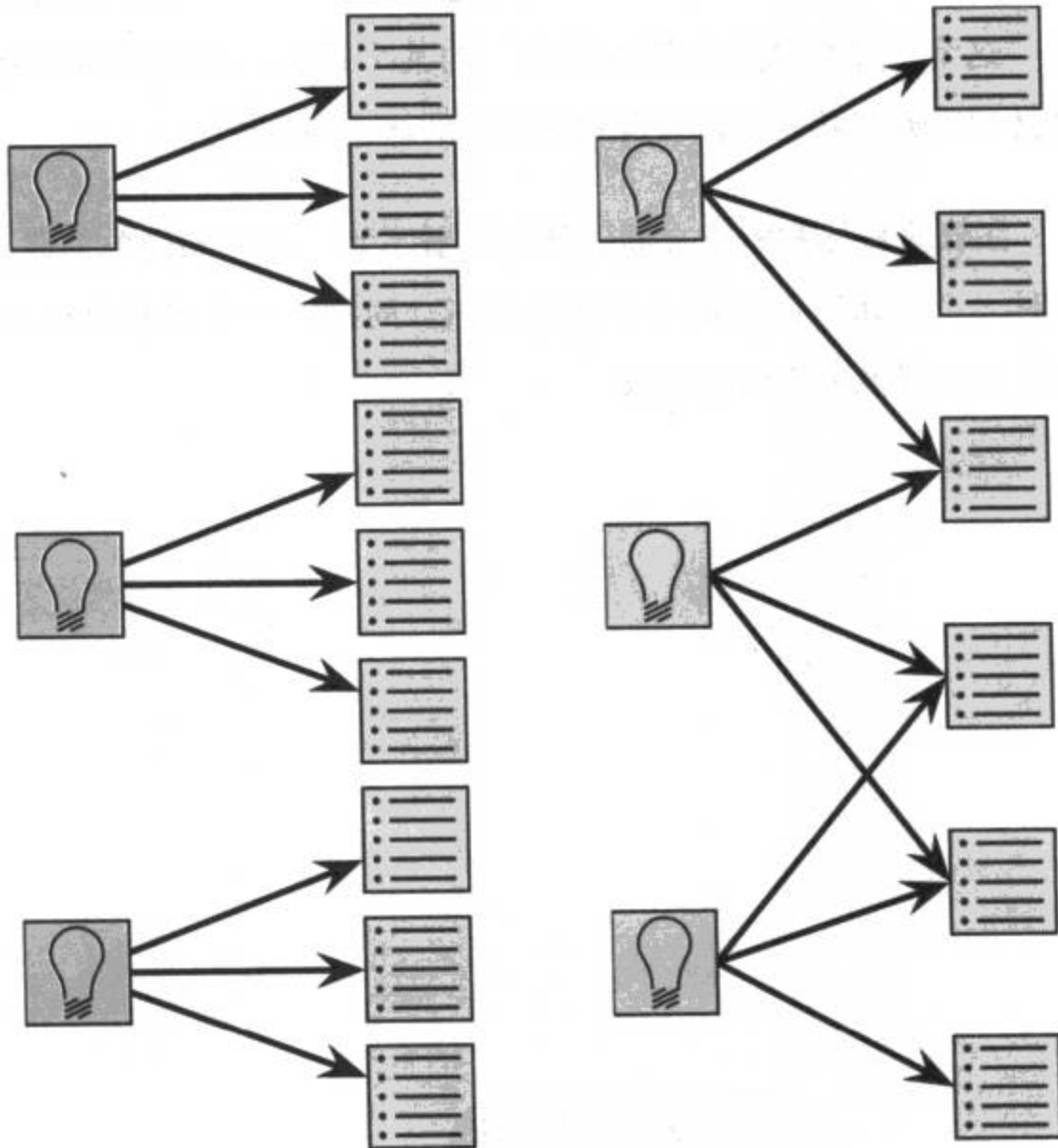
## 确定需求优先级

收集潜在需求的想法不是很困难。几乎每一个经常接触产品的人——不管他们是企业内部还是外部的人——都至少能说出“这个产品应该增加哪些特性”的某个想法。最棘手的部分是排列出哪些功能应该包含到你现在的这个项目中去。

在战略目标和需求之间，你几乎看不到一对一的简单关联。有时一个需求可以满足多个战略目标。同样，一个战略目标也常常关系到多个不同的需求。



有时一个战略目标将产生多个需求（左）。另一方面，一个需求也可以实现多个战略目标。





因为这些范围在建立是战略层的基础上的，因此我们就需要去评估这些需求是否能满足我们的战略目标（无论是网站目标还是用户需求）。除了这两种情况，我们还要额外确定第三种范围：实现这些需求的可行性有多大？

有些特性可能会因为技术上的局限而无法实现——举个例子来说，现在还完全没有办法让用户能闻到网页上产品，无论他们是如何渴望这个功能。另外一些特性（尤其是内容方面的）之所以不可行，则是因为它们需要很多资源去做——无论是人力还是财力——超出了我们的能力范围。而有时候，仅仅是因为时间不够：这个特性需要花费三个月来完成，但是企业的管理层要求在两个月以后上线。

如果是因为时间有限，那你可以把这个特性放到下一个版本或项目里程中。如果是资源有限，则技术或企业的改变有时——但重要的是，并不总是——能减少资源的负担，使某个特性得以实施（无论如何，不可能的事情仍然不可能实现，这很遗憾）。

很少有功能是独立存在的。甚至网站上的内容特性也要依赖其他特性的支持，并告诉用户怎样最好地利用所提供的内容。这不可避免地导致了特性之间的冲突。有些特性要和其他的一起权衡，才能得到一个连贯的、统一的产品。举例来说，一些用户也许想要一步提交订单的过程——但是网站所使用的、混乱的老数据库无法适应这个需要。采用多个步骤的流程是更好的办法，也许你可以重新设计数据库系统？这完全取决于你的战略目标是什么。



留意那些看上去有可能需要改变战略的特性建议，它们在开发可视文档期间并不明显。任何不符合当前项目的战略目标特性建议，都要通过定义将其排除到范围之外。但是如果有一个特性的建议，尽管它不在项目范围之内，超过任何类型的限制条件，并且听起来像一个不错的想法，但你可能需要重要审视某些战略目标了。不管怎么样，如果你发现自己正在重新查看战略目标的话，那么你可能是太早地进入需求收集阶段了。

如果你的战略计划或可视文档在战略目标的范围内确定了一个清晰的优先级别顺序，那么这些优先级别应该是决定人们所建议的相关特性的首要因素。有时候，两个不同战略目标之间的重要程度也会出现不是很清楚的情况。这时候，特性最后是否能纳入项目范围之中，往往取决于企业的政治局面。

当管理层谈到战略的时候，他们通常从某种产品特性开始，然后你不得不耐心地把他们引导到后面的战略因素上去。由于管理层常常分不清特性和战略，某些特性就会在这个讨论需求的过程中占据了上风。因此需求的收集过程就变成了与这些管理层进行谈判的过程。





控制谈判的过程会非常困难。解决管理层之间的争论的最好办法是要求“确定战略”。关注战略目标，而不是各种实现这些目标的手段。如果你能向一个总是把注意力放在某个战略目标特征上的高层决策者保证，她所关注的这个特征可以用另一个方式来满足的话，她就不会感觉自己的意见被忽略了。诚然，说起来容易做起来难。对决策者的需求表示认同，是解决特性冲突的关键。谁说技术人员不需要沟通技巧呢？

## 推荐阅读

Wiegers, Karl E. *Software Requirements*. Microsoft Press, 1999.

Robertson, Suzanne and James Roberston. *Mastering the Requirements Process*. Addison Wesley, 1999.

Web resources: [www.jjg.net/elements/resources/](http://www.jjg.net/elements/resources/).





# 第5章

## 结 构 层

交互设计与信息架构

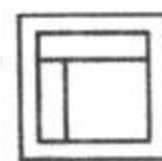




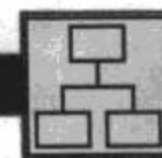
在收集完用户需求并将其排列好优先级别之后，我们对于最终产品将会包括什么特性已经有了清楚的图像。然而，这些需求并没有说明如何将这分散的片段组成一个整体。这就是范围层的上面一层：为网站创建一个概念结构。



表现层



框架层



结构层



范围层



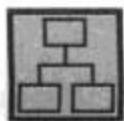
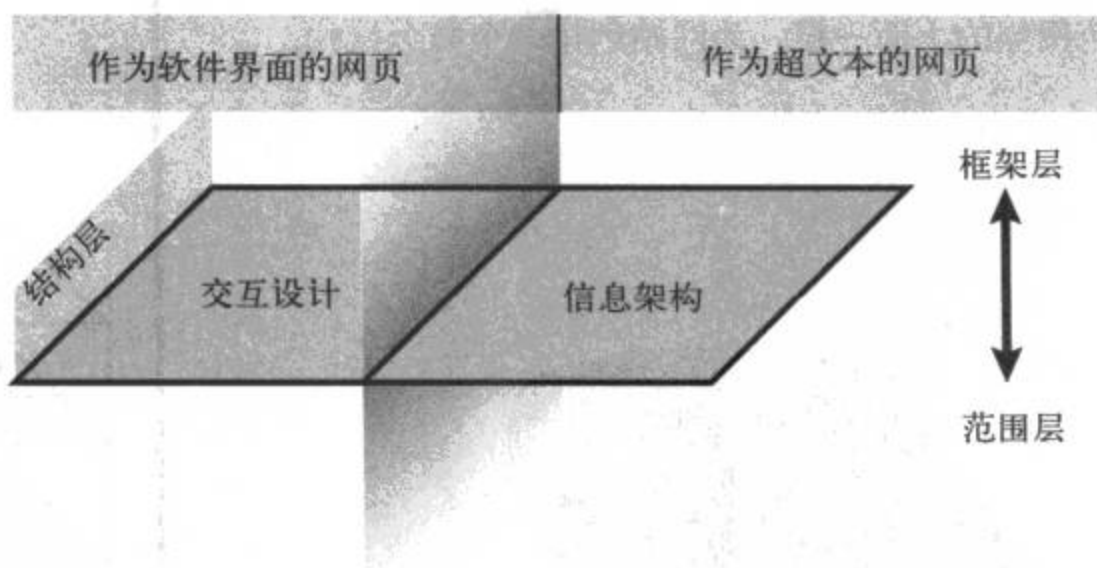
战略层



## 结构层定义

结构层是五个层面中的第三层，它也适当地将我们的关注点从抽象的决策与范围问题，转移到更能影响最后的用户体验的具体因素。然而，在抽象和具体之间的分隔线有时会变得模糊不清——虽然我们在此时做出的决定对最终产品将会产生显而易见、真实可触的影响，但是这里所做的决策本身仍然包括大部分的概念性内容。

在传统的软件开发行业，涉及到“为用户设计结构化体验”的方法被称为交互设计（interaction design）。它曾经被归类在“界面设计”的范畴之内，但近些年来（主要是由于互联网应用程序的出现和从业者的积极倡导），交互设计已经成为了一个独立的学科。





在内容建设方面，主要是通过**信息架构**（information architecture）来构建用户体验。这个领域涉及到多个学科，包括向来都要考虑的组织管理、分类、顺序排列，以及与内容呈现有关的：图书管理、新闻学和技术通信等其他学科。

交互设计和信息架构都强调一个重点：确定各个将要呈现给用户的选项的模式和顺序。交互设计关注于将影响用户执行和完成任务的选项。信息架构则关注如何将信息表达给用户的选项。

交互设计和信息架构听起来很神秘、很高科技，但这些工作实际上并不完全是技术的，它们关心的是理解用户、用户的工作方式和思考方式。将了解到的这些知识加入到我们的产品结构中，通过这个方法来帮助确保那些必须使用这些产品的用户的体验。

## 交互设计

交互设计关注于描述“可能的用户行为”，同时定义“系统如何配合与响应”这些用户行为。人类在使用计算机的时候，用户和机器之间都会产生某种类似舞蹈的步伐。用户移动，系统响应；作为对系统的回应，用户又接着移动，这样舞蹈才能继续进行。但一般的软件设计并没有刻意地留意这样的舞动。这种软件的设计思路是，反正每一种应用程序的舞步都会有一些不同的地方，让用户来适应这些不同的舞步并不算过份的要



求。所以系统就可以自己跳自己的，要是某些用户的脚被踩了，那也只能当成是学习过程的一部分。然而，每一位舞者都会告诉你，成功的舞蹈需要每一个参与者能够预测对方的移动。

传统上，程序员最关注软件的两个方面：“它做什么”和“它怎么做”。程序员之所以会这样是有原因的——正是由于他们对细节的热情，使得程序员能做好他们的本职工作。也正是由于这样的关注，意味着程序员可能会擅自建立一个在技术上效率很高的系统，却忽略了什么才是对用户而言最好的系统。尤其是在过去，计算能力是一种稀缺资源，所以最佳的方法就是在种种系统局限下让软件正常运作。

对计算机而言，最好的工作方式从来都和用户所期望的最好的工作方式背道而驰。因而，计算机软件自存在以来，一直为这样的恶名所困扰：软件是复杂的、混乱的、难以使用的。就在十年以前，“计算机基础培训”——告诉人们计算机内部的程序是如何运作的课程——曾被广泛地认为是用户和软件能和平相处的唯一方式。

经过了很长的时间，当计算机技术变得更加强大，同时我们更加了解人们如何使用计算机之后，我们终于开始有了这样的想法：与其针对机器的最佳工作方式来设计系统，还不如设计一个对用户而言最好的系统。从此以后，把文职员工送去上编程课以提高他们的计算机基础，这样的活动就被省略了。取而代之来帮助软件开发者的的是一个被称作交互设计的新兴学科。





## 概念模型

用户对于“交互组件将怎样工作”的观点称为**概念模型** (conceptual model)。例如，这是内容元素？还是用户访问的位置？还是用户请求的对象？不同的网站采用了不同的方法。规划好概念模型能帮助你做出一致的设计决定。内容元素是一个位置还是对象并不重要；重要的是网站能够将这些内容元素从头到尾一致地表现出来，而不是有时候将此元素当成位置，有时候又当成对象。

举个例子来讲，“购物车”在典型的电子商务网站的概念模型中是一个容器。这个比喻的概念同时影响了它的视觉设计和在界面上使用的语言。它是一个装东西的容器；作为一个容器，从我们“放进东西”到“推车”，以及从里面“拿出东西”来，同时系统必须提供能完成这些任务的功能。

假设购物车的概念模型是来自现实世界中的另一个实物，譬如分类订货单。系统就应该使用“编辑”来代替传统购物车的“添加”与“移除”两个功能，并且用户也应该是“寄出”他们的订单，而不是使用“结账”的比喻来完成购物。

零售商店和产品目录的概念模型似乎都可以完美地让用户在网上发出订单。要选择哪一个呢？零售商店的概念模型非常广泛地被应用在网络商城，因为它是传统的方式。如果你的用户也常常在其他网站购物，那你应该会希望继续使用这种传统方式。使用人们熟悉的概念模型，会使用户很快适应一个不熟悉的网站。当然，打破传统也没有错——只要你有一个好理由



说明“为什么这样做”，同时准备好另一个符合用户需求的概念模型以备使用。

一个概念模型可以反映系统的一个组件或是整个系统。一个主要由新闻和评论组成的网站Slate上线时，它的概念模型是一本现实世界中的杂志：网站上有封面页和封底页，并且每一个网页都有页数和允许用户“翻页”的界面。可上线后的结果是，杂志的概念模型在网络上并不能有效地使用，Slate最终还是放弃了这个概念。

我们不必将概念模型明确地告诉用户。事实上，这样做会让用户觉得很混淆，反而无法帮助他们。更重要地是，概念模型是用于在交互设计的开发过程中保持使用方式的一致性。了解用户对网站模式的想法（是零售商店的工作方式吗？还是产品目录的工作方式？）可以帮助我们挑选出最有效的概念模型。在理想情况下，我们不需要告诉用户网站使用的是什么样概念模型；用户在使用网站的时候，基本上是凭直觉的，因为这个网站的交互行为与他们的期望值完全相符。

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，这对系统功能的设计可能会有一定的价值。不过，很重要地是，不要将比喻从现实世界中一字不落地照搬过来。西南航空公司网站的首页曾经是一张客户服务的书桌图片，书桌上一边堆着简介的小册子，另一边放着电话，等等。很长时间以来，这个网站常被当成“过份运用”概念模型的一个典型例子——预订机票可以通过电话，但那并不意味着预订系统就应该真的用电

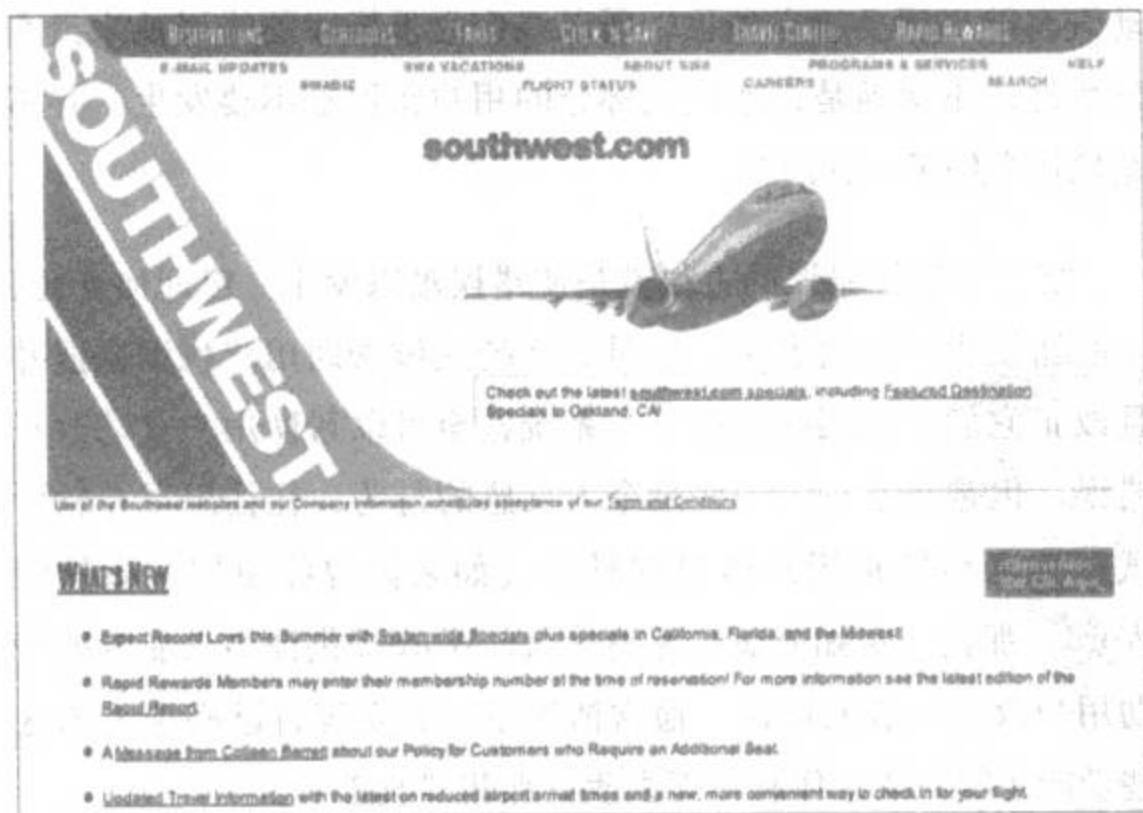




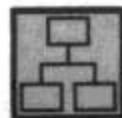
话来代表。西南航空公司一定对他们的网站常被举为坏例子感到厌烦；他们现在改版后的网站不仅避免了太强的比喻，同时也被认为更加好用。



