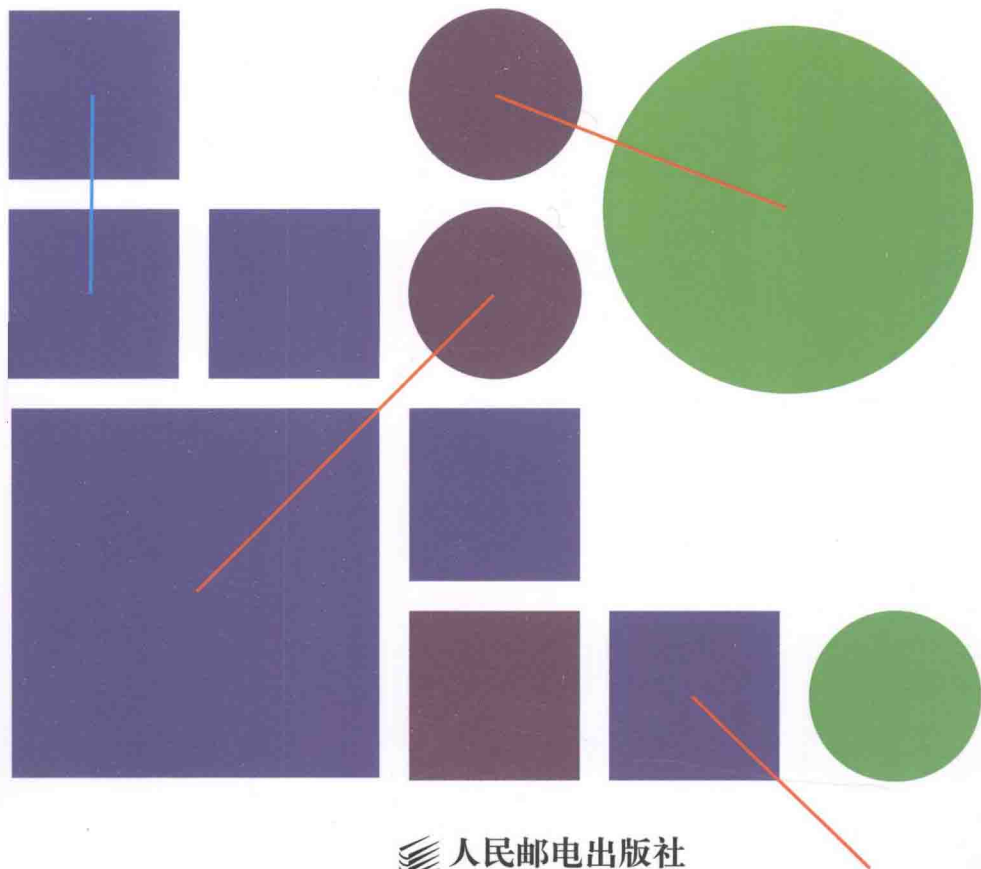


New
Riders

响应式Web设计 全流程解析

【美】Stephen Hay 著 余果 等 译

RESPONSIVE
DESIGN
WORKFLOW



人民邮电出版社
POSTS & TELECOM PRESS

响应式Web设计 全流程解析

RESPONSIVE DESIGN
WORKFLOW

【美】Stephen Hay 著 余果 等 译

人民邮电出版社
北京

图书在版编目(CIP)数据

响应式Web设计全流程解析 / (美) 海伊 (Hay, S.)
著; 余果等译. — 北京: 人民邮电出版社, 2014. 9
ISBN 978-7-115-36421-0

I. ①响… II. ①海… ②余… III. ①网页制作工具
IV. ①TP393.092

中国版本图书馆CIP数据核字(2014)第168978号

版 权 声 明

Responsive Design Workflow (ISBN: 9780321887863)

Stephen Hay

Copyright © 2013

Authorized translation from the English language edition published by New Riders.

All rights reserved.

