
UNDERCOVER USER EXPERIENCE DESIGN



Learn how to do great UX work
with tiny budgets, no time,
and limited support

CENNYDD BOWLES AND JAMES BOX

UNDERCOVER USER EXPERIENCE



Learn how to do great UX work
with tiny budgets, no time,
and limited support

CENNYDD BOWLES AND JAMES BOX

Undercover User Experience Design

Cennydd Bowles and James Box

New Riders
1249 Eighth Street
Berkeley, CA 94710
510/524-2178
510/524-2221 (fax)

Find us on the Web at www.newriders.com
To report errors, please send a note to errata@peachpit.com

New Riders is an imprint of Peachpit, a division of Pearson Education
Copyright © 2011 by Cennydd Bowles and James Box

Editor: Wendy Sharp
Production Coordinator: Tracey Croom
Copyeditor: Jacqueline Aaron
Compositor: Danielle Foster
Indexer: Emily Glossbrenner, FireCrystal Communications
Cover design: Mimi Heft
Interior design: Kathleen Cunningham

Notice of Rights

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Notice of Liability

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of the book, neither the authors nor Peachpit Press shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN 10: 0-321-71990-5
ISBN 13: 978-0-321-71990-4

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

★ ENDORSEMENTS

“A wonderful, practical