未改版前的西南航空公司的网站是概念模型与现实物品过于相近的经典范例。



重新设计过的西南航空公司的网站让内容与功能更加明显。



## 错误处理

任何一个交互设计的项目都有很大的部份牵涉到处理“用户错误”——当人们犯错误时系统要怎么反应，并且当错误第一次发生时，系统要如何防止人们继续犯错？

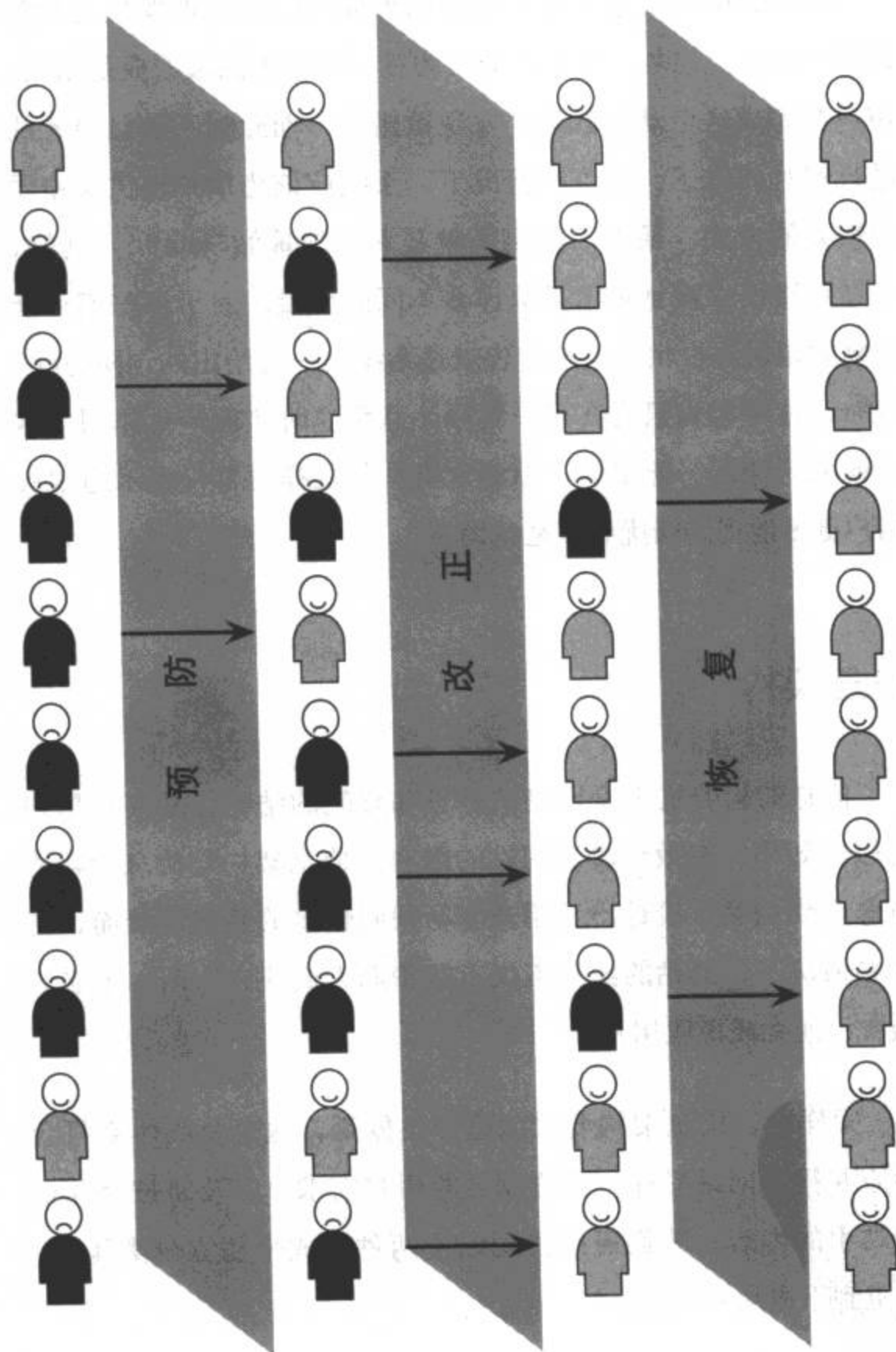
第一个同时也是最好的防止错误的方法，是将系统设计成不可能犯错的那种。这类防范方法的一个好的例子是自动档的汽车。在汽车挂在任何一个不是“P”（停车）档的档位时，启动汽车有可能会损坏变速系统的中轴和复合件；同时尽管汽车并没有真正启动，但还是有可能突然往前滑动。这有可能会伤害司机，甚至也有可能会伤害一个偶然走在滑动的汽车前边的无辜路人。

为了防止这样的情况发生，一辆自动档的汽车都会被设计成“除非挂在空档（停车档）上，否则不能启动发动机”。想发动汽车就必须清空档位，所以我们之前描述的错误从来没有发生过。不幸地是，想让大部分的用户错误都不要发生，并不是像这个例子一样简单。

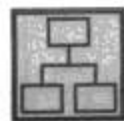
第二个避免错误的方法就是使错误难以发生。但即使如此，一些错误也一定会发生。这时，系统应该帮助用户找出错误并且改正它们。在某些情况下，系统甚至可以帮助用户自动改正错误。但是小心——一些最令人反感的行爲，往往出现在软件试图善意地修正用户错误的时候（如果你曾经使用过微软的Word，那就一定知道我在说些什么。Word提供许多功能意欲帮助用户改正一般的错误；通常情况下，我发现自己必须关闭这些功能才能完成工作和不再去改正那些“改正”）。







交互设计会处理每一个级别的错误，以确保更高比例的用户能有积极的体验。

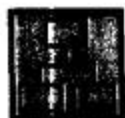


有效的错误信息和设计完善的界面可以在错误发生之后帮助用户纠正。但是一些用户的行为在一开始无法被看成是错误，用户在完成这个动作以后才发现做错了，而此时系统已经无法进行实时纠错了。在这些情况下，系统应该为用户提供从错误中恢复的方式。最有名的例子就是著名的重做（undo）功能，不过从错误中恢复可以采取许多不同的手段。对于那些不可能恢复的错误，提供大量的警告就是系统唯一可给出的预防方法。当然，这种警告只有在用户实际注意到它时才能产生作用，这也就是为什么一连串的“你确定吗？”这样的警告提示通常被用户更多地视为骚扰而不是帮助。

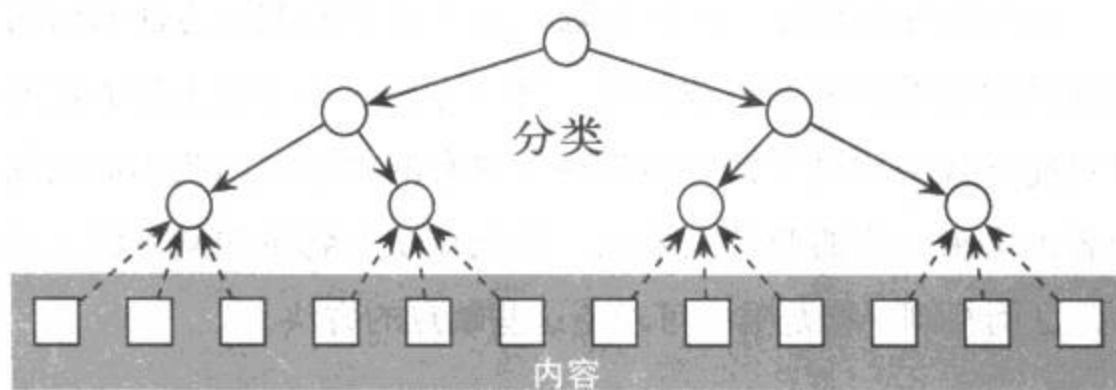
## 信息架构

信息架构着重于设计组织分类和导航的结构，从而让用户可以高效率、有效地浏览网站的内容。信息架构与信息检索的概念密切相关：设计出让用户容易找到信息的系统。然而，在许多情况下，网站的结构不仅不能帮助人们找到东西，还必须教育、通知或说服用户。

同样地，信息架构要求创建分类体系，这个分类体系将会对应并符合网站目标、希望满足的用户需求，以及将被合并在网站中的内容。我们可以使用以下两种方式来建立分类体系：从上到下或从下到上。



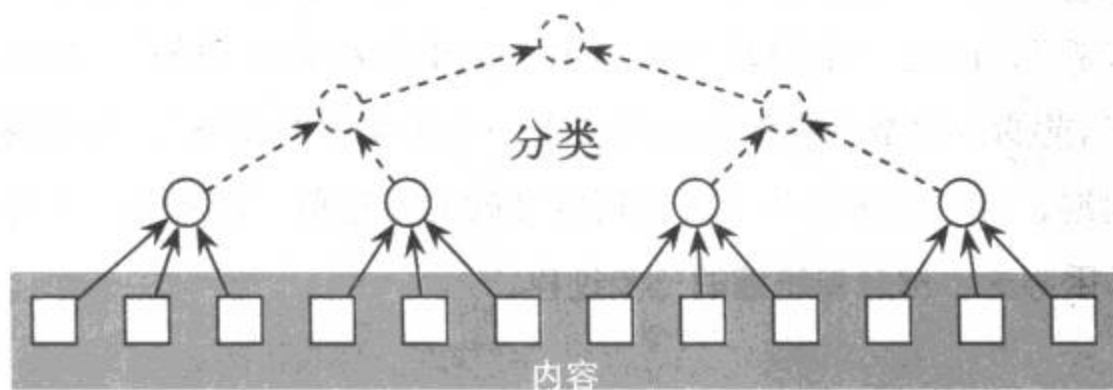




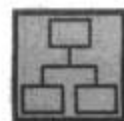
从上到下的架构方法

从上到下 (top-down approach) 的信息架构方法将从“网站目标与用户需求的理解”开始直接进行结构设计。先从最广泛的、满足决策目标的潜在内容与功能开始进行分类，然后再依据逻辑细分出次级分类。这样的主要分类与次级分类的层级结构就像一个个的空槽，而内容和功能将按顺序一一填入。

从下到上 (bottom-up approach) 的信息架构方法也包括了主要分类与次级分类，但它是根据对“内容和功能需求的分析”而来的。先从已有的资料（或者当网站发布后将存在的资料）开始，我们把这些资料统统放到最低级别的分类中，然后再将它们分别归属到高一级的类别，从而逐渐构建出能反映我们的网站目标和用户需求的结构。



从下到上的架构方法



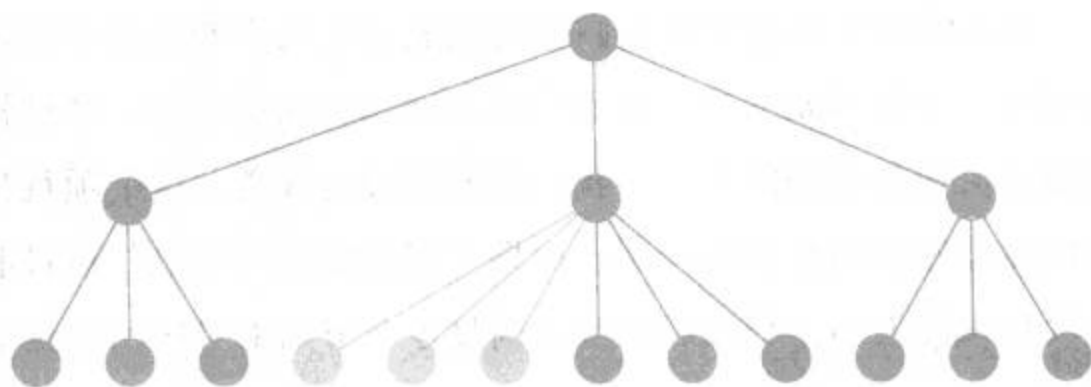
这两种方法都有一定的局限。从上到下的架构方法有时可能导致内容的重要细节被忽略。另一方面，从下到上的方法有时可能导致架构过于精确地反映了现有的内容，因此不能灵活地容纳未来内容的变动或增加。因此在从上到下和从下到上的方法之间找到平衡是唯一可避免这些缺点的方法。

网站是有生命的个体，它们需要持续的关心和灌溉。同时网站也不可避免地会随着时间的流逝而成长、改变。在许多情况下，满足新的需求不应该导致重新考虑网站的整体结构。一个有效结构的特点就是具备“容纳成长和适应变动”的能力。然而，新内容的积累最终将会需要再次审视网站的组织分类原则。举个例子来说，在你只有几个月的新闻量的时候，将新闻按日期分类，并让用户翻页查找阅读的结构或许已经足够了，但是在几年以后，按照主题来组织新闻或许更加实用。

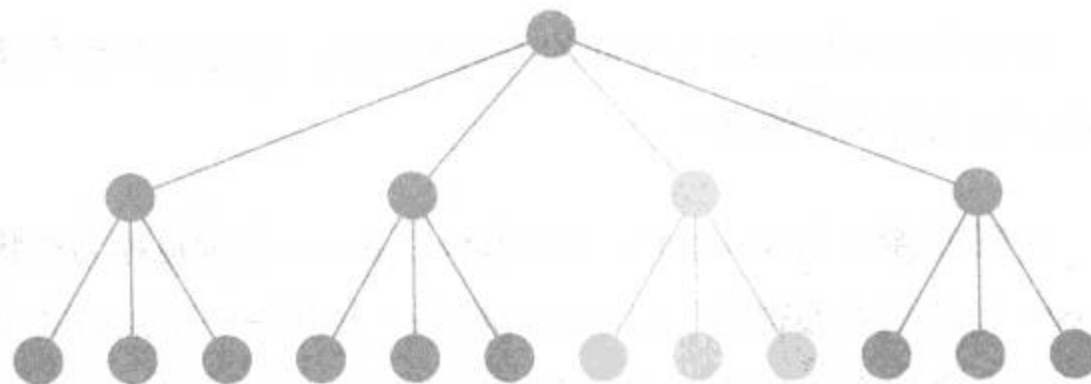
不一定非要给某个级别或某部分结构加上一个特定数目类别。类别只要能正确地反映用户与他们的需求就可以了。有些人喜欢计算“完成任务所需要的步骤”，或是计算“用户到达某一地点的点击数”，将这个作为评估网站结构质量的一种方法。然而，结构质量最重要的标准，不是“整个过程一共需要多少步骤”，而是“用户是否认为每一个步骤都是合理的”，以及“当前的步骤是否自然地延续了上一个步骤中的任务”。毫无疑问地，用户会喜欢一个被清晰定义的七步过程，而不是一个令人困惑的、被勉强压缩的三步过程。







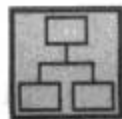
一个适应性强的信息架构系统，能把新内容作为现有结构的一部分容纳进来（上图），也可以把新内容当成一个完整的新部分加入（下图）。



一个完整的用户体验，包括网站结构，都是建立在对网站目标和用户需求的理解之上的。如果要重新定义网站希望达到的目标，或是之前设想的、网站必须满足的需求发生了变化，那么你就应该准备相应地重新调整网站结构了。但是，像这样的结构变动很少会有事先的预告，当你发现需要重新调整结构时，用户常常已经被折磨了一段时间了。

## 结构方法

信息架构的基本单位是**节点（node）**。节点可以对应任意的信息片段或组合——它可以小到是一个数字（比如产品的价格），或者大到是整个图书馆。我们要处理的是节点，而不是页面、文档或组件，这个思路有助于我们使用一种共同的语言和一组共同的结构的概念来对付各种不同的问题。

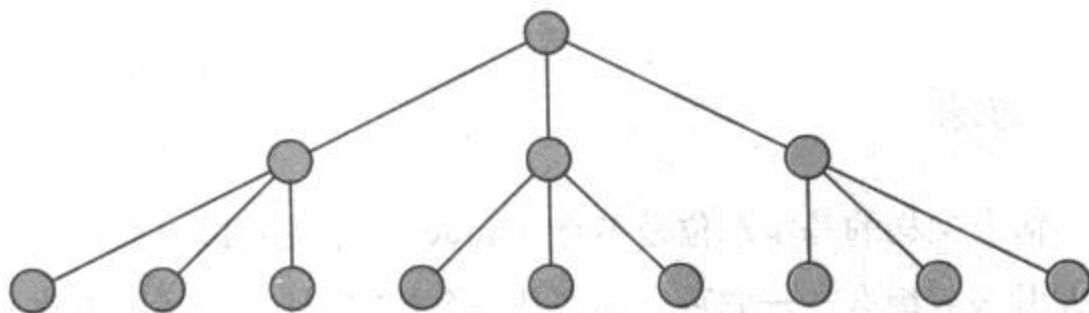


节点的抽象性也使得我们能明确地设定我们的关注点的详略程度。多数网站的架构项目只关心网站页面的安排；通过把页面定义成最基础的节点，我们就能明确地知道，这个项目将不需要处理任何更小的东西。如果“把页面作为节点”对目前的项目来说太小，那我们也可以调整各个节点来对应网站整个部分。

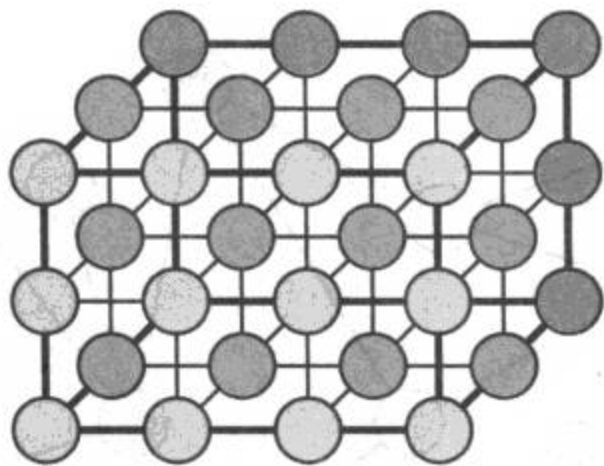
这些节点可以用许多不同的方式来安排，不过这些结构实际上只有几种常见的类型。

在层级结构（hierarchical structure）中——有时也称为树状（tree）结构或中心辐射（hub-and-spoke）结构——节点与其他相关节点之间存在父级/子级的关系。子节点代表着更狭义的概念，从属于代表着更广义类别的父节点。不是每个节点都有子节点，但是每个节点都有一个父节点，一直往上直到整个结构的父节点（或你更喜欢把它称为“树”的“根”）。层级关系的概念对于用户来说非常容易理解，同时计算机也是倾向于层级的工作方式，因此这种类型的结构是最常见的。

层级结构







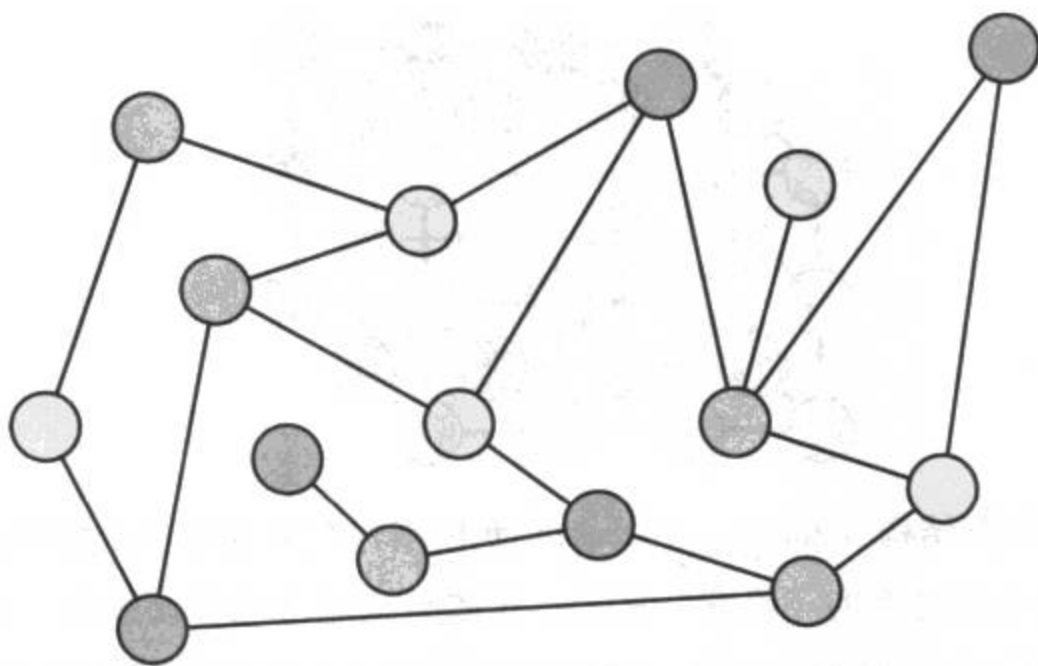
矩阵结构

**矩阵结构** (matrix structure) 允许用户在节点与节点之间沿着两个或更多的“维度”移动。矩阵结构通常能帮助那些“带着不同需求而来的用户”在相同内容中寻找他们想要的东西，因为每一个用户的需求都可以和矩阵中的一个“轴”联系在一起。举个例子来说，如果你的某些用户真的很希望能通过颜色来浏览产品，而其他人偏偏希望能通过产品的尺寸来浏览，那么矩阵就可以同时容纳这两种不同的用户。然而，如果你期望用户把这个当成主要的导航工具，那么超过三个维度的矩阵可能就会出现问题。在四个或更多维度的空间下，人脑基本上不可能很好地可视化这些移动。