本书中文简体字版由美国New Riders出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。
版权所有, 侵权必究。



-
- ◆ 著 [美] Stephen Hay
译 余 果 等
责任编辑 赵 轩
责任印制 彭志环 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京捷迅佳彩印刷有限公司印刷
- ◆ 开本: 720×960 1/16
印张: 14
字数: 190 千字 2014 年 9 月第 1 版
印数: 1—2 500 册 2014 年 9 月北京第 1 次印刷
著作权合同登记号 图字: 01-2013-5722 号
-

定价: 55.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

序

我不得不说，你正在读的这本书非常棒。

路德维希·维特根斯坦说过一句话，我很喜欢：“我的语言的界限就是我的世界的界限。”这个概念很神奇：这意味着你的词汇量越大，你的视野就越宽广。

我常常会想这句话，还有我的第一份工作。因为回忆过去，我才意识到我的 Web 设计入门遵循了一条非常狭隘的路，被四个词局限住：发现、设计、开发、发布。我被教导的是，每一个设计项目都被分为这四个独立的步骤。先是调研，然后设计阶段，然后就是编码实现，最后站点上线。简单、直观、线性开发。

开发流程有点像是接力比赛，团队成员必须完成自己的工作，然后其他人才能开始，一个一个传下去直到最终站点上线。当然，现实中有时候会出现一些意外和混乱。但是当我们不要仅为桌面设计站点，还要把我们的设计延伸到越来越多的设备上时，这种老旧的线性工作流程开始暴露出局限性。团队需要更多的合作。调研、设计和开发之间的联系比我们想象中更加紧密，而传统瀑布流的方法将它们割裂开了。

幸运的是，在本书中，Stephen 分享了他在这几年的思考，这是一种为 Web 设计和开发量身定做的响应式设计流程。他引导我们通过一个设计练习来了解新的线框模型，然后给客户展示响应式的设计，这让我们确信，这确实是更好的工作方式。

如果我们的世界的界限是由我们的语言限制的，那么 Stephen 的书就是一本名副其实的词典，它包含了新的原则和技术，它会颠覆你已有的思考方式，不光使你重新思考设计，还有 Web 的本质。

这本书会大大扩展你的视野，好好享受吧。

译者序

关于什么是正确的 **Web** 设计流程，这个问题没有统一的答案，探其原因，是因为 **Web** 设计是一门非常年轻的行业。

十几年前，**Web** 设计是由传统的平面设计师兼职的，在之后十几年变化中，这一职位渐渐分裂为 **Web** 设计相关的各个职位，包括用户研究、交互设计、视觉设计、前端开发。这些职位的分割带来了设计流程上的隔阂，每个人成为流水线上的一环，专注于输出自己的“交付物”。交互设计师输出线框图、流程图以及可交互的原型。视觉设计师参考交互设计师的交付物，输出精美的视觉设计稿，如果是响应式设计，需要输出多份设计稿。前端工程师根据视觉设计稿来输出静态页面。

这一现象带来的弊端有：很慢的开发流程，在输出产出物上无谓地浪费了时间；在针对需求变更的时候不能很好地响应变化；在移动浪潮来临的时代因为要输出多份交付物，流程显得更慢；大家的工作方式更像是传统软件公司的流程，而不是互联网公司的流程。

本书作者 **Stephen** 对这一现象提出了自己的疑问，并给出了自己亲身实践的一套流程方案。

Stephen 鼓励耦合度更高的设计流程：设计团队更加紧密地坐在一起，为同一个项目快速研发。

鼓励关注内容，关注基本内容的可用性，这样能确保在最糟糕的情况下，用户得到的网站也是可用的。

鼓励设计师身兼数职，不再区分交互设计师和视觉设计师，因为现在的交互线框图太具体明确，视觉设计师没有发挥才能的空间。

即使仍然保留交互设计师和视觉设计师的职位区分，也不再输出传统交互线框图，而是使用可迭代的交付物。交互设计师输出 HTML 页面，视觉设计师可以基于此来工作，避免时间浪费在交付物的输出上。

鼓励设计师学习一些自动化工具，比如简单的 HTML 和 CSS、自动截图脚本、静态服务器等，把自己从无聊的大量输出中解放出来。

目标读者

如果你是：

大公司或者小公司中的交互设计师、视觉设计师、前端工程师，并且你希望能改善现有的设计流程，更好地拥抱变化和移动互联网时代。

独立 Web 开发者，在针对客户的需求变更和响应式的需求时疲于修改，希望能改善流程。

希望能学习更多 Web 知识的学生。

那么这本书就是为你而写。

对于专注交互和视觉设计师来说，这一创新流程是不小的挑战，因为要跳出自己的舒适区，学习一些技术相关的知识。而对于已经习惯已有流程的同事，要推进新的流程在整个部门的落地势必会遇到很大的阻力。