**自然结构** (organic structures) 不会遵循任何一致的模式。节点是逐一被连接起来的，同时这种结构没有太强的“分类”的概念。自然结构对于探索一系列关系不明确或一直在演变的主题是很合适的。但是自然结构没有给用户提供一个清晰的指示，从而让用户能感觉他们在结构中的哪个部分。如果你想要鼓励自由探险的感觉，比如某些娱乐或教育网站，那自然结构可能会是个好的选择；但是，如果你的用户下次还需要依靠同样的路径，去找到同样的内容，那么这种结构就可能会把用户的经历变成一次挑战。



自然结构



线性结构



**线性结构** (sequential structures) 来自于你最熟悉的线下的媒体——事实上，你现在正在体验其中一种。连贯的语言流程是最基本的信息结构类型，而且处理它的装置早已被深深地植入我们的大脑中了。书、文章、音像和录像全部都被设计成一种线性的体验。在互联网中线性结构经常被用于小规模的结构，比如单篇的文章或单个部分；大规模的线性结构则被用于限制那些所呈现的内容顺序对于符合用户需求非常关键的应用程序，比如教学资料。





## 组织原则

节点在信息架构中是依据**组织原则**（organizing principle）来安置的。基本上，组织原则就是我们决定哪些节点要编成一组，而哪些节点要保持独立的标准。不同的组织原则将被应用在不同的区域和网站不同的层面。

以一个公司的信息网站为例，我们的树状结构中也许有最上层的类别，比如“消费者”、“企业集团”和“投资者”。在这个阶段，组织原则是“不同内容所针对的观众”。其他网站也许有另外的上层类别，比如“北美洲”、“欧洲”和“非洲”，使用地区作为另一种组织原则是满足全球用户需求的一种方法。

一般说来，在网站最高层级使用的组织原则应该紧密地与“网站目标”和“用户需求”相关。而在结构中较低的层级，内容与功能需求的考虑将对你所采用的组织原则产生很大影响。

例如，一个做新闻内容的网站经常以时间顺序作为它最显著的组织原则。实时性对于用户来说是唯一最重要的因素（用户希望在新闻网站看到关于时事的信息，而不是历史），对网站的创建者也同样重要（创建者必须强调这些内容的实时性才能在竞争中得以生存）。



结构的下一个层级是其他与内容紧密相关的因素。以体育新闻网站来说，内容也许被划分成像“棒球”、“网球”和“曲棍球”这样的类别；而更偏向于广泛兴趣的网站也许会有类似“国际新闻”、“国内新闻”和“地方新闻”这样的类别。

任何一种信息收集——不论它是包括两个项目、200个或是2000个——都有一个固定的概念性结构。实际上，这种概念结构通常不止一个。那也是我们必须解决的问题之一。我们所面临的困难不是创建一个结构，而是创建一个能与“我们的目标”和“用户需求”相对应的、正确的结构。

比如说，假设我们的网站包括了大量的汽车信息。一个可能的组织原则是考虑把信息按照汽车的重量来排列，这样用户看见的第一件事将会是数据库中最重的汽车信息，然后是排第二的，一直到最轻的汽车。

对于一般消费者的信息网站来说，这可能是一种错误的组织信息方式。大多数人，大部分时间，并不关心汽车的重量。对于这样的用户群来说，依据汽车外观、型号和类型来组织信息或许更为适当。另一方面，如果我们的用户是每天买卖、运送汽车到国外的专业人士，重量就变成了一个非常重要因素。对于这样的用户群来说，像汽车本身的燃料经济性和引擎类型就变得比较不重要或都根本无关紧要。





这些属性，在图书馆学的术语中，被称为**截面 (facets)**，而且它们几乎能为任何内容提供一套简单、灵活的组织原则。但是先前的例子说明，使用错误的截面可能比根本没使用截面更加糟糕。对于这个问题的一个常见反应，是将每一个有可能的截面都当作组织原则来使用，从而让用户自己去选择对他们而言是最重要的那个。

不幸地是，除非你的信息非常简单，只包括几个不同的截面，否则这种方法很快就会把信息架构变得既笨重又混乱。由于用户有太多的方法可以将信息排序、过滤，这就造成没有人能找到自己想要的东西。这样的负担不应该丢给用户——让他们自己使用所有的属性来排序，并且挑选什么是重要的——这样的负担应该由我们来解决。战略告诉我们“用户的需求是什么”，范围则告诉我们“什么样的信息将满足那些用户需求”。在创建结构时，我们就要具体地识别出用户心目中至关重要的那些信息。成功的用户体验就是能事先知道用户的期望。

## 语言和元数据

即使结构完全准确地代表了用户对网站的理解，用户也仍然无法在结构中找到他们想走的路，这是因为他们无法了解你的**命名原则 (nomenclature)**：描述、标签和网站使用的其他术语。因此，“使用用户的语言”并且“保持一致性”是非常重要的。我们过去常常用来强调一致性的工具被称为**控制性词典 (controlled vocabulary)**。



控制性词典是网站使用的一套标准语言。这是用户研究中很重要的一个领域。与用户谈话并了解他们的沟通方式，是开发出一个让用户感到命名原则自然的系统的最有效方式。创造一个反映了用户语言的控制性词典（并遵守它）是防止企业内部专用术语侵入网站的最佳方法，那些专用术语只会混淆你的用户。

控制性词典也有助于建立起贯穿所有内容的一致性。无论内容创建者是坐在邻近位置，还是坐在不同国家的办公室内，控制性词典提供了一个明确的资源以确保大家都能使用用户的语言。

控制词汇的另一种较为精细的方法，就是创造类词词典（thesaurus）。与简单列出所使用词汇的清单不同，类词词典会提供常用的、但未纳入该网站标准用语的词汇以供选择。就类词词典来说，你可添加内部专用术语、速写语、俚语或缩写词等对其相对应的词汇进行补充。类词词典可能还包含词汇之间的类型关系，并提供更广义、更狭义或相关词汇的建议。将这些关系记录存档会让你对内容概念的整个范围有了一个更完整的印象，同时也有可能推荐你使用另外的结构方法。



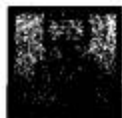


使用控制性词典或类词词典对于建立包含元数据 (metadata) 的系统特别有用。元数据这个名称，简单地说就是“关于信息的信息”，即以一种结构化的方式来描述内容。

假设我们正分析一篇关于志愿消防队如何使用最新产品的文章。这篇文章的元数据可能包括：

- 作者名
- 发布日期
- 内容类型（如：例如，案例研究或文章）
- 产品名称
- 产品类型
- 客户所在行业（如：志愿消防队）
- 其他相关信息（如：市政代办处或紧急情况服务）

准备好这些信息让我们能把各种各样可能性都考虑周全，包括那些没有它们就会很难（不是完全不可能）实施的结构方法。简而言之，掌握内容信息越详细，在建设信息架构时灵活性就越高。假如“紧急服务”突然变得很有潜力，从而成为公司想要扩展的新兴市场，那么有这样的元数据将帮助我们迅速地运用已有的内容创造出适应用户需求的一个新部分。



但是，如果数据本身不一致，那么建立技术系统来收集和跟踪全部的元数据就不会对我们有任何帮助。这正是需要控制性词典出现的地方。在你的内容中，每一个独特的概念都对应了一个固定的词，通过这种对应，就可以依靠自动化来帮助你定义内容元素之间的联系。你的网站可以动态地将一组与某个主题有关的页面链接到一起，没有任何人需要额外做什么，只要它们的元数据始终一致地使用同样的词组。

另外，好的元数据比基本的全文搜索引擎能提供更可靠的搜索结果，它能帮助用户在网站中更快速地找到信息。搜索引擎是强有力的，但一般说来它们非常非常笨——给它们一个字符串，它们寻找的几乎就是那个一模一样的字符串，搜索引擎并不了解任何一个字符串的意义。

将搜索引擎与类词词典联接起来，再加上元数据，就能让搜索引擎变得更聪明。搜索引擎使用类词词典来区分“禁用词”与“首选词”，接着它从元数据中查找这些“首选词”。与搜索结果为零相反的是，用户得到高度精准的、相关的搜索结果——甚至还可以推荐一些用户可能感兴趣的相关主题。

## 团队角色和流程

文档一定要描述清楚网站的结构——从命名原则和元数据的具体细节，到信息架构和交互设计的整体概况——根据项目复杂度的不同，可以有很大的不同。对于内容涉及到很多层结





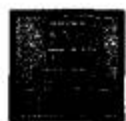
构的项目，简单的文字概述可能是纪录结构的一个最有效的方式。在某些情况下，报表和数据库这样的工具会被用于帮助捕捉复杂结构的细微差异。

然而信息架构或交互设计的主要文档是示意图。视觉化地呈现结构，对我们而言，这是表述“分支、群组、组件之间的联系”的一种最高效的方式。网站结构总是很复杂的，用文字去表达这些复杂的概念，有谁会真的去看呢？

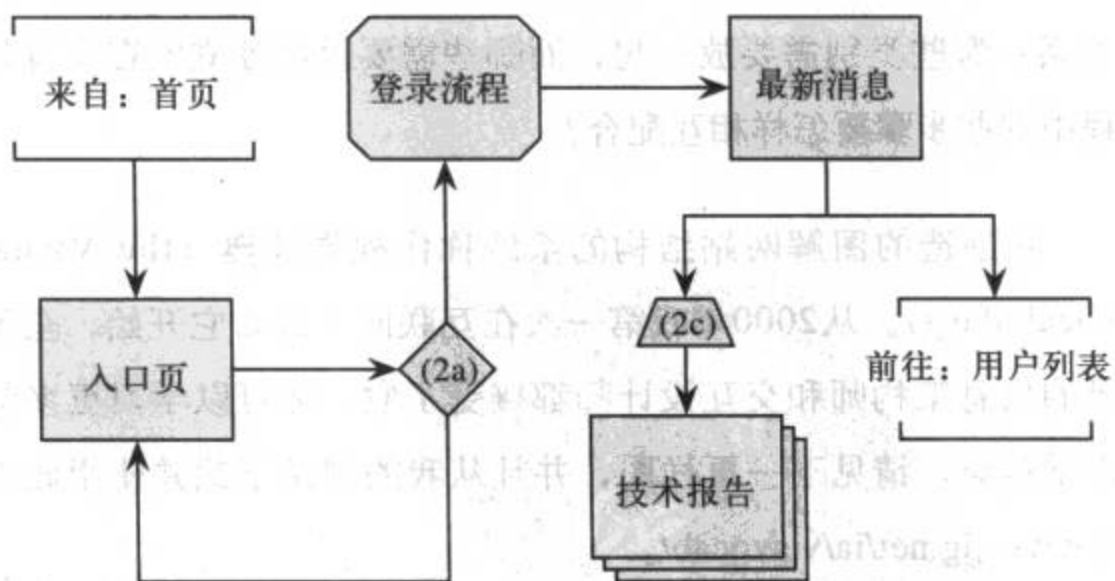
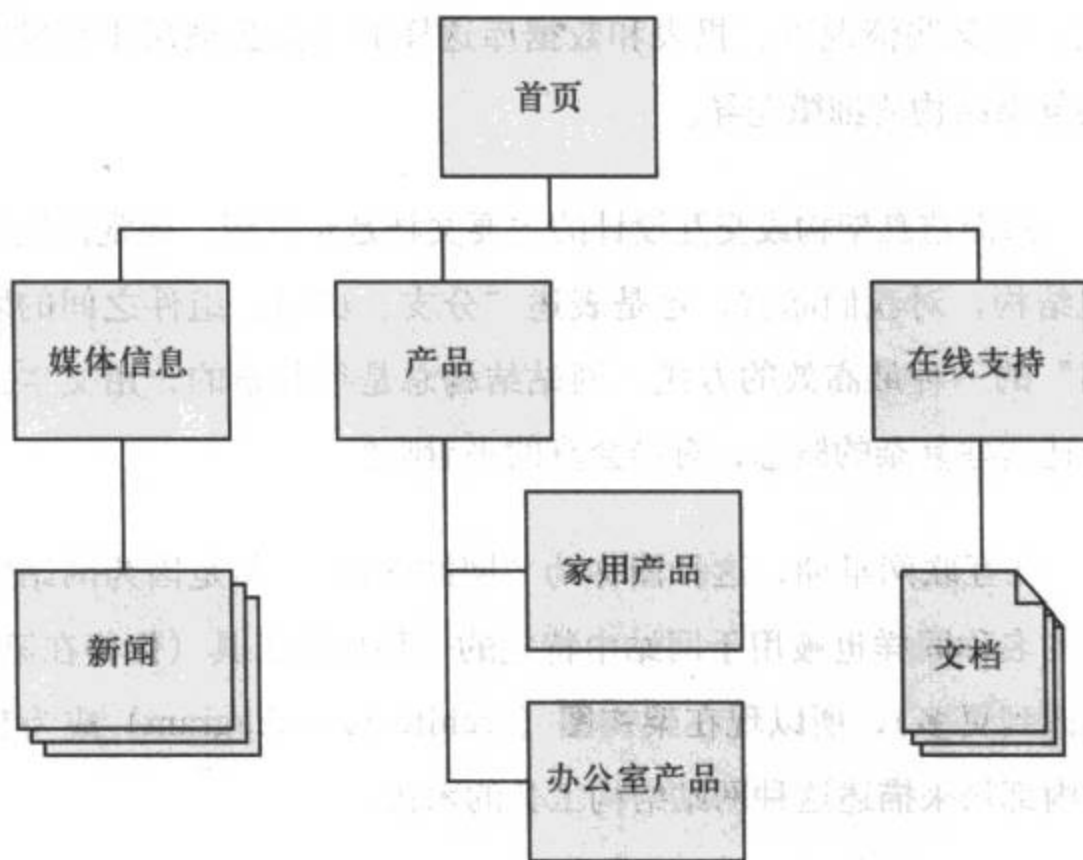
在互联网早期，这种图称为“网站地图”，但是因为网站地图的名称同样也被用于网站中特定的一种导航工具（你将在第6章读到更多），所以现在架构图（architecture diagram）成为我们内部用来描述这种网站结构工具的术语。

这种架构图并不一定要写明网站每一页的每一个链接。实际上，详细到这种程度的架构图，在许多情况下只会造成混淆并且屏蔽了团队真正需要的信息。架构图最重要的是记录概念关系：哪些类别需要放一起，而哪些需要保持独立？在交互过程中那些步骤要怎样相互配合？

我创造的图解网站结构的系统称作视觉辞典（the Visual Vocabulary）。从2000年我第一次在互联网上公布它开始，全世界的信息架构师和交互设计师都接受了它。你可以学习更多的视觉辞典，请见下一页样图，并且从我的网站下载并使用此工具[www.jjg.net/ia/visvocab/](http://www.jjg.net/ia/visvocab/)。



视觉辞典是一个提供从非常简单（上图）到非常复杂（下图）的示意结构系统。访问[www.jjg.net/ia/visvocab/](http://www.jjg.net/ia/visvocab/)了解更多。





由于交互设计和信息架构的概念在用户体验的领域里相对来说比较新颖，所以在网站开发团队内，没有人明确地为这些区域负责的现象仍旧非常普遍。因此，当你知道其实很少有网站可以证明它的结构是经过事先规划的时候，你也不会感到惊讶了。

负责结构的任务常常是顺便交给了某个人而不是通过明确地规划。“倒底谁在负责这件事”最后通常取决于企业的文化或项目的本质。在早期的互联网历史中，网站通常由企业已有的技术人员创建和管理，而这是企业内部变动发生较慢（或资源非常有限）的地方，这可能仍然是今天的实际情形。

对于内容量繁重，或那些起初将创建网站看作营销活动的企业，决定网站结构的责任被放到了内容建设、编辑或是公共关系部门。如果企业习惯由技术人员主导，或企业文化是技术导向的，那么结构的责任一般会落到技术项目负责人的身上。



招聘一名全职负责结构问题的专家可以使每个项目都得到好处。有时这个人的职位名称被称作“交互设计师”，但通常他们是指“信息架构师”。不要让名称把你搞糊涂了——虽然一些信息架构师确实主要负责创建内容网站的组织和导航结构，但是大多数情况下，信息架构师也具有某种程度的交互设计经验。实际上，拥有信息架构师头衔的某些人更接近于交互设计专家。

企业正在进行中的工作的数量，也许还不足以招聘一名全职的信息架构师作为团队的长期成员。如果你的网站开发主要是内容的更新，而且你不需要定期对整个网站进行重新设计和新的开发，那么招聘信息架构师可能就不太合适。但如果你的网站将稳定而持续地增加新内容和新功能，那么一个全职的信息架构师，就能帮助你确保整个新增内容的过程能够最有效地满足用户需求与企业的战略目标。

你是否有一位专家来解决结构问题并不重要，重要的是这些问题能由某个人来负责。不论你是不是做过这方面的规划，网站都会有一个结构。一个建立在完整定义、结构明确的规划上的网站，会减少频繁检查维护的工作量，也能为网站的所有者带来可见的结果，同时还能满足他们的用户的需求。





## 推荐阅读

Cooper, Alan. *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Sams, 1999.

Norman, Donald A. *The Design of Everyday Things. Revised edition*. Currency/Doubleday, 1990.

Rosenfeld, Louis and Peter Morville. *Information Architecture for the World Wide Web*. 2nd edition. O'Reilly, 2002.

Web resources: [www.jjg.net/elements/resources/](http://www.jjg.net/elements/resources/).



# 第6章

## 框架层

界面设计、导航设计和信息设计

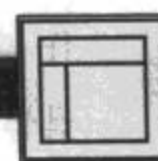




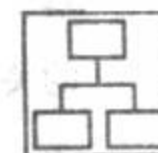
在充满概念的结构层中开始形成了大量的需求，这些需求都是来自我们的战略目标的需求。在框架层，我们要更进一步地提炼这些结构，确定很详细的界面外观、导航和信息设计，这能让晦涩的结构变得更实在。



表现层



框架层



结构层



范围层



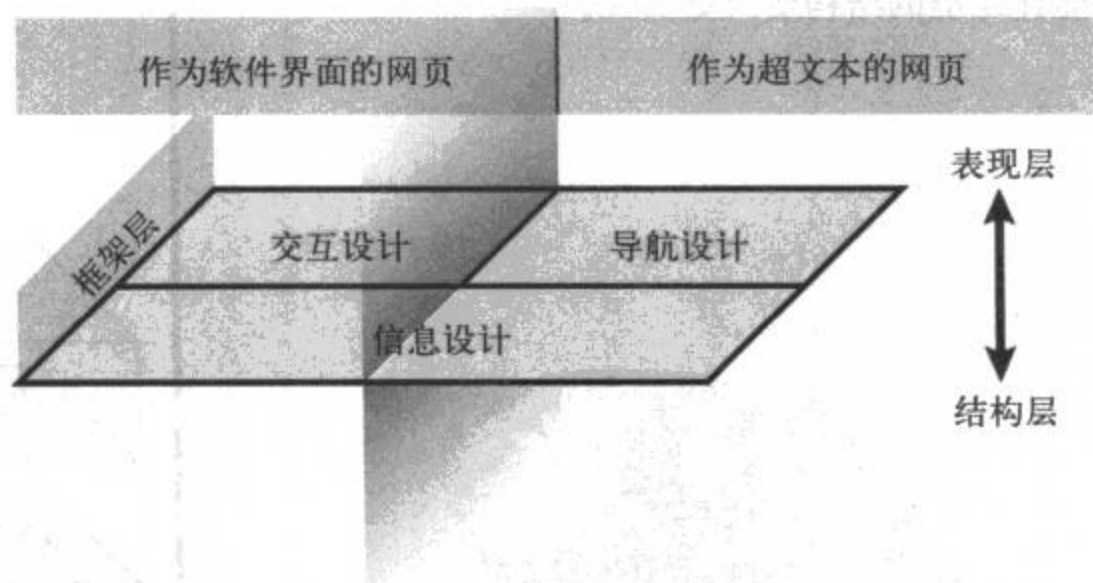
战略层



## 框架层定义

前面章节中的结构层界定了我们的网站将用什么方式来运作；框架层则用于确定用什么样的功能形式来实现。除了解决具体的这些议题，框架层还要处理更精准的细节问题。在结构层，我们看到一个较大的架构和交互的设计；在框架层，我们的关注点几乎全部在不同的页面和它们的组件上。

在软件方面，我们通过界面设计（interface design）来确定框架——一个大家所熟知的、“按钮、输入框和其他界面控件”的领域。但是在信息空间方面，要解决的是一个独一无二的问题：导航设计（navigation design），这是专门用于呈现信息的一种界面形式。最后，信息设计（information design）是软件和信息两方面都必须要做的，它用于呈现有效的信息沟通。





这三种要素紧密地结合在一起——比这本书中提到过的、任何两个要素之间的关系，都要紧密得多。在面对“界面设计”问题的时候，首先要考虑“信息设计”是否太模糊，或者遇到的“信息设计”问题最后变成“导航设计”的问题，这都是很常见的情况。

即使这些边界有时会变得模棱两可，但是，把它们定义成独立的领域仍然能帮助我们更准确地评估是否已经找到了合适的解决方案。“好的导航设计”不能纠正“不好的信息设计”，如果我们不能分辨这些问题的类型，那么就无法知道是不是真正解决了它们。

如果这涉及提供给用户做某些事的能力，则属于“界面设计”。界面的意思是说，通过它，用户能真正接触到那些“在结构层的交互设计中”确定的“具体功能”。

如果是提供给用户去某个地方的能力，则是“导航设计”。信息架构把一个结构应用到我们设定好的“内容需求清单”之中，而导航设计则是一个用户能看到那个结构的镜头，这就表示，通过它，用户可以“在结构中自由穿行”。

如果是传达想法给用户的话，就是“信息设计”。它是这个层面中范围最广的一个要素，所涉及的事情几乎是到目前为止，我们在软件界面和超文本系统两边看到过的方方面面。信息设计跨越了“以任务为导向”的软件系统和“以信息为导向”的超文本系统的边界，因为无论是界面设计还是导航设计，都不可能在没有“一个良好的信息设计的支持”的前提下取得成功。



## 习惯和比喻

习惯和反射作用是我们与这个世界交互时的基础——的确，如果我们减少很多反射作用的话，我们每天能完成的事情也会大大减少。你能想像开一辆“永远不会比你第一次开的时候更容易”的车会怎么样吗？你的驾驶技术、烹饪技术或使用计算机的技术——在没有被那些庞大的、注意力高度集中的任务弄得精疲力尽的情况下——都依赖于大量的、细小的反射作用的积累。

习惯使我们可以把这些反射作用应用到不同的环境中。我曾经有一辆车，不管我的哪个朋友驾驶它都不可避免地会遇到一些麻烦。当他们启动它时，他们做的第一件事是“清洗挡风玻璃”。这不是因为他们认为挡风玻璃脏了（尽量它有可能是脏了）；而是因为他们想打开前灯。我车上的控制方式与他们过去的习惯完全不一样的。

电话是另一个说明习惯的好例子。时不时地，制造商总是试着设计一些不符合“三行四列”按钮布局的电话，比如两排按钮，每排六个，或三排按钮，每排四个。“圆形”布局按钮的电话仍然不断地生产出来，但是这些随着它们所依赖的转盘式拨号电话的消失而逐渐消失，进而在被遗忘技术的迷雾中了。





看上去布局不应该和习惯有那么大的冲突，但是它的确有。如果你统计过“一个用户花了多少时间去研究非标准电话上的按钮”的话，每一次打电话时大约需要花三秒左右。这并不是很大的差异——但是对于用户来讲，这三秒钟不仅仅是浪费时间那么简单。这三秒钟完全是被挫折感占据的，一件“下意识就能完成的事情”变成“难以忍受的缓慢”，仅仅是因为用户脚下的“习惯魔毯”被人取走了。

事实上，电话中“三乘四”的数字矩阵是非常根深蒂固的习惯，因为它已经成为其他设备的一种标准，而不仅仅是电话，这些设备包括微波炉或电视机和录像机（VCR：Video Cassette Recorder）的遥控器（有趣的是，电话的面板并不是这个领域的唯一标准：老式计算机所使用的“十键”标准布局<sup>①</sup>，把电话按键的数字顺序颠倒了过来。现在，在计算器、电脑键盘、自动提款机、收银机和一些特殊的数据输入的应用程序，比如存储系统上得到广泛应用。由于这些标准都使用三乘四的矩阵，因此人们就已经相对容易地适应了它们，虽然这个单一的标准并不一定是真正的最佳解决办法）。

这并不是说，每一个界面问题的解决办法都必须毫无条件地死守这些习惯。当一种不同的方式有很明显的益处时，你反而应该试着谨慎地违背一些习惯。建设一个成功的用户体验，要求你在做每一个决定的时候都有充分的、明确的理由。

---

① “十键”标准布局（“10-key” standard）是按照“7-8-9，4-5-6，1-2-3”来排列的。



让你的界面和你的用户已经熟悉的那些习惯保持一致很重要，但是更重要的是，让你的界面与它自身保持一致。网站所具有的特性的概念模型能帮助你保持内部一致性。如果有两个特性都具有同样的概念模型，那么它们很可能就有相类似的界面要求。在这两个地方使用同样的习惯，能使熟悉了其中一个特性的用户很快适应另外一个。

即使是在概念模型不同的地方，各个概念模型都会用到的模块也应该以相似的方式来对待（如果不是相同的话），无论它们出现在哪儿。就像“开始”、“结束”、“返回”或“保存”一类的概念，会在很大的范围内出现。给它们一个统一的处理方式，让用户可以应用他们已经从系统的其他部分所了解到的知识，这有助于用户更快地达到自己的目标，更少地犯错。

就像不应该过于强调交互设计背后的概念模型一样，你应该抵制在网站四周建立起比喻（metaphor）的冲动。对你的网站的特性来说，比喻是很可爱、很有趣的，但是它们几乎无法像你想像的那样产生作用。事实上，它们根本起不了作用。

在某些情况下，你可以为了某个功能，而模仿现实世界的某个物体来设计界面。还记得Slate的导航，使你能像翻阅真正的杂志一样“翻过”一页吗？在大部分真实世界中的界面和导航设备，是只能在真实世界使用的：物理的、材料性能等等。网页同样也有一些制约。





将网站的特性和人们在真实世界中曾有的经验相对比，看上去是一个有助于人们掌握“那些特性是什么”的好方式。但无论如何，这种方式往往不能揭示特性的本质，而反而会使其更加混淆。即使特性和它所代表的比喻之间的联系，对于你来说是显而易见的，但它也仅仅是你的用户可能会联想到的、众多比喻中的一个——尤其当这些用户来自于和你完全不同的文化背景时。一个“电话的小图片”表示什么？是说我能用这个网站打电话？还是检查我的语音信箱？或者是交电话费用？

当然，网站的内容应该提供一定程度的上下文，从而帮助用户更好地猜测你所采用的比喻试图代表什么样的特性。但是你提供的各种各样的内容和功能越多，这些猜测就变得越不可靠——大部分时候，一部分用户总是猜不对。更好（也是更简单）的做法就是完全去除猜测的成分。

避免比喻，就是要减少对用户在“理解和使用网站功能”时对心理的要求。用一个电话簿的图标来代表真实的电话号码簿也许还行得通，但用一个咖啡店的图片代表聊天区域可能就会有问题。



## 界面设计

成功的界面设计是那些能让用户一眼就看到“最重要的东西”的界面设计。而另一方面，不重要的东西，不应该被注意到——有时候则是因为它们根本就没有出现在那儿。设计复杂系统的界面所面临的最大挑战之一，是弄清楚用户不需要知道哪些东西，并减少它们的可视性（或者完全把它们排除出去）。

对于在程序开发方面有一定背景的人来说，这种思考问题的方式要求他们改变一些即有的思路。因为它与他们过去的思考方式完全不同。好的程序员总是要考虑到很少发生的场景（在开发术语里称为“边缘情况”）。毕竟，对于程序员来说最有成就感的事是建立一个“永远不会出错的系统”，但是不考虑到边缘情况的程序，很可能就在这些极端情况发生时出现错误。所以经验丰富的程序员，总是平等地对待每一种可能性，不管它代表了一个用户还是一千个用户。

这种思路对于界面设计是行不通的。一个给出少量极端情况的界面，等于给大多数主要用户提供一个设计不良的界面，而让少数用户满意。一个设计良好的界面是要组织好用户最常采用的行为，同时让这些界面元素用最容易的方式获取和使用。





这并不是说每一个界面问题的解决办法都是把用户最有可能点击的按钮设计成这个页面最大的一个。界面设计可以采用各种各样的技巧，让用户完成目标的过程变得容易。一个简单的技巧，就是在这个界面第一次呈现给用户的时候，仔细考虑每一个选项的默认值。如果你理解了用户的任务和目标，认为他们中的大多数人都希望在快速搜索的结果中看到更多细节的话，保持“显示更多细节”复选框为默认选中状态，就意味着大部分人都会对他们所得到的结果感到满意，无论他们是否花时间去阅读复选框的标签并作出自己的决定（另一个更好的做法是，能自动记住某个用户最后一次选择状态的系统，但这有时候比在界面上出现必要信息对技术有更高的要求，而且有可能的结果是对于某些开发团队来讲，是不切实际的、不可能成功实施的工作）。

在网页上应用最多的两个主要技术，HTML和Flash，其技术的自身局限使得我们可选择的界面选项受到限制。这同时具有好和坏两个方面。坏的一面是因为它限制了我们发明的机会——一些在传统的桌面软件中很常见的界面方式根本不可能在网页上实现。但是这种情况同样也具有好的一面，因为学习相对较小的标准控制方式的用户，可以把他们的知识应用到更大范围的网站中去。



虽然HTML最初是用于简单的超级文本信息，但人们很快就发现它存在提供更多交互性方面的潜力。在它发布之后没多久，HTML的一小部分元素就成为了标准界面元素：

**复选框 (checkbox)** 允许用户独立地选择各个选项。

- ☐ 复选框是独立的
- ☒ 所以它们可以是一小组
- ☐ 或只有一个

**单选框 (radio button)** 允许用户从一组互斥的选项中选择一个。

- ☐ 单选框
- ☐ 总是以一组的形式出现
- ☐ 并且用于选择
- ☒ 互不相容的选项
- ☐ Burma-Shave

**文本框 (text field)** 允许用户——等待着用户——输入文字

文本框能让你输入文字

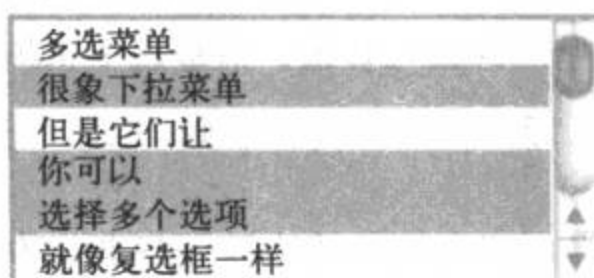




**下拉菜单** (dropdown list) 提供和单选框相似的功能，但是它们在一个更紧凑的空间中完成这件事，允许更有效地呈现更多的选项。

下拉菜单的工作方式类似单选框

**多选菜单** (list box) 提供和复选框相似的功能，但是它们在一个更紧凑的空间中做这件事（因为多选菜单有滚动条）。和下拉菜单相同，这使得多选菜单更容易支持大量的选项。



**按钮** (action button) 可以做很多不同的事情。通常情况下，它们告诉系统接受用户通过其他界面元素提交的所有信息，并用这些信息来做一些事情——采取动作。

按钮执行动作



Flash提供了一套同样的基本元素，但是由于最初是作为动画工具来使用的，所以它具有更强的灵活性，使界面对用户的响应更加积极。因此，Flash界面使设计过程增加了更多的选择，这也使得它们似乎很难被正确地应用。

妥善处理所有不同的界面元素，并从它们中间选择合适的那个，这不可避免地会涉及权衡问题。的确，下拉菜单比起一组单选框来能为你节约页面的一些空间，但是它也使用户不能一眼看到可选的选项；让人们输入他们想要搜索的分类名称，也许会降低数据库的载入负担，但是这个负担却转移到了用户身上；如果不管怎样都只有六个选项，也许一些复选框会更好。

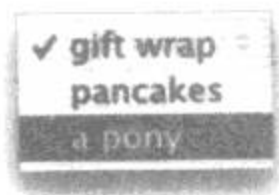
设计网页界面完全就是做这样一些事：为用户想完成的任务选择正确的界面元素，通过一种能迅速理解和易于使用的方式，把它们放置到页面上去。网站上的任务通常都会横跨多个页面，每一个页面都包含了一组不同的、需要用户去处理的界面元素。“哪个功能应该在哪些页面上完成”是结构层的交互设计的主要工作；“这些功能在页面上如何被实现”则是界面设计范围内的事。

下拉菜单（左）由于在视觉上隐藏了重要的选项，可能会妨碍用户。单选框（右）很容易显示所有可选项，但它们需要更多的界面空间。

Additional options:



Additional options:



Additional options:





当界面不仅仅要从用户那儿收集信息，还必须要向用户传达一些信息的时候，信息设计会在这里扮演一个角色。“错误消息”是在设计有效的界面时经常出现的消息设计问题；提供“说明信息”是另外一个，因为最大的挑战是让用户真正去阅读这些说明文字。任何时候系统都必须给用户一些信息，来帮助他们有效地使用这些界面——不管是因为他们操作错误还是因为他们是第一次使用——这是信息设计的问题。

## 导航设计

导航设计看上去像是一件很简单的工作：在每个页面上放一些允许用户浏览整个网站的链接。如果你去掉界面，无论如何，导航设计的复杂性就会变得显而易见。任何一个网站的导航设计都必须同时完成以下三个目标：

- 首先，它必须提供给用户一种在网站间跳转的方法。由于一般来讲把每个页面和别的都链接起来是不现实的（即使是现实的，它也不是一个好方法），导航元素就必须选择那些能促进用户行为的——也就是说，这些链接必须也是真实有效的。



- ▶ 其次，导航设计必须传达出这些元素和它们所包含内容之间的关系。仅仅提供一个链接的列表是不够的。这些链接相互之间有什么关系？是否其中一些比别的更重要？它们之间相关的差异在哪里？这些传达出来的信息对于用户理解“哪些选择对他们是有用的”是非常必要的。
- ▶ 第三，导航设计必须传达出它的内容和用户当前浏览页面之间的关系。其他的那些内容对于我正在浏览的这个页面有什么影响？这些传达出来的信息帮助用户去理解“哪个有效的选择会最好地支持他们的任务或他们想要达到的目标”。

在物理空间中，人们可以在某种程度上依靠天生的方向感来给自己定位（当然，有些人永远失去了这种感觉）。但是这些帮助我们在真实世界中找到方向的大脑机制（让我们来看看，我觉得我进来的那个入口应该在我的左后方），在信息空间中根本起不到作用。

这就是为什么在网站中，清晰地告诉用户“他们在哪儿”以及“他们能去哪儿”是非常重要的原因。用户在信息空间中的自身定位具有什么样的程度，目前还在讨论中：一些人很坚决地主张当用户访问网站的时候在他们的脑子里已经有了一张大概的地图，就像他们在五金商店和图书馆一样；另一方则认为用户完全依赖于导航以及在他们面前的一些指示线索，好像他们在网站中走过的每一步，都会在走过以后不久逐渐地从记忆中消退一样。



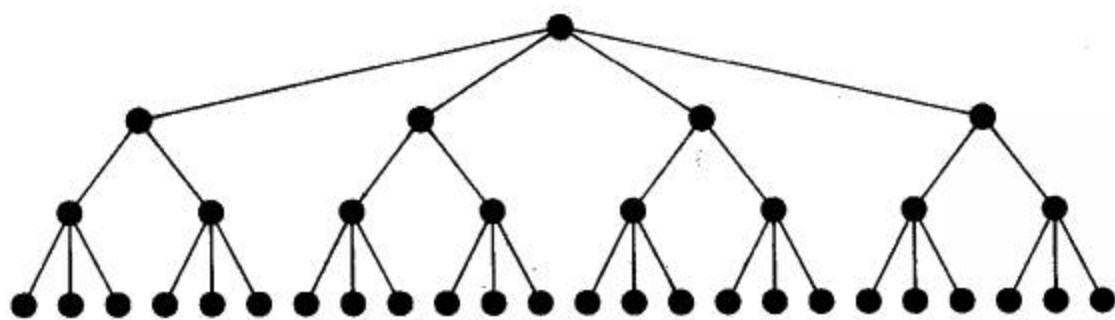


我们仍然不知道人们是怎样在脑海中记忆（或能记住多少）网站的结构。在我们弄清这件事以前，最好的方法就是假设用户不会将上一页的信息带到下一页中（毕竟，如果一个类似google一样的公众搜索引擎收藏了你的网站，任何一个页面都可能成为你的网站的入口）。

大多数的网站实际上都会提供一个多重的导航系统（navigation system），每一个都要完成在各种情形中成功引导用户的任务。在实践过程中涌现出了几种常见的导航系统。

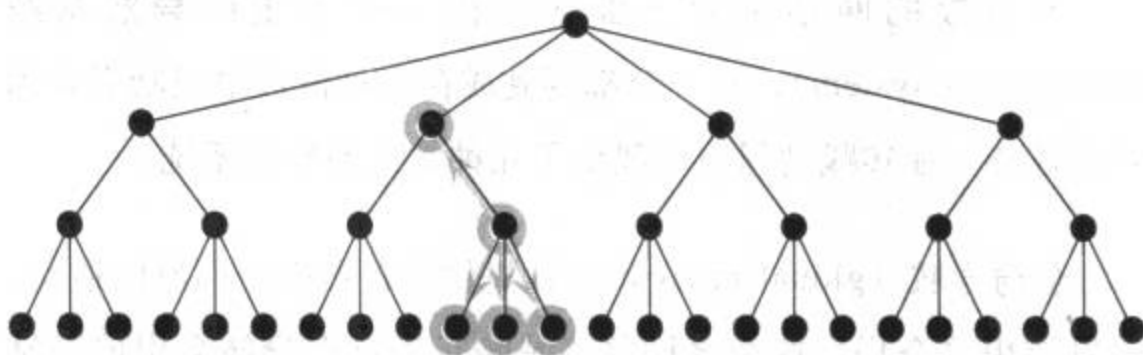
**全局导航**（global navigation）提供了覆盖整个网站的通路。这里使用“全局”这个名词并不是暗示着这个导航会出现网站的每一个页面中——即使这不算是个坏主意（我们使用“固定”一词来表示贯穿整个网站的导航；再次提醒的是，固定的元素不一定是全局的），相反地，全局导航提供的是用户最有可能需要从网站的最终页面到其他什么地方的一组关键点。不管你想去哪儿，你都能从全局导航中（最终）到达那儿。