虽然挑战很大，但随之而来的回报是颇为丰厚的，在移动互联网的时代，谁更快、更能拥抱变化、拥抱多终端设备，谁就能更成功。

余 果

腾讯 ISUX 高级前端设计师

2014 年 6 月 8 日

译者简介

余 果

腾讯高级前端设计师，负责本书的部分翻译和全书的校对工作。
有志成为全栈工程师，并积极地利用技术来优化用户体验。

周天牧

前腾讯产品经理，现创业。
信仰互联网精神，坚持学习，对产品、技术、设计和商业模式。

陈伟华

前腾讯前端设计师，现居新西兰，
是一位自由职业者，正在学习油画。

林钊仕

前腾讯前端工程师，现创业。
热爱 Web 事业，热爱分享。

黄雪莲 (snowdrop)

金立前端工程师，常年混迹于互联网。
专注技术，捣鼓翻译，热爱跑步，享受在个人局限中无限燃烧自己。

王琨琛

iFLYTEK 项目经理，热血青年。
喜欢摇滚、足球、机车、马拉松。

目录

第1章 拥抱变化 1

- 1.1 精美设计稿的诞生 2
- 1.2 静态设计稿舒适区 4
- 1.3 专家的入侵 5
- 1.4 我们都是交互设计师 8
- 1.5 跳出瀑布模型 8
- 1.6 压死骆驼的稻草 9
- 1.7 屋里的大象 10
- 1.8 这并非福音书 11
- 1.9 这是个挑战 11

第2章 从内容开始 13

- 2.1 微结构 vs. 模块化结构 14
- 2.2 懒人的内容清单 16
- 2.3 通用的例子：本书网站 16
- 2.4 渐进的设计原则：零界面 17
- 2.5 创建内容清单示例 18
- 2.6 试试看 21

第3章 内容参考线框图 23

- 3.1 别把线框图复杂化 25
- 3.2 第一步：创建低保真的、基于网络的设计稿 26

3.2.1	创建基础 HTML	27
3.2.2	形成基本样式	30
3.2.3	移动优先版线框图	33
3.2.4	添加导航	35
3.2.5	为大尺寸屏幕创造变量	37
3.3	打破神话	43
3.3.1	交互设计师应该做线框图	43
3.3.2	线框图应该详细	44
3.3.3	内容参考线框图是否限制了设计选择?	44
3.3.4	现在就考虑布局是不是有点太早了?	45
3.3.5	我应该做什么样的线框图?	45
3.3.6	我应何时让客户参与? (或者说“我的漂亮的交付品在哪儿?”)	46
3.4	动手试试	46

第 4 章 基于文本而设计

49

4.1	内容为王	50
4.2	标记纯文本	53
4.2.1	用 Markdown 来处理图书页面的内容	54
4.2.2	这个阶段修改的意义	56
4.2.3	思考很重要	58
4.3	将纯文本转换为 HTML	59
4.3.1	使用命令行	60
4.3.2	转化为 HTML	65

第 5 章 线性设计

67

5.1	开发一种设计语言	68
5.1.1	使用设计漏斗	70

5.1.2 在实际设备中运行设计	72
5.2 增强结构化内容	74
5.2.1 模板介绍	76
5.2.2 目前你的项目文件夹	79
5.2.3 思考和画草图	80
5.2.4 多尝试字体和颜色	81
5.2.5 暂时不要做太多	85

第6章 断点图

87

6.1 文档断点	89
6.1.1 剖析断点	90
6.1.2 可视化断点	92
6.1.3 断点图组件	93
6.2 创建简单的断点图	95
6.3 主次断点	97
6.4 添加更多东西	98
6.5 总结	102

第7章 为断点而设计

103

7.1 首先, 关于草图的一点点东西	104
7.1.1 如何素描	105
7.1.2 在设备上素描	109
7.1.3 养成素描的习惯	111
7.2 专注于主断点	112
7.3 素描的时候仔细思考内容	114
7.3.1 文本	114
7.3.2 导航	115