全局导航



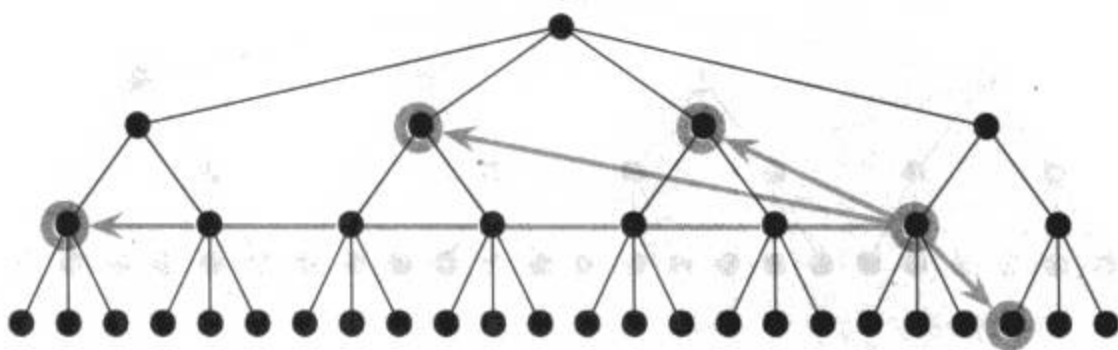
**局部导航** (local navigation) 提供给用户在这个架构中到“附近地点”的通路。在一个严格的层次结构中，局部导航可能只提供一个页面的父级、兄弟级和子级通路。如果你的架构反映了用户对这个网站的内容结构的思路，那么局部导航通常都会比其他导航系统更有用。

局部导航



**辅助导航** (supplementary navigation) 提供了全局导航或局部导航不能快速达到的相关内容的快捷途径。这种类型的导航提供了一些分类方面的好处（允许用户转移他们浏览时的方向，而不需要从头开始），同时仍然能让网站保持一个主要的层级结构。

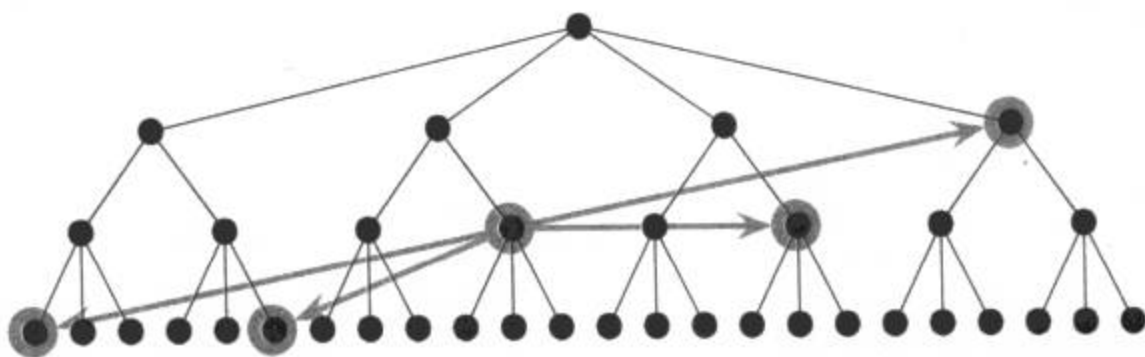
辅助导航





**上下文导航** (contextual navigation, 有时也叫“内联导航” inline navigation) 是嵌入页面自身内容的一种导航。这种类型的导航——比如, 一个在页面文字中的超级链接——常常没有得到充分利用 (或总是被错误地利用)。用户认为, 他们需要另一些信息的时候恰恰是在他们正在读文本的时候。与其强迫你的用户去扫描右侧的导航元素——或者更糟的是, 让他们不得不求助于搜索引擎——还不如放一些相关的链接在他正在读的地方。

所有这些方法都要回到战略层去看一看, 对你的用户和他们的需求理解得越准确, 你的上下文导航就能设计得越高效。如果它们不能明确地支持用户的任务和目标——如果你的文字中塞满了超级链接, 那么用户就会不知道哪个是他们需要的——则上下文导航将 (很正确地) 被看成一团乱麻。



上下文导航



**友好导航** (courtesy navigation) 提供给用户他们通常不会需要的链接, 但它们是作为一种便利的途径来使用的。在物理世界, 一个零食商店常常把它的营业时间摆放在入口处。对于大多数时候的大部分购物者来说, 这种信息并不总是有用的: 任何人都可以很快地告诉他这个商店是否正在营业中。但是你需要知道的是, 这种信息在他们确实需要的时候就能快速有效地帮助到他们。联系信息、反馈表单和法律声明的链接通常都放置在友好导航中。

一些导航并没有包含在页面结构中, 而是以它们自己的方式存在, 独立于你的网站的内容或功能。这些称为**远程导航** (remote navigation) 工具, 在用户被你所提供的其他导航系统搞得头晕脑涨, 或在看过你的导航系统否, 他们很快决定放弃, 想都没有想过要弄明白它们的时候, 大部分人都会想从远程导航中找到解决办法。





**网站地图** (site map) 就是一种常见的远程导航工具，它给用户一个简明的、单页的网站整体结构的快捷浏览方式。网站地图通常作为网站的一个分级概要出现，提供所有一级导航的链接，并与缩进的显示的、主要的二级导航链接起来。网站地图通常不会显示超过两个层级的导航——这之后更详细的内容往往超出了用户的需求（如果不是的话，那就是你高层次的结构定义错了）。

**索引表** (index) 是按字母顺序排列的、链接到相关页面的列表，它与一些书籍最后所列的索引表非常相像。这种类型的工具对于涵盖了不同主题的、大量内容的网站非常有效。在大多数情况下，一个网站地图和一个规划良好的结构应该足够了。索引表有时会为网站的某个部分特别开发，而不是要去覆盖整个网站的内容；如果你的网站试图相对独立地服务于拥有不同信息需求的不同用户，这个方法会非常有用。

## 信息设计

信息设计有时候会很难入手。它常常充当一种把各种设计元素聚合到一起的粘合剂的角色。最后，信息设计变成决定如何呈现这些信息，使人们能很容易使用或理解它们。

有时信息设计是视觉上的。对我们的用户来说，饼图是展现数据的最好方式？还是柱状图更好一点？是“望远镜”能充分表达网站的搜索概念，还是一个“放大镜”更容易理解？



有时信息设计涉及到“分组”或“整理”散乱的信息。我们通常把这方面的设计看成理所应当的，因为我们过去对信息设计的理解正是这种。举个例子来说，看看下面这个列表：

- 省份
- 职位名称
- 电话号码
- 地址
- 姓名
- 邮政编码
- 所在公司
- 所在城市
- E-mail地址

它看上去有一点让人迷惑，因为通常我们都会这样排列：

- 姓名
- 职位名称
- 所在公司
- 地址
- 所在城市
- 省份
- 邮政编码





► 电话号码

► E-mail地址

即使是这样的排列也可以更进一步地整理成这样：

► 个人信息

- 姓名
- 职位名称
- 所在公司

► 联系方式

- 地址
- 所在城市
- 省份
- 邮政编码

► 其他联系方式

- 电话号码
- E-mail地址

这个例子看上去很简单明了，现在我们用一个略有不同的列表来证明这项工作的挑战性：

► 功率限制

► 转轴尺寸

► 油箱容积



- ▶ 变速器类型
- ▶ 角速率中间值
- ▶ 底盘类型
- ▶ 最大输出功率

当然，最关键的是，用一种能“反映你的用户的思路”和“支持他们的任务和目标”的方式来分类和排列这些信息元素。在这些元素之间的概念的关系是真正属于微观的信息架构的，当我们必须要在这个页面上传达结构的时候，信息设计就呈现它的作用了。

## 指示标识

信息设计和导航设计结合到一起执行的是一个重要的功能——支持指示标识（wayfinding）。指示标识的想法来自于物理世界中公共空间的设计。公园、商场、公路、机场和停车场，大部分都从指示设施中获益。比如说停车场，有时候会使用“颜色编码”（color coding），来为人们提供线索，帮助他们记住停车的地方。在机场，标志、地图以及其他的指示设施，帮助人们找到他们周围的路径。

在网站中，指示标识通常会涉及到导航设计和信息设计。一个网站的导航系统不仅仅是要提供到网站不同区域的通路，还必须要成功地传达出这些选项。好的指示标识能使用户很快地得到一个心理图像，“他们在哪儿”、“能去哪儿”和“哪条路能使他们离自己的目标更近”。





指示标识的信息设计元素涉及页面元素，但并不是以导航功能来执行的。比如说，就像停车场一样，一些网站非常成功地使用了“颜色标识”来告诉用户他们正在看的是哪个部分（无论如何，颜色编码几乎从来没有被独立使用过——相反地，它用于在合适的位置强化另一套指示系统）。图标、标签系统和排版是另外的信息设计系统，有时用于帮助用户加强“你在这里”的感觉。

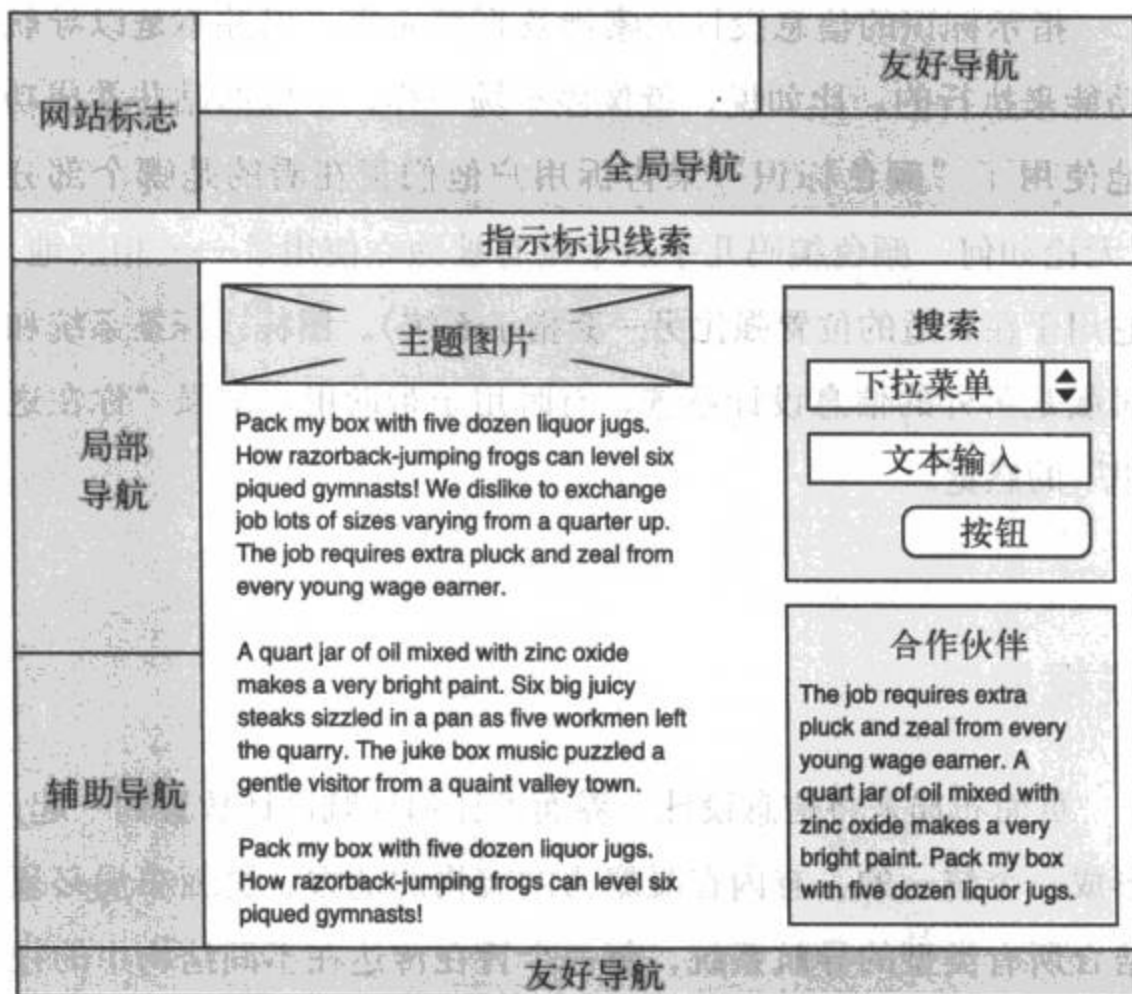
## 线框图

页面布局是将信息设计、界面设计和导航设计放置到一起，形成一个统一的、有内在凝聚力的架构的地方。页面布局必须结合所有类型的导航系统，每一个旨在传达在不同结构中的视图设计；任何一个在这个页面上的功能所需要的所有界面元素；以及支持以上这些内容的信息设计，也包括在这个页面上的内容的信息设计本身。

这一次需要平衡很多东西。这就是为什么页面布局被纳入一个详细的文档，并称为页面示意图或线框图（wireframe）的原因。这个线框图对是一个页面中所有的组成部分以及它们如何结合到一起的、露骨的描述（正如它的名字一样）。



线框图捕获所有在框架层做出的决定，并用一个文档来展现它们。它作为视觉设计和网站实施的参考来使用。线框图可以包括各种不同程度细节——你看到的这个是非常粗略的。



这些简单的线条画成的图一般要着重地注明，建议读者在必要的时候参考结构图表或其他交互设计文档、内容需求或功能规格说明，或者其他类型的详细文档。举例来说，如果一个线框图涉及个别已有的内容元素，它也许会给出指示，说明他们在哪儿能找到这些内容。另外，线框图通常还包括附加说明，用于说明在线框图和结构图表看得不太明显的网站行为。





现在，让我们再回去看看结构层的结构图示，它是这个项目的宏伟远景；而在框架层，线框图是正是展示那些远景如何完成的详细文档。线框图有时也需要结合导航规格的支持，以便能更详细、准确地描述各种导航元素的每一个组成成分。

对于更小或更简单的网站来说，一个线框图就足够作为所有即将建立的页面的模板。对于大多数项目来说，无论如何，都需要用多个线框图来传达复杂的预期结果。不过，你不需要为网站的每一个页面都准备一个线框图。正如结构设计流程允许我们把内容要素总结成各个种类一样，一个数量相对较少的标准页面类型将在绘制线框图的过程中慢慢浮现。

线框图在正式建立网站的视觉设计的流程中，是必要的的第一步，但是几乎每一个参与这个开发过程的人都会在一些任务点中使用它。负责战略层、范围层和结构层的设计者可以借助线框图来保证最终产品能满足他们的期望。真正负责建设这个网站的人，则使用线框图来回答关于网站应该如何运作的问题。

随着用户体验领域的不断成熟和发展，线框图的责任有时还成为企业内部某种口水战的主题。一些网站研发团队很鲜明地把这部分工作分成两个部分，由两个独立的角色（有时候是整个部门）“信息架构师”和“设计师”来承担。



线框图，成为信息架构和视觉设计汇集的地方，接着又变成了争论和纠纷的中心。信息架构师抱怨创建线框图的设计师将导航系统背后的结构描绘得模糊不清，不能正确反映结构的基本概念。视觉设计师抱怨绘制线框图的信息架构师将他们的功能减少成一个数码绘画师的角色，浪费了他们为信息设计问题带来的、在视觉传达方面的经验和专长。

当你有两个独立的信息架构师和设计师的时候，生产成功的线框图的唯一办法就是“协作”。在共同协作做出线框图的细节的过程中，双方都可以站在对方的角度来看待件事，并且在这个过程中还有助于及早地揭示出问题（而不是到后来，在这个网站正在建设的时候，每个人都纳闷为什么没有像计划中那样运作）。

所有的这些都使线框图听起来像是一件非常庞大的工作。这是不一定的。文档本身并不是目的，它只是达到目的一种手段。为了文档本身而创建文档不仅仅是在浪费时间——它可能还会降低生产力和打击工作积极性，根据你的需求来撰写正确级别的文档——同时不要欺骗自己可以用较少的文档糊弄过去——把文档从一件麻烦事变成一件有益的事。

我曾经完成过的、一些最成功的线框图，除了即时贴和画在上面的架构以外没有别的。对于一个设计师和程序员座位紧挨在一起的小团队而言，这种级别的文档已经完全足够了。但是当程序员对整个团队负责而不是一个人的时候——甚至这个团队在地球的另一端——那么就需要一些较为正式的文档了。





线框图是整合在结构层的全部三种要素的方法：通过安排和选择界面元素来整合界面设计；通过识别和定义核心导航系统来整合导航设计；通过放置和排列信息组成部分的优先级来整合信息设计。通过把这三者放到一个文档中，线框图可以确定一个建立在基本概念结构上的架构，同时指出了视觉设计应该前进的方向。

## 推荐阅读

Fleming, Jennifer. *Web Navigation: Designing the User Experience*. O'Reilly, 1998.

Spolsky, Joel. *User Interface Design for Programmers*. Apress, 2001.

Tufte, Edward. *Envisioning Information*. Graphics Press, 1990.

Veen, Jeffrey. *The Art & Science of Web Design*. New Riders, 2000.

Web resources: [www.jjg.net/elements/resources/](http://www.jjg.net/elements/resources/).





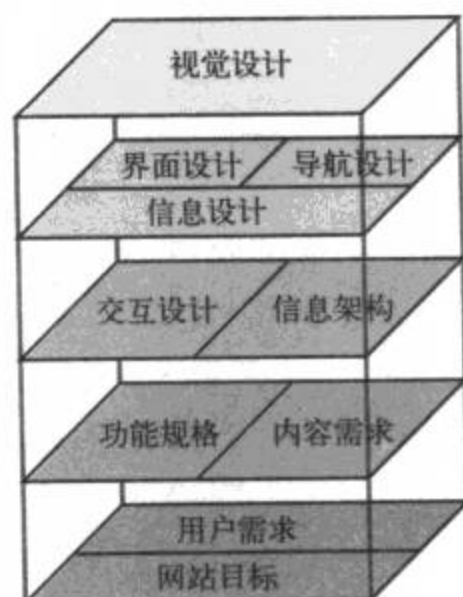
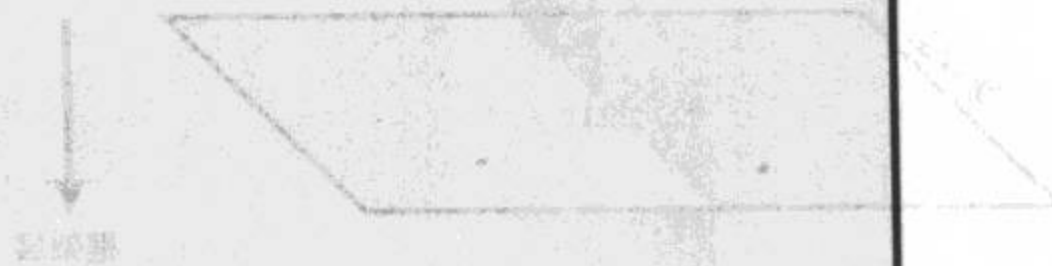
# 第7章

## 表现层

视觉设计



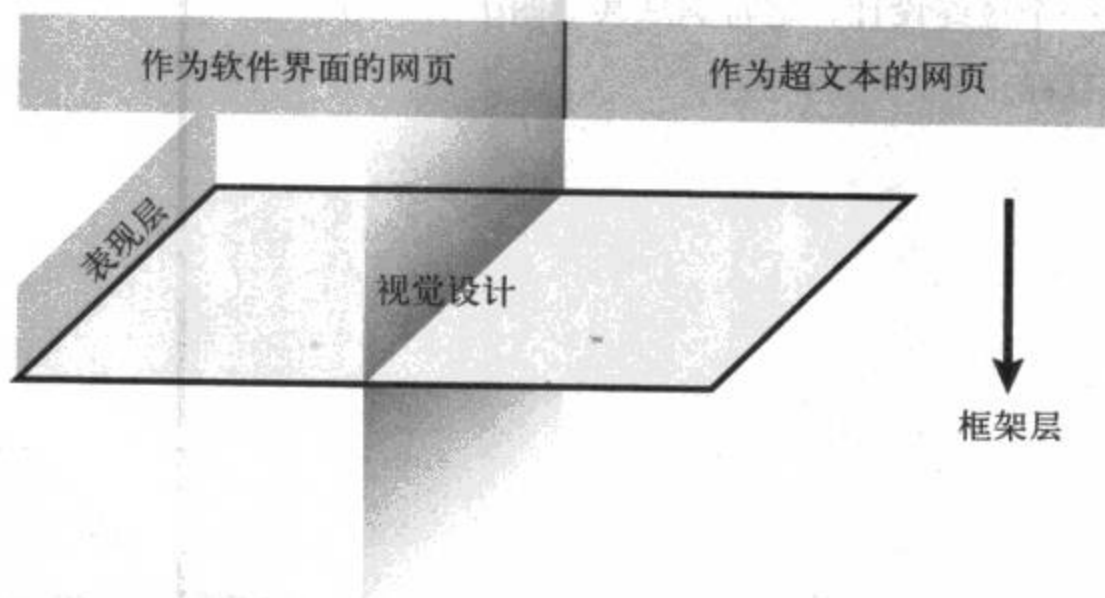




## 表现层定义

在框架层，我们主要解决放置的事情。界面设计考虑交互元素的布局，导航设计考虑在网站中引导用户移动的元素安排，而信息设计则考虑传达给用户的信息要素的排布。

再向上就是表现层，我们要在这里解决“弥补网站框架层的逻辑排布”的视觉呈现问题。举个例子来说，通过关注信息设计，我们决定了这些信息元素应该如何分组和排列；通过关注视觉设计，我们也决定了这种安排在视觉上应该如何呈现。





一开始，你可能会认为视觉设计就是一件很简单的事：美术。对于什么构成了视觉上的愉悦感觉，每个人都有不同的品味和想法，所以关于设计方案的每一次讨论总会归结到个人偏好上，不是吗？嗯，的确每个人对于美感都有不同的见解，但是这并不意味着设计决策就必须建立在所有参与者都认为“酷”的那个方案上。

代替用“什么具有美感”来评估一个视觉设计方案的是，你应该把注意力集中在它们的“运作是否良好”上。对于那些在之前的层面就确定的目标，视觉设计给予它们的支持效果如何？网站的外观是破坏了结构，使结构中的各个模块之间的区别变得不清晰、模棱两可了吗？还是强化了结构，使用户可用的选项清楚明了了？

比如，传达品牌的形象，这是一个网站的最常见的战略目标之一。传达品牌形象有很多种方法——你的网站所使用的语气，或网站功能的交互设计——但用于传达品牌形象的主要工具之一是视觉设计。如果你想表达的品牌形象是技术性和权威性，那么使用漫画字体和亮粉色可能就不是正确的选择。这不仅仅是一个美学的问题，而是战略定位的问题。



## 忠于眼睛

评估一个页面的视觉设计的简单方法之一，是提出这样的问题：你的视线首先落在什么地方？哪个设计要素在第一时间吸引了用户的注意力？它们对于战略目标来讲是很重要的东西吗？或者用户第一时间注意到的东西与他们的（或你的）目标是背道而驰的吗？

研究人员有时使用精密的眼球追踪（eyetracking）仪器来确定被测人到底正在看什么，以及他们的视线是如何在这个页面上移动的。无论如何，如果只是想略微调整一下某个页面的视觉设计，那么一般你只需要简单地询问一下人们就可以——甚至你自己也可以。有时这种方法不能提供最准确的结果，并且它也永远无法捕捉眼球追踪仪器所能提供的细微差别。但是在大多数时候，简单地询问问题是非常适合的。另一种找出主要设计元素的方法是眯着眼睛或斜着去看这个页面，直到你不能认出任何细节——或者走到房间的另一头从那个地方来看这个页面。

然后试着确定视线所停留的地方。如果你本人是被测者，那么一定要注意你的眼睛在页面周围的、无意识的移动。对于你正在看的东西，不要想得太多，只要让你的视线自然地落在页面上。如果另外还有一些被测者，让他们按被吸引的顺序来指出页面中的那些元素。





一般情况下，你会发现，人们的视线在这个页面上的移动方式遵循着相当一致的模式——毕竟，这些大多数是无意识的、本能的移动。如果测试的报告显示，某些人的视线移动和别人的模式不一样，那么他们有可能是没有真正察觉到自己眼睛的自然移动，或者他们只是把他们认为你想听到的事情告诉你了（或者两者都有）。

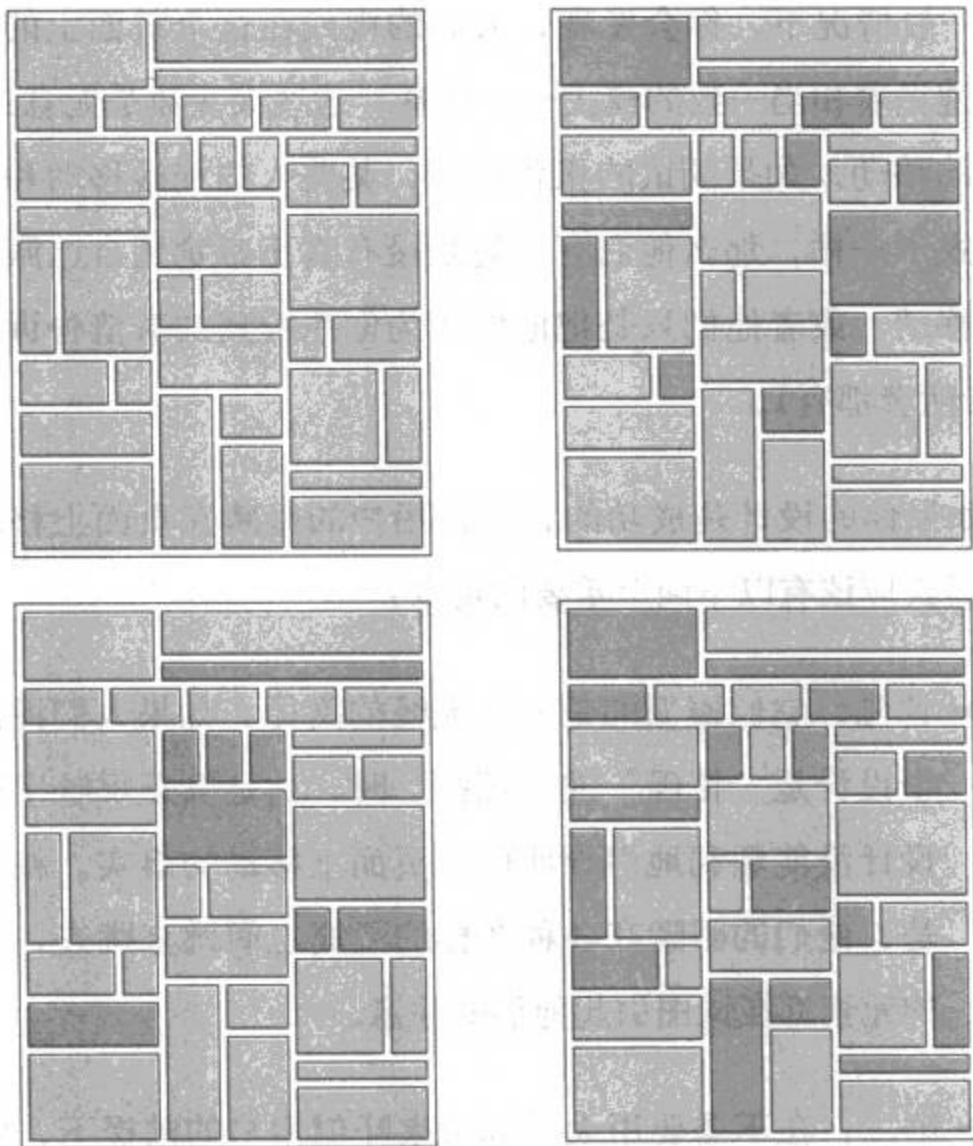
如果你的设计是成功的，那么用户的眼睛在页面上移动的轨迹模式应该有以下两个重要的特点：

- 首先，它们遵循的是一条流畅的路径。如果人们评论一个设计是“忙碌”或“拥挤”时，这是真正反映了这个设计没能顺利地引导他们在页面上移动的事实。相反地是，他们的眼睛在各种各样的元素之间跳来跳去，所有的元素都在试图引起他们的注意。
- 第二，在不需要用太多细节来吓倒用户的前提下，它为用户提供有效选择的、某种可能的“引导”。就像我们一直在说的那样，这些引导应该支持用户试图去完成他们的目标和任务。也许更重要的是，这些引导不应该分散用户对那些“能完成目标的信息或功能”的注意力。

用户在页面上的视线移动并不是随机的。它是一种所有人类共有的、对于视觉刺激而产生的、一系列复杂的原始本能反应。非常幸运地是，对于我们的设计师而言，这些反应并不是完全无法控制的——数百年以来，我们已经发展出了各种各样的有效的视觉手段，来吸引或分散注意力。



一个视觉上的中性布局(左上),没有任何一个元素突出。对比可以用来引导用户在页面上的视线(右上)或将他们的注意力吸引到几个关键要素上(左下)。过度的对比导致了混乱的视觉(右下)。



## 对比和一致性

在视觉设计中,我们用于吸引用户注意的一个主要工具就是**对比**(contrast)。一个没有对比的设计,会被看成一个灰色的、平凡的东西,导致用户的视线四处游离,而无法解决任何特别的事情。把用户的注意力吸引到界面中的关键部分,对比是一个重要手段,能帮助用户理解页面导航元素之间的关系。同时,对比还是传达信息设计中的概念群组的主要手段。





当一个元素在设计中显得与众不同时，用户就会注意到。这是他们不能控制的。你可以利用这个本能的行为，来使用户注意到那些“真正需要从这个页面的其他元素中突出的东西”。在网页界面上的错误提示通常会被融进页面的其他元素中；通过给文本一些不同的颜色（比如说，红色）或用一个醒目的图形将它们凸显出来，就能让整个界面完全不同了。

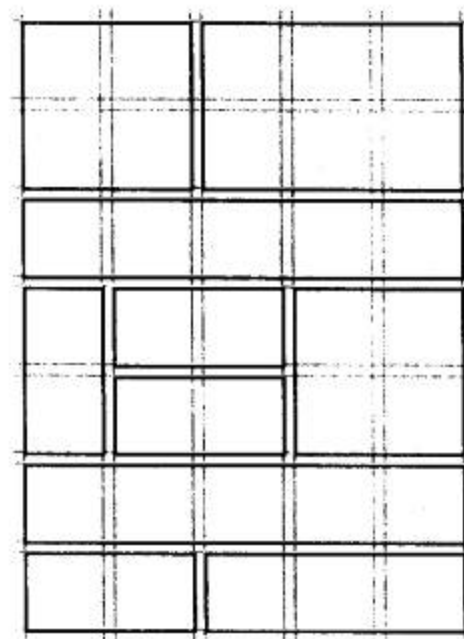
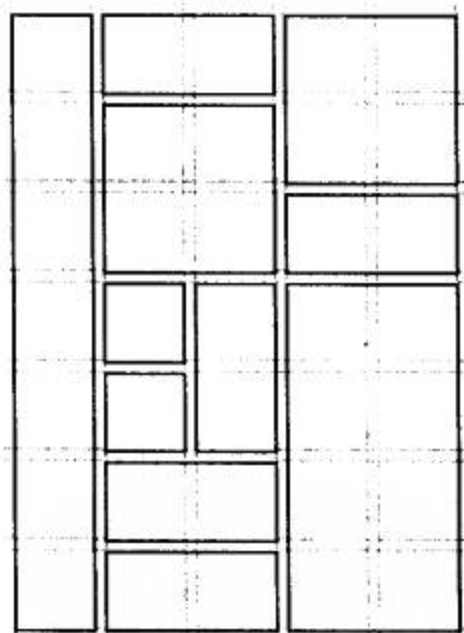
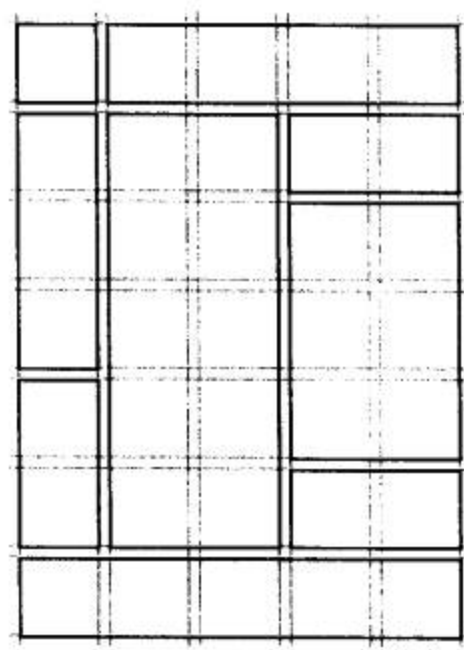
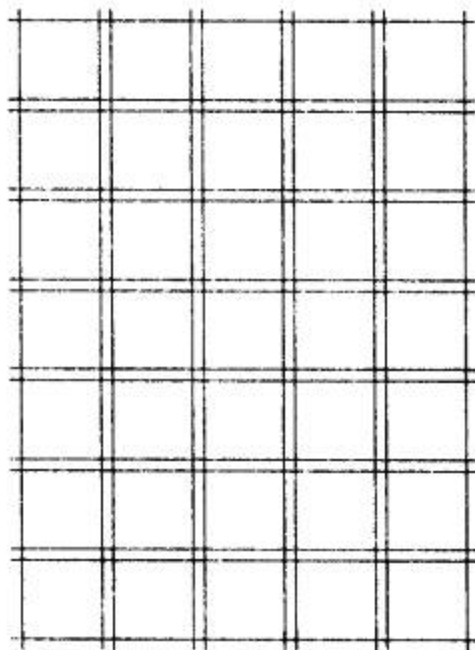
不管怎样，这些工作的总体策略是，让“差异”必须足够清晰，用户要足够分辨出某个设计选择是特意要传达一些信息的。当两个设计元素的处理相似而又不太一样的时候，用户就会困惑。“为什么这些会不一样？它们本来是一样的吗？也许只是弄错了。还是我应该在这里注意到什么东西吗？”而事实是，我们希望这两者都能抓住用户的目光，并且让他们认为这是有意为之的。

在设计中保持一致性（uniformity）是另一个重要的组成部分，它能使你的设计有效地传达信息，而不会导致用户迷惑或焦虑。“一致性”在视觉设计的许多不同方面都会起到作用。

将视觉元素的大小保持一致的尺寸，这可以使你在需要的时候把它们更容易地重新组合成一个新的设计。举个例子来说，如果你在导航中使用的所有图形按钮都是同一个高度，那么它们就可以在需要的时候被混合并匹配，而不会形成一个布局杂乱或要求重新设计的图形。



创建一个网格来指导布局工作可以保证一致性而不会牺牲统一。





**基于网格线的布局**（grid-based layout）是来自平面设计的一种技术，是一种对网页也同样有效的技术。这个方法通过使用“母版”来确保设计的一致性，各种布局都是根据这个模板来创建的。不是每一个布局都要使用网格的每一个部分——事实上，大多数的布局可能只会用到很少的一部分——但是每一个元素在网格上的位置应该统一和一致的。不管怎样，由于网页浏览器不能提供对文本元素尺寸的完全控制，因此把网格应用到网页设计上不会像平面设计一样简单。

你很容易就掉到“坚持使用网格系统”的漩涡之中——或者任何一种可以保证一致的标准——甚至在它们根本就行不通的时候。在没有设计标准的情况下进行工作是不好的，但是教条地遵守设计标准，而不顾适当的需求就可能会更糟。也许网站采取了一个在网格被开发出来的时候没有人能想像出来的新功能；也许这个网格从一开始就不怎么行得通。不管原因是什么，重要的是，能够认识到什么时候应该重新考虑你的设计系统的基础。

## 内部和外部的 consistency

由于网站被生产出来的方法——通常是在企业内部的、其他进行中的设计中，被逐渐、临时、独立生产的——它们的视觉设计的一致性已经被折腾得千疮百孔了，它们的毛病会一般有以下两种形式：



- ▶ 首先，内部一致性的问题。这是说，在网站的两个不同的地方反映了不同的设计方法。
- ▶ 然后，还有外部一致性的问题。这是说，这个网站没有在同一企业的其他产品中，反映出被使用的、相同的设计方法。

解决“内部一致性”问题的好办法，是基于对网站框架的深刻理解上的。确定有可能在网站的各种各样的界面、导航和信息设计等不同环境中反复出现的设计元素。在进行设计之前，从这些不同的环境中移除每一个设计元素，通过这个方法，我们就可以更清楚地看到我们应该解决的小规模的问题，而不是被环境所造成的大规模问题分散了注意力。与其一次又一次地设计同样的元素，我们不如试着独立地设计一次，然后将这个设计方案应用到整个网站中去。

很显然，对于这样的一种工作方法，我们还是必须要检查这些设计元素在不同环境中的呈现。可能一个较大的、圆形的、红色的“停止”按钮在结账页面上的表现不错，但是为它放到一个拥挤的产品定制页面的时候，效果可能就不那么理想了。最好的办法是把每一个元素都独立出来，设计好它们，在不同的环境中应用它们，然后在需要的时候再调整。

即使大多数的设计元素被相互独立地设计出来，它们最终也还是要放到一起的。一个成功的设计不仅仅是收集小巧的、精心设计的东西；相反，这些东西应该能形成一个系统，作为一个有凝聚力、连贯的整体来使用。





在过去，内部一致性的问题源头，往往是由于大部分网站是被IT部门设计出来的。一般来讲，这样的部门对于企业的视觉标准所知甚少（甚至根本就不知道）。在早些年，这种新媒体的设计隔阂只会被极深的技术局限性所强化。市场部门习惯了在平面印刷和广播媒体中几乎没有限制的设计，基本上把网站设计看成一种很简陋的媒介，从而不值得他们投入资源而放弃了。IT部门只好自行处理网站的事务，学着接受这种新媒体的局限，并逐渐形成了自己的设计标准。

无论如何，现在大部分的市场部门对企业网站的外观都拥有更多的直接控制权。同时，先进的技术和更精密的设计技术已经使网站看上去比以往更像他们所熟悉的平面印刷和广播了。因此，网站呈现出一种截然不同的外观，与他们自己在其他媒体上使用的元素完全不同，并且越来越少见。

你仍然不能期待网页设计能做到你在其他媒体上所能做到的一切。尽管如此，你还是可以从任何一个“能确保你线上和线下风格一致”的地方获得许多有价值的东西。这并不意味着这种风格必须要完全一样——相反地是，只要能产生同样的效果，它们就应该被设计出来。



用一个“统一的品牌识别形象”强化呈现在你的用户——客户、潜在客户、管理层、员工或其他访问者——面前的产品的“跨媒体的一致性”，这种品牌识别的一致性出现在你的网站的每一个级别的设计中，从每一页都会出现的导航元素到只出现一次的普通按钮。

出现一个在其他媒体上有的、与你的网站不统一的样式，影响的不仅仅是访问者的印象；它还影响了他们对整个企业的印象。人们对那些有具有明确定义的企业会有积极的反应。不统一的视觉样式会破坏企业形象的清晰程度，并且留给访问者一个“企业还没搞清楚自身定位”的坏印象。

## 配色方案和排版

色彩可能是向外界传递品牌识别的一个最有效的方法。一些品牌与色彩具有如此密切的关系，很难相信如果没有色彩，这些企业如何才能被不假思索地记住——想想可口可乐、UPS或柯达。这些企业多年来一直坚持使用了同一种很特别的颜色（红、棕、黄），这在公众的脑海中创造了一个很强烈的感觉。

这并不是说他们使用这些色彩来排斥其他所有的颜色。核心的品牌色彩通常是一个更广泛的配色方案（color palette）的一部分，这套配色方案是要在一个企业的所有材料中得到应用的。一个企业的标准配色方案中所使用的色彩，是为了它们在一起工作而专门挑选出来的，它们之间是互补而不冲突的。





一套配色方案应该能整合其中的色彩，以便能将它们应用到一个广泛的范围之中。在大多数情况下，更亮或更醒目的色彩可以用于设计你的前景色——那些你希望得到更多注意的元素中。更暗淡的色彩最好用于那些不需要跳出页面的背景元素中。拥有多种可选择的色彩，为我们提供了一套能做出高效的、设计选择的工具包。

正如“对比”和“一致性”对于视觉设计的其他领域很重要一样，它们在创建的配色方案也扮演了一个重要的角色。在同样的环境中使用时，一个非常接近其他颜色、但又不完全一样的色彩，会破坏你的配色方案的效率。这并不意味着你只能使用一种红色、蓝色，等等。这是说如果你想使用不同色度的红色，那么要保证它们之间的差异足够用户把它们区别开，同时也用了一致的方式来应用它们。

对于一些企业来说，排版（typography）——对于创建一个特殊的视觉样式而言，如何使用字体或字型——对他们的品牌识别是如此重要，他们已经发明了特殊的字型来专门供自己使用。一些企业，从苹果电脑到大众汽车，再到伦敦地铁已经使用了专门定制的字型，在他们的品牌传达中创建一个强烈的印象。但即使你选择不采取这个特殊的步骤，字体仍然可以作为“用设计来有效传达形象”的一部分而被利用。



就像许多企业一样，苹果电脑使用一致的字型——不管是在网站还是在其他媒体中——来传达一个统一的品牌形象。



The advertisement for the iMac G4 features a navigation bar at the top with links: Apple logo, Store, Switch, iMac, QuickTime, Support, and Mac OS X. Below these are sub-links: Hot News, Hardware, Software, Made4iMac, Education, Creative, SmallBiz, Developer, and Where to Buy.