7.3.3 表格	116
7.4 当你没有灵感的时候怎么办	119

第 8 章 创建 Web 设计模型

121

8.1 跨越障碍	123
8.1.1 客户不关心	123
8.1.2 其他的人	124
8.1.3 你自己	125
8.1.4 展示你的模型	127
8.2 开始实践	128
8.3 从静态页面到静态网站生成器	133
8.3.1 模板	133
8.3.2 选择一个静态站点生成器	134
8.3.3 关于 DEXY	135
8.3.4 安装 DEXY	136
8.3.5 你积累的资源	140
8.3.6 添加样式	140
8.3.7 添加内容	142
8.3.8 给内容分组	142
8.3.9 DEXY 控制中心: dexy.yaml 文件	146
8.3.10 使用 CSS 完成 Web 设计模型	148
8.3.11 多个页面	148
8.4 总结	150

第9章 截屏

151

- 9.1 为什么不直接在网页上进行展示? 152
 - 9.1.1 演示 / 现实的平衡 153
 - 9.1.2 截屏: 从 Web 设计模型回到图像 154
- 9.2 如何截屏 157
 - 9.2.1 手动截屏 157
 - 9.2.2 自动截屏 159
- 9.3 展示截屏 164

第10章 成果展示: 浏览器体验

167

- 10.1 你会在你的设计中发现很多的 bug 168
- 10.2 沟通与协作 169
- 10.3 怎样去展示你的交互原型 171
 - 10.3.1 用设备来让你的设计更有说服力 171
 - 10.3.2 解释你的设计 172
- 10.4 测试和客户审阅 173
 - 10.4.1 客户审阅 175
 - 10.4.2 做好笔记 176
 - 10.4.3 使用笔记去修改设计 182

第11章 创建设计手册

185

- 11.1 网页设计规范 187
- 11.2 设计手册的内容与结构 189
- 11.3 我心仪的规范设计软件 191
- 11.4 创建设计文档 193
 - 11.4.1 动手写文档 194

11.4.2	添加各种形式的示例	196
11.4.3	生成截图	197
11.4.4	配置 Dexy	199
11.4.5	测试整个 Dexy 项目	200
11.4.6	捕获特定元素的屏幕截图	201
11.4.7	引入能渲染出效果的 HTML	204
11.4.8	包含高亮的代码	207
11.4.9	动手做一个你自己的文档	210
11.5	写在末尾的话	211

第 1 章

拥抱变化

互联网是个神奇的存在，是个不断变化，并且在创新中见证奇迹的地方。如果你记得最早的互联网长什么样，你会有更深刻的感触。1995年建立的网站和那时的“设计”，在现在来看可能更像一个笑话。

某种程度上说，现在网站的设计流程已截然不同，但反过来说，却也丝毫未变。

从互联网的伊始，设计师们就在绞尽脑汁将创意落实到浏览器里。最早火起来的互联网设计师是那些爱耍小聪明的家伙，用 **hacking** 技巧来帮助自己实现目标。从 **spacer GIF** 和表格布局（**layout tables**）到滑动门（**sliding doors**）、伪等高布局（**faux columns**）和图像置换，再到编程框架、**CSS** 预处理器和 **JavaScript polyfills**¹，他们将设计与网页融合的招儿想了个遍，即使需要把页面内容和无实际含义的表现元素相结合，甚至破坏语义化的 **Web**。

时光流逝，不论是上网设备、浏览器还是用户本身，都随整个互联网发生了巨变，但设计流程还基本保持着早期互联网时代的模样，更准确地说，与互联网产生之前一样。

1.1 精美设计稿的诞生

我以前在传统设计行业工作。那时候刚毕业几个月，我就去了荷兰，花几周时间打遍了周边所有设计公司的电话，通过了其中一家公司的面试并拿到了实习 **offer**。那是 1992 年，大部分荷兰人在 3 年后才接触到互联网。我所在的这家小公司有台电脑，我记得是台苹果的 **Macintosh LC**。它只能用来写信或者开发票，唯独不能做设计，而那时的设计稿都是用昂贵的彩色马克笔手工绘制的。