**iMac.**  
How millions of new users get on the Internet — in minutes.

With all the built-in hardware and software you need to get online, the iMac is an expert at connecting to the Internet — so you don't have to be.

This iMac gives you the same core functionality and ease of use that enabled millions of iMac owners to enjoy their first experience of the Internet. Only now, the entry-level iMac has more than twice the performance of the original iMac, offers four times the hard disk capacity, four times as much RAM — and costs \$500 less.

**What makes the iMac so special?**

		
<b>Tech Specs.</b> All you want to know about CPU, memory, hard disk, ports and more.	<b>Software.</b> Every iMac includes a suite of useful, powerful and fun applications.	<b>Internet.</b> Legendary for its ease of use, the iMac gets you on the Internet in minutes.

The iMac features a built-in 15-inch shadow-mask CRT display with three crisp, super-sharp screen resolutions — perfect for editing text, viewing graphics and playing games — and comes with 128MB of RAM. Plus 256K of Level 2 cache running at full processor speed, ATI RAGE 128 Ultra graphics accelerator with 16MB of dedicated SDRAM, built-in microphone, dual mini-headphone jacks, dual FireWire ports and USB ports, a 10/100BASE-T Ethernet port, 56K modem, and a slot for an AirPort Card. The hard disk is a

**Buy Now**  
The Apple Store offers same-day online ordering 24 hours a day, every day.

**You can't beat the system**  
With the advanced, yet easy-to-use Mac OS X preinstalled on your iMac, you'll work and play with the most rock-solid — and reliable — OS technologies on the planet.

**A million fans on the outside, none on the inside**  
The one thing you won't hear is a fan. There isn't one. The iMac takes advantage of a unique design feature to cool itself quietly.



The 108-key Apple Pro Keyboard has a built-in two-port USB hub and special keys that let you adjust audio volume, mute the speakers and even eject a CD from the slot-loading drive. The smooth-gliding Apple Pro Mouse uses a high-precision optical sensor for unparalleled accuracy.





由于电脑屏幕的分辨率是有限的，一些在纸面上完全可以使用的字体在网站上会变得难以阅读。出于这个原因，一些设计得很清晰并在屏幕上非常容易阅读的字体（比如微软的Georgia和Verdana字体）已经成为热门的选择，并逐渐取代了类似Arial或Times New Roman之类的“默认”字体。

对于更大的文本元素或者类似在导航元素中看到的短标签一样的文本，稍微具有个性化的字体是非常恰当的。但我们的目标之一，不是用混乱的视觉、或不必要的各种字体——甚至通过不一致的方式来使用一小部分字体——来吓跑我们的用户，而是要对那种凌乱的感觉有所贡献。在大多数情况下，你不需要用太多字体来满足你的沟通需求。

有效地使用字体的原则与那些视觉设计的其他原则完全一样：不要使用非常相似但又不完全一样的风格。只有在你需要传达不同的信息时才使用不同的风格。风格之间要有足够的“对比”才能在你需要的时候吸引用户的注意，但是不要使用过于广泛和多样的风格。



## 设计合成品和风格指南

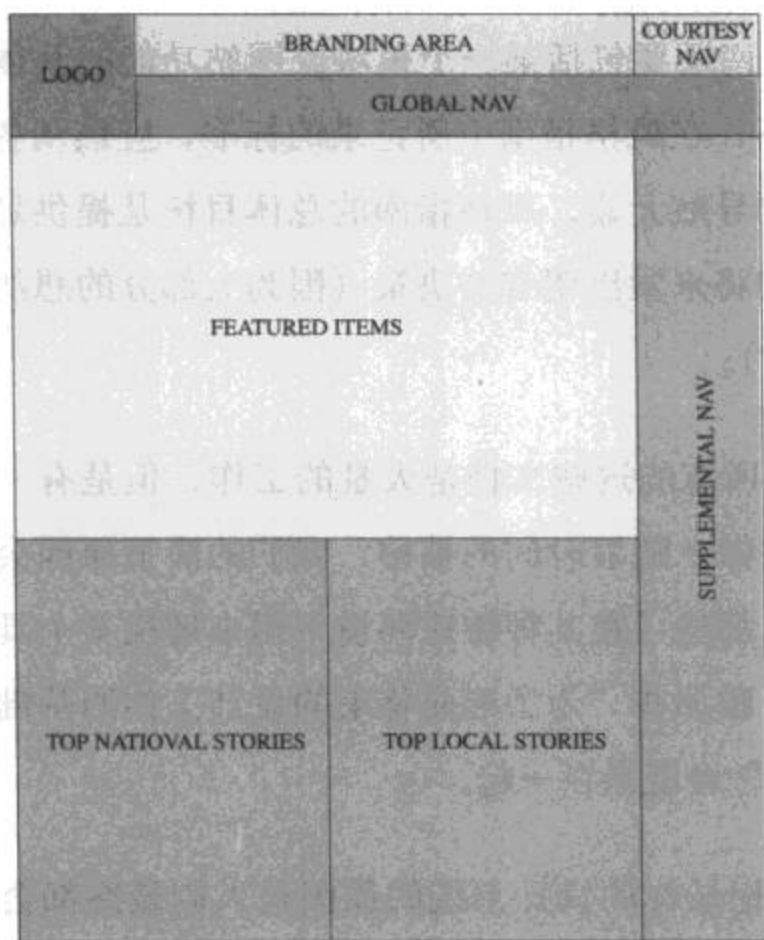
在视觉设计领域中对线框图最直接的模拟物是视觉模型或设计合成品 (design comp)。“合成”的意思是“综合的”，因为确切地说，它就是从已选定的组件中建立起来的、一个最终的可视化产品。这种合成物显示了各个独立的组件是如何结合到一起而形成一个有机的整体的；或者，如果它们没有组成一个整体，就说明某个地方破坏了它，同时也表明这是一个任何解决办法都必须要考虑到的约束条件。

你应该能看到在线框图的组件和设计合成品的组件之间的一个简单的一对一的相互关系。这个合成品不一定忠实地再现线框图的布局——在大多数情况下都没有。线框图没有说明视觉设计的关注点，而是侧重于记录框架层。在我们处理设计合成品之前建立起线框图，使我们首先能独立地了解框架层的问题，然后再关心表现层的问题。尽管如此，线框图各方面的概念，尤其是信息设计方面，应该显著地呈现在设计合成品中，即使它们没有精确地按照在线框图中出现的样子来进行组合。

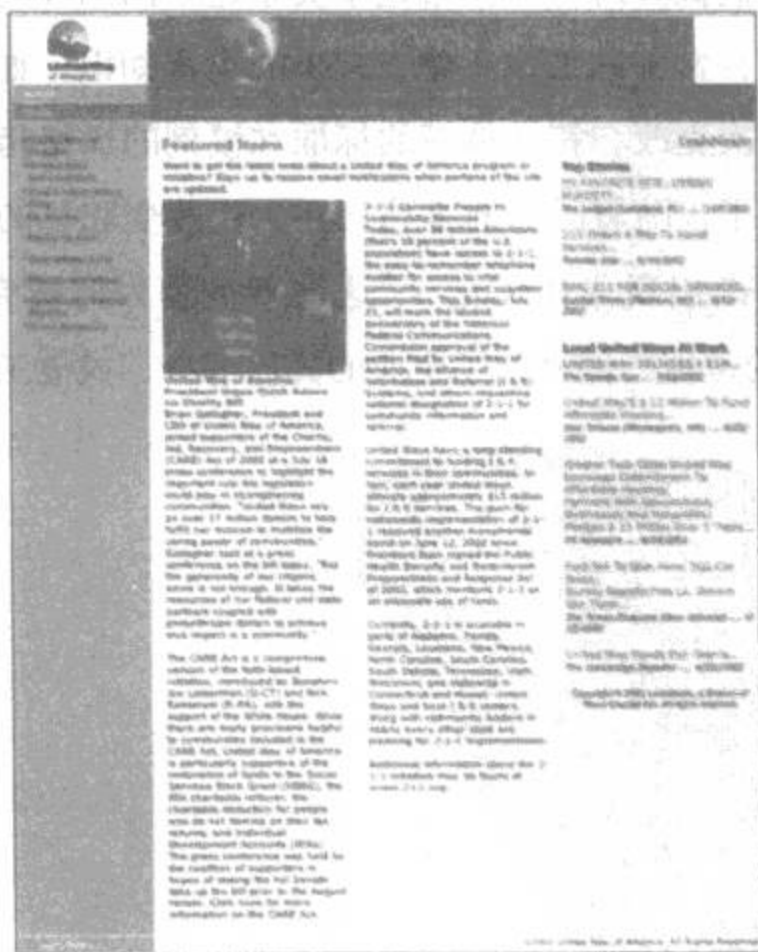
我们做出的所有设计决策的权威性文档是**风格指南** (style guide)。这个汇总文档确定了视觉设计的每个方面，从最大到最小的范围内的所有元素。影响到网站的每一个页面的全局标准——比如设计网格、配色方案、字体标准或标志应用指南——通常是风格指南中的第一部分。







视觉设计不一定要精确地按照线框图来做——只要它考虑到了相关的重要级别以及线框图中各元素的组合关系。



风格指南还要包括某一个模块或网站功能的具体标准。在某些情况下，在风格指南中所记录的标准，应该从各个独立界面到统一的导航元素。风格指南的总体目标是提供足够的细节来帮助人们将来做出明智的决策（因为大部分的想法都已经为他们实现了）。

当然，所有的这些文档是大量的工作，但是有一个很好的理由值得去做：随着时间的推移，我们的决策原因会逐渐从记忆中消失。那些“在某种特殊环境下用来解决某个具体问题的临时决策”和那些“为了形成将来的设计工作的基础而有意识做出的决策”会混杂在一起。

另一个记录你的设计系统的原因是人们最终都会离开这个工作。当他们离开时，他们带走了关于这个网站如何设计出来的、如何在日常基础工作上建立起来的丰富知识。如果没有一个保留每一次更改、符合最新的标准和惯例的风格指南，这些知识就丢失了。随着时间的推移，在人们改变职位的同时，整个企业会逐渐出现集体失忆，“这些事情是如何被完成”，和“这些决策的理由”偏移到了企业的其他部分，或被工作人员给丢弃了。





创建一个风格指南同样有助于在一个分散的企业中实施设计的一致性。如果网站运营包括各种各样正在执行中的独立项目，并且由分布在世界各地的人员来完成的话，那么你的网站很可能会像一个“风格和标准随机混杂的产品”。让所有的这些人遵循一套统一的标准来运作，需要做大量的工作，这就是为什么负责执行风格指南的团队在企业中的级别往往比你所预料的高的原因。一个风格指南，设计时要考虑所有这些不同的项目的需求，这是一个大的工程，但并非不可能。同时风格指南也是让你的网站看起来像是一个协调一致的整体，而不是一堆乱七八糟的碎片的唯一最有效的办法。

## 推荐阅读

Mullet, Kevin and Darrell Sano. *Designing Visual Interfaces: Communication Oriented Techniques*. Prentice Hall, 1994.

Williams, Robin. *The Non-Designer's Design Book*. Peachpit, 1994.

Web resources: [www.jjg.net/elements/resources/](http://www.jjg.net/elements/resources/)





# 第 8 章

## 要素的应用



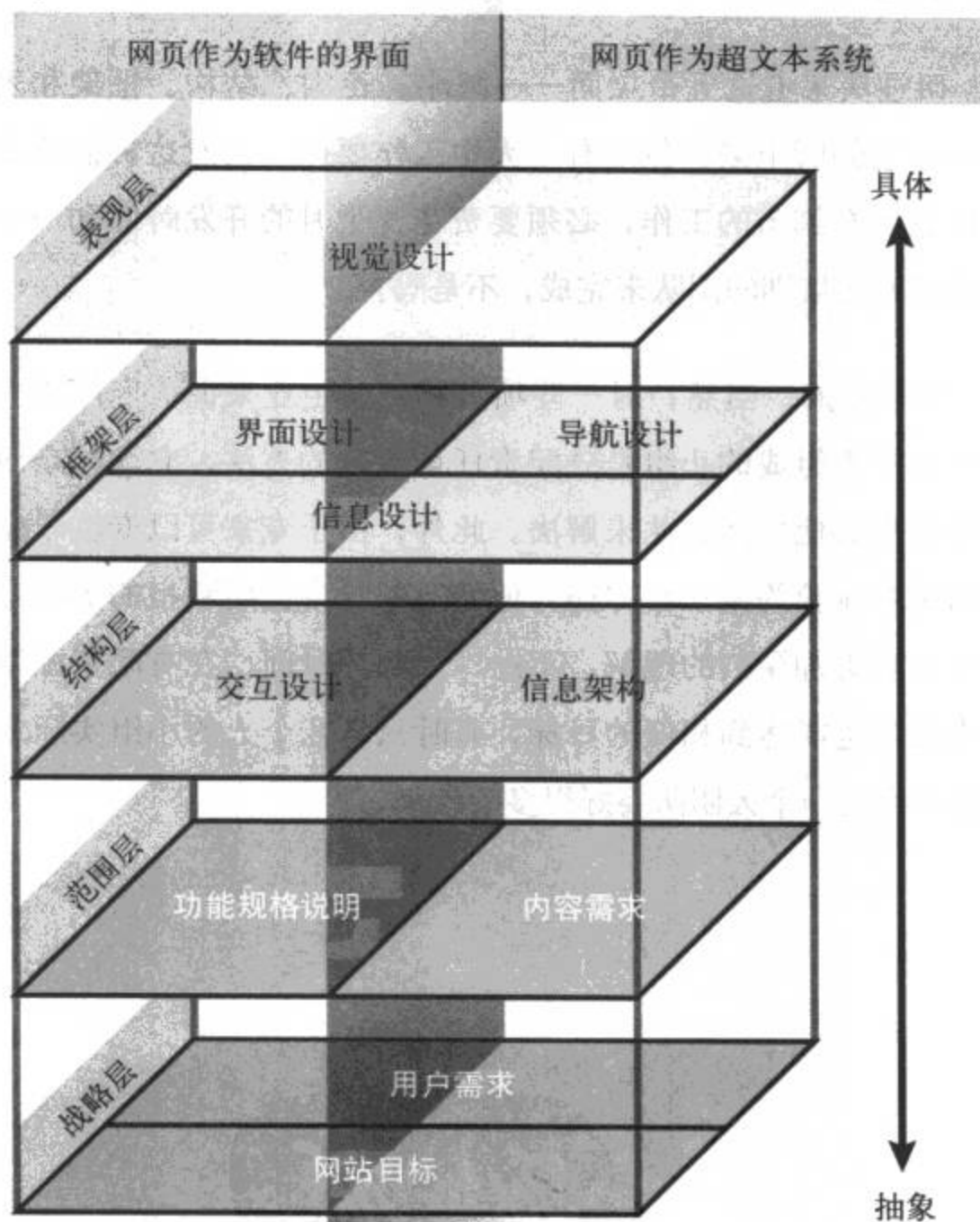


不管你的网站有多大，用户体验的要素都是一样的。但是，将这些要素背后的想法付诸实施看起来像是一次自我挑战。这不仅仅是时间和资源的问题——它常常是一个心态的问题。

回过头来看这五个层面——战略、范围、结构、框架和表现——它们听上去都像是有一大堆工作要做。当然这种需要高度注意所有细节的工作，必须要费花几个月的开发时间和一小组经过专业培训的团队来完成，不是吗？

不一定的。当然，对一些项目和一些企业来说，专门成立一个由专家组成的小组是分配责任最有效的方法，它太复杂而很难采用其他任何方法来解决。此外，由于专家可以专注于完整的用户体验的某一个方面，他们通常对自己所承担的那部分工作有着更加深刻的理解。不过，大部分时间，在有限资源下的小团队也能达到相似的目标。有时只有几个人的小组实际上可以产生比一个大团队要好得多的结果。







创建用户体验其实就是大量收集亟待解决的非常细微的问题。“成功的方法”和“注定会失败的方法”的差异归根结底就是以下两点：

- ▶ **了解你正在试着去解决的问题。**你已经知道在主页的那个紫色的大按钮是个问题，是按钮太大还是紫色不合适？它们哪个需要改变（表现层）？是这个按钮在这个页面上放置的地方不对（框架层）？还是这个按钮所代表的功能并不是按照用户所期望的那样在工作（结构层）？
- ▶ **了解你的解决办法所造成的后果。**要记住你所做出的每一个决定对其上、其下层面都有可能会产生“连锁反应”。在你的网站的某个部分运作得非常好的导航设计，可能完全不符合结构层的另一个部分。产品选择向导的交互设计也许是一种创新的方法，但它是否能满足“技术恐惧症用户”的需求呢？



创建你的网站的用户体验，这个任务看上去可能是一件显然很艰苦的工作。但你一定会惊讶于有多少组成用户设计开发过程的细微决定完全是不自觉而产生的。大部分时候，关于用户体验的决策总会体现在以下这些场景之中：

- ▶ **由现状决定的设计 (design by default)**。这发生在当用户体验的结构遵循其背后的技术，或你的企业的结构时。将客户的定单记录和消费信息分别保存在独立的数据库中，也许对于你现有的技术系统是合理的，但是这并不意味着把它们从用户体验中分开也同样是一个好主意。相似地，来自企业内部不同部门的内容，也只有在放到一起而不是保持独立的时候才会更好地服务于用户。
- ▶ **由模仿决定的设计 (design by mimicry)**。这发生在当用户体验依靠于来自其他网站、公共刊物或软件应用程序的相似情况时，不管这些情况对你的用户（或甚至对网站）来讲是否确实合适。最常见的例子是，在上个世纪90年代晚期，某个网站偶然把标签作为全局导航的一种设计，现在已经成为一个普遍的现象。
- ▶ **由领导决定的设计 (design by fiat)**。这发生在当用户体验由个人喜好来决定而不是由用户需求或网站目标来驱动的时候。如果由于某个高级副总裁很喜欢橙色，配色方案中橙色就占了主导地位，或者由于你的研发工程师的领导喜欢下拉菜单，结果导航元素就变成了下拉菜单的话，你就已经忽略了应该用于驱动你所做出的决策的战略目标。





## 实例：搜索引擎的实施

搜索引擎也许是网页中任何一个地方都会用到的常见功能——每个人都不止一次地使用过它们，同时没有提供搜索引擎的网站越来越少了。但即使它们似乎哪儿都有，这些看似简单的工具要想成功地实现出来也需要一系列横跨用户体验的要素的复杂决策。

现在的网站几乎到处都放置搜索引擎，这个现象反映了设计者越来越能理解对于“补偿符合特殊标准的内容的能力”是一个几乎所有用户都需要的功能。不管具体的用户或内容是什么，处理这类用户需求是一个关键的**战略**（strategy）的决策。

这个网站的内容需求和功能规格表明这个搜索引擎应该提供给用户的一个**范围**（scope）。如果内容的元数据是网站的一个需求的话，那么搜索引擎就可以用这个来给用户提供一个“利用这些数据的加强的能力”，比如，允许他们可以只通过某个作者或公开发表的、某个特定时期来搜索文章。如果不希望搜索引擎找到另外一些元数据，那么功能规格说明就可以详细地表明用户只能执行哪些类型的搜索。





在讨论结构 (structure) 时, 这个搜索引擎就真正地开始形成了——交互设计和信息架构——被纳入考虑范围。搜索功能的交互设计决定了用户实际上是如何与这个搜索引擎一起工作的。也许这个搜索的功能太复杂, 在用户得到真正的结果之前必须要经过一个漫长的、有组织的过程; 另一方面, 也许你所需要的仅仅是在每一个页面都放置一个关键词的输入框。

同样, 如果搜索引擎知道网站的信息架构是一个整体的话, 它就可以提供给用户更多的功能, 比如在网站的某个特殊区域内提供限制条件的搜索, 或根据它们在信息架构中的位置自动排列搜索结果。此外, 搜索结果也将有一个自己的信息架构。是所有的结果出现在一个页面上, 还是被分到多个页面之中? 如果是后者, 用户如何才能从一个页面到另一个页面中去? 他能随意地在各页之间跳转, 还是必须要遵守一个严格的、连贯的路径?



这些结构性的决策所采取的形式是这个搜索引擎的框架 (skeleton) 所决定的。在安排好按钮、输入框和其他“允许用户向系统提交一个请求的界面元素”的时候, 交互设计就成形了。信息架构则反映在“允许用户通过搜索结果在网站内移动的导航元素的设计”中。自始至终, 信息设计显示的是用户如何提出一个请求, 以及允许他们能够通过快速浏览搜索结果来找到他们所需要的东西。





在表现（surface）层面，所有这些元素由搜索引擎的视觉设计结合到一起成为一个统一的整体。它给予界面和信息元素一致的外观，并且吸引用户注意到页面的某个他们最有可能感兴趣的（或他们最有可能错过的）区域。



## 提出正确的问题

面对那些创建用户体验需要解决的纠缠不清的小问题，有时会是一件令人气馁的事情。有时某一个问题的解决办法都会让你不得不重新思考你认为已经解决的其他问题。很多时候，你必须在不同的做法之间作出妥协并评估利弊以及进行取舍。当你夹在不得不做出此类决定的中间左右为难的时候，不管你是否采取了正确的做法，都很容易变得沮丧和疑惑。这个简单的事实是这样：正确的做法，正是在用户体验的任何一个层面都没法“靠运气就能找到”的那一个。有意识、有目的地做出每一个决定，并在执行时将每一个决定都建立在对其背后的议题的理解之上。

对于你即将面临的问题抱有一种正确的心态是最重要的。每一个用户体验的开发过程的其他方面都有可能导致调整，以适应时间、金钱和为你工作的人员。没有时间去收集你的用户数据？也许你能找到办法去看看已经拿到手的信息，比如客服的日志或反馈的邮件，去找到你的用户需求的一些感觉。负担不起租用可用性实验室的费用？那就找几个朋友、家人或同事来做一些非正式的测试。



你所能犯的最大的错误，就是以“节省项目时间或金钱”的名义对用户体验问题敷衍了事。在某些项目中，一些人会自作聪明地在这个过程的最末尾添加“用户体验评估”——在应该提出这些问题的时机已经过去很久以后。当发布日期确定后，你告诉自己“比赛开始以后不要顾虑太多”，这看上去好像是一个不错的主意，但这样最后很可能是得到的是一个满足所有技术需求却恰恰对你的用户毫无用处的网站。甚至更糟的是，通过在结束时附加的用户体验评估，最后你可能会发布一个明知道已经被损坏却没有机会（或多余的金钱）去修复的网站。

一些企业很喜欢这种做法，称之为“用户接受测试”，“接受”这个词在这里的意思非常明显——问题不是说他们是否会喜欢或是否会使用这个网站，而是他们是否能接受它。这种类型的测试往往发生在整个流程的最后，在那个时候无数的假设已经在没有经过任何检查的情况下进入形成用户体验的过程中了。想在用户测试中揭露出这些假设是极度困难的，因为它们藏在了界面和交互的外衣下面。

很多人提倡将用户测试作为确保良好的用户体验的一个主要手段。这种思路看上去是你应该做一些事，将它们摆到一些人的面前，来看看他们有多喜欢它，然后无论他们抱怨什么都将其修正。但是测试永远无法取代一个考虑周密的、准备充分的用户体验设计过程。





专注于用户体验中某个具体元素的问题，能帮助你收集来自用户的更多的相关信息。没有着眼于用户体验的要素的用户测试，很可能以提出错误的问题而告终，这相反又会导致你得到错误的答案。例如，在测试原型的时候，知道要在调查中列出哪些问题是展示出你的测试主题的关键，经验是“不要用不相关的内容来把事情搞得更混乱”。导航条的问题真的只是跟颜色有关吗？还是你的用户对所用词汇的回应？

你不能简单地依赖用户来阐明自己的需求。创建任何一种用户体验的挑战是“比用户自己更准确地去理解他们的需求”。测试可以帮助你了解用户的需求，但是它只是能达到同样的目的的许多工具之一。

## 马拉松和短跑

就如同不应该拿用户体验的任何一部分来碰运气一样，你也不应该靠运气来完成自己的开发过程。一个企业中永远处在紧急情况下的开发团队太多了。每一个项目都被看成是对某些被察觉到的危机的回应，同时，这样的结果就是，每一个项目在它刚刚开始的时候就已经落后于计划了。

当我向客户描述问题的时候，我常常使用一个比喻来形容用户体验开发过程：它是一场“马拉松”而不是“短跑”。了解你所参加的比赛类型才能用适合的方式去参赛。



短跑比赛是短距离的比赛。短跑运动员必须在发令枪响起的那一刻聚集起所有储备的能量——而且他们必须要在那几分钟内迸发出所有的能量。在离开起跑线的一瞬间，短跑运动员就必须尽可能快地跑起来，并且尽可能地保持这种奔跑速度直至到达终点线。

马拉松是长距离的比赛。马拉松运动员需要和短跑运动员一样的能量，但它们的使用方式是完全不同的。成功的马拉松取决于运动员如何有效地控制自己的步伐。在所有的其他因素相同的情况下，运动员知道“何时加速”以及“何时减速”才更有可能赢得比赛——或者甚至彻底结束这场比赛。

短跑的战略——从开始到结束都要尽可能快地奔跑——显然是在这样的比赛中唯一最明智的做法。看上去你应该可以进行一场马拉松比赛，把它当成一系列全速冲刺的组合——但是这种方法是行不通的。行不通的部分原因是因为人类身体的耐力极限。这里还有另外一个因素：为了适应这个极限，马拉松运动员需要持续地监控自己的表现，密切注意哪些可行哪些不可行，并且适时地调整自己的方式。

网站开发很少是短跑比赛。更常见的是，有时候你会向前推进，建立原型和产生想法，然后随着时间推移，你再返回来，测试你所建立起来的东西，看看各个组成部分如何结合在一起，并且为这个项目提炼出一个综合的画面。有些任务需要重点强调速度，另一些则要求一个更加深思熟虑的方法。优秀的马拉松运动员知道哪些是哪些——所以你也应该。





经过深思熟虑的设计决策，可能会在短期内花费一定的时间，但是它们将在一个更长时期中节省更多的时间。设计师和开发者总是在他们的工作进行到某个阶段时，才后悔没有提前关注战略、范围和结构。我曾经参与了不止一个的项目，在这些项目中的活动总是处在有可能被取消的威胁之下。图形或一段代码产生了实际网站的组成部分，而对于那些没有产生可视化成果的任务，有些人会变得不耐烦。这些任务通常会在进程落后或预算超支的时候成为第一个被砍掉的项目。

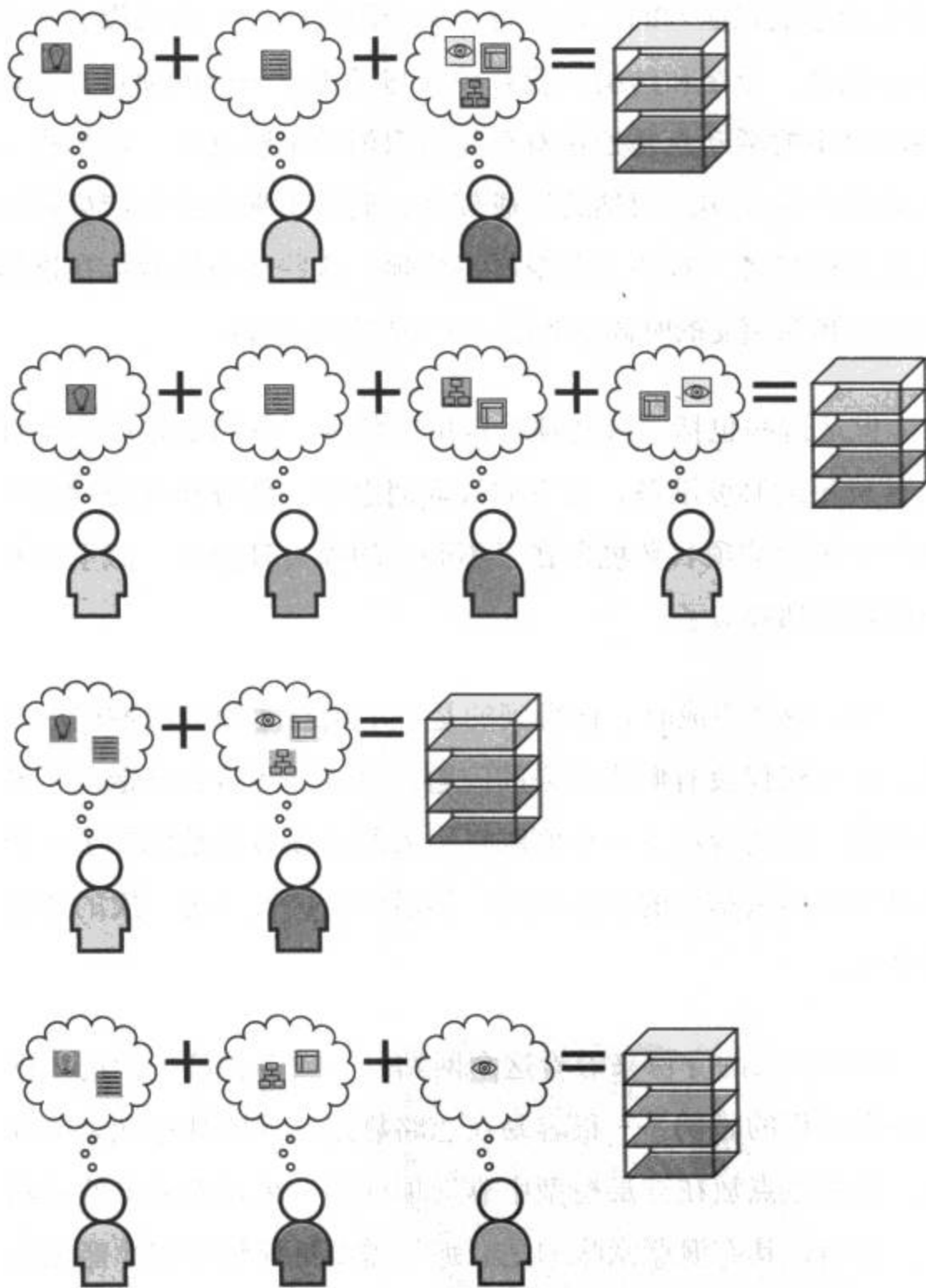
但是这些包括了最初项目范围的任务，它们是稍后产生可交付成果的必要准备。当它们被取消之后，任务和可交付成果被一个更大的项目环境留在了不确定的项目日程中，似乎与其他的那些都脱节了。

当你最终完成时，你得到的是一个没有达到大家期望的产品。你不仅仅没有解决原来的问题，事实上还给自己造成了新的问题，因为现在下一个刚刚冒出来的大项目正是你在上一个项目中试图去解决的一些缺陷。然后你又进入了另一次的恶性循环中。

以局外人的身份来看看这个网站——或者在第一次进入网站开发过程的时候——很容易在忽略靠近底部的那些元素的同时，把关注点放在五层模型中靠近顶部的、更显而易见的元素上。然而，具有讽刺意味的是，那些需要更仔细审查才能感知的元素——网站的战略层、范围层和结构层——在决定用户体验的最终成功或失败方面扮演了一个必不可少的角色。



让企业中的某个人去关注每一个层面的决策，只有这样你才能解决创建一个成功的用户体验所要考虑的全部关键因素。企业内部如何分配这些责任的问题，并不是和“确保所有用户体验的要素都已被列入考虑范围”一样重要。



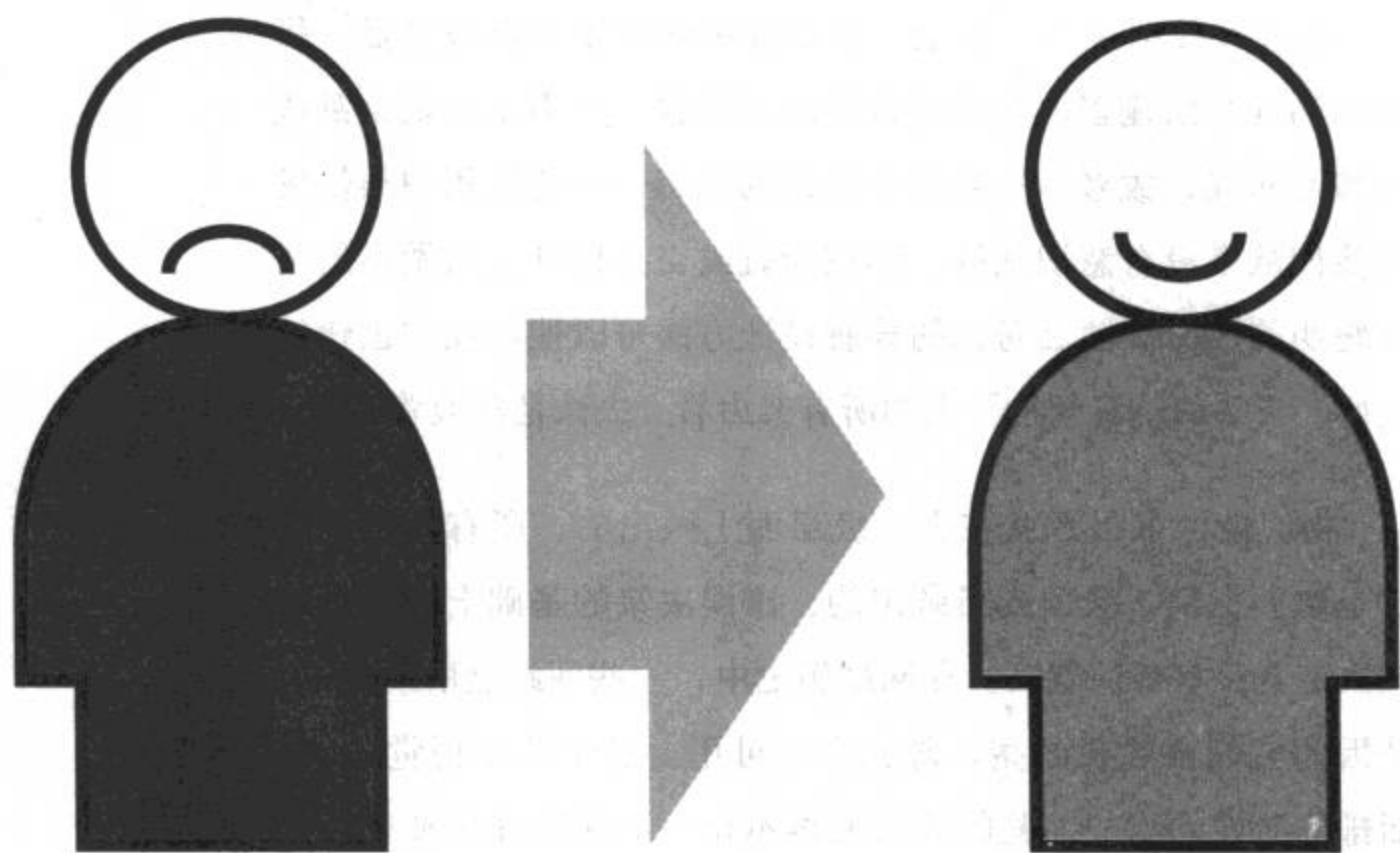


在大多数情况下，在上一级层面中的错误可以被更低一级层面的成功所掩盖。在视觉设计的问题——看上去很杂乱或混淆的布局，或者不一致或不协调的色彩——能让用户很快发现他们从来没有意识到的、你在导航或交互设计上所做出的所有聪明的选择。缺乏考虑的导航设计方法可以使你在“创建的良好、灵活的信息架构”上的所有努力看上去像是在浪费时间。

相似地，如果那些在上一级层面上做出的、所有正确的决定是建立在低一级层面所做出的、错误决策的基础上的话，那些决定就没有任何意义。在网站历史中，一些网站之所以失败，是因为它们虽然很漂亮，却完全不可用。过于关注视觉设计，而排除其他的用户体验的要素使得不止一个网站宣告破产，并使其他公司完全不明白为什么总是被网站问题所困扰。

这种糟糕的结果并不是必然的。如果你在网站开发的时候，始终从完整的用户体验出发，那么最后得到的网站就是一份有价值的资产，而不是无休止的债务。每一件与网站的用户体验有关的事情都是经过有意识地、明确地决策的结果，只有这样才能确保这个网站能同时满足你的战略目标和用户需求。







## 分享说明

如果你看到这段文字，那么通常来说你已经看完了本书，恭喜你！

马云曾说：“昨天很残酷，今天更残酷，明天很美好... 绝大部分人死在今天晚上。”也许坚持到明天，我们未必能成功。但是更多时候，是我们根本没有坚持！

在这个充满诱惑的时代，**我们花了多少时间在玩手机游戏、看电影、抑或是刷微博看小说上面，又花了多少时间静下心来读几本书？**你有多久没有完整地看完一本书了？更多时候是匆匆翻几页吧。让自己花更多时间系统化地阅读，更深入的思考，不让自己的生活被碎片化。

有时候看到一本好书，会心跳加速，觉得世间最大的享受莫过于此。我在 2011 年的第一天买了《三只鞋》送给自己，最开始是被美捷步创始人的商业传奇吸引买的，可是读到后来，发现自己深受启发的远非他的商业创奇，更多的是被他对生活的热情所鼓舞。当然，别指望看一本书就能改变很多，读书是一个过程。

## 关于我

非著名产品经理，产品观察家，痴迷交互设计、用户心理、产品体验、原型工具，给极客公园、虎嗅网、钛媒体、创业邦、i 黑马、互联网的一些事、搜狐新闻等码过几个字，扯扯产品设计。微信公众号：「idesign123」。信仰好设计，更有爱。

## 联系方式

以后有修订的版本，或者其他设计资料，会在微博@设计湿了没网站 和微信公众号（ID：[idesign123](#)）里通知，大家可以扫描下面的二维码在微信里和我交流，或者访问网站（[www.idesign123.cn](#)）了解更多互联网产品视觉、交互、用研及设计思想趋势和观点。



扫一扫或搜索：[idesign123](#)