¹ **JavaScript polyfills** 指一些浏览器暂不支持的高级 **CSS** 或者 **HTML** 特性，使用 **JavaScript** 实现类似的功能，作用是让使用高级 **CSS** 和 **HTML** 功能的开发者不用纠结浏览器兼容性问题（译者注）。

那时候我们做设计纯靠手绘。就像在学校学过的那样，先画草稿，越多越好，选几幅好的，再画初稿。然后再从初稿里选 1 到 3 幅画出成品，因为广告公司都喜欢以量取胜，所以一般都是 3 幅。在那个年代，负责马克笔渲染的设计师是最牛的。印前工作都是整体或部分外包给专业公司的。而常和电脑打交道的我，很费解为什么电脑在行业里没有被充分利用，但这种情况很快就不一样了。

6 个月后我转正了。源于对电脑字体排版的钟爱，我探索出一条路：在公司的 Macintosh Quadra 700 电脑上（在我开始实习不久后买的）用 QuarkXPress 创建设计稿，打印出来，并在这个设计稿的基础上做马克笔渲染。这对我们的客户来说有更实际的好处——所见即所得。我认识的其他设计师也纷纷效仿，开始把彩印图片加入设计初稿而非手绘。没过多久我就开始用 Photoshop 和 QuarkXPress 在电脑上做完整的设计稿。这期间许多经验告诉我，用电脑做设计很省事儿，只需要在印出的设计稿上用马克笔画阴影就行了。于是我开始用 Photoshop 完成全部设计，因为这样效果看起来更真实。客户都爱 Photoshop 渲染，而那些曾经很厉害的马克笔渲染设计师，从此以后就接不到我们的活儿了。

我对这样的工作方式很满意，但当时却没有意识到电脑正在逐渐取代马克笔，我在创造产品而非仅仅实现视觉。不过事后来看，用电脑排版，批量复制样式精美的作品确实节省了时间，也成为了常规的工作方式。当我开始用 Photoshop 的时候，与其他纯手绘的公司比起来，作品质量确实鹤立鸡群。客户都希望看到他们将得到的是什么，于是我们每次都胜出。最终，大家都选择这么做。

有趣的讨论接踵而至。是否因为我们使用了新的工具就削弱了创意？以往大家的印象是，用马克笔在纸上作画是创意的延伸和实现，而电脑却从某些程度上僵化了这些创意。作为视觉设计的工具，电脑是不是让我们变得缺乏创意？抑或是一个更有效的实现已有创意的工具？整个行业在基础重

复性工作上投入大量时间值得么？真的有必要在售前阶段细化如此多的细节吗？

不论如何，事情自然发展着。客户都开始期待优质的精美的静态设计稿。我们的做法是，把打印好的设计稿贴在展板上展示给客户，告诉他们这就是方案。当时我们也没想到更好的办法展示，因为稿子用电脑打印总比马克笔手绘更接近现实。

1.2 静态设计稿舒适区

在我们做精美的静态设计稿几年后，发生了有趣的事情——互联网诞生了。你可能会期待，这个改变全世界的新事物能否也在设计领域催生什么变化，正如当年电脑帮助设计脱离了纯手工劳动那样。但什么都没发生。我们仍然用 **Photoshop** 合成漂亮的产品设计稿，而客户也认可，于是我们就这样继续了多年。

停下来想想，设计界随着电脑的出现产生了天翻地覆的变革。作品变得更真实，它们比以往任何时候都更准确地展示了最终产品的样子。为什么类似的变革没有随着互联网的出现而产生？

我个人的观点是，网上展示出来的东西看起来不够好。早些年里，网页字体不是抗锯齿的，而 **Photoshop** 设计稿却能使抗锯齿。所以它看起来更美观，因此也能卖得好。

这种总把网站设计稿处理成漂亮图片的办法带来了麻烦。最后我也厌倦了总是向客户解释，为什么网站上线后看起来总是比设计稿差很多。于是我放弃了抗锯齿化，并且要求我的员工也这么做，因为我不想让客户有这种不愉快的“惊喜”。这么多年来，我们使用图像编辑软件，比如 **Illustrator** 或者 **Photoshop** 时，一直尽可能坚持真实。与此同时，我们采取两个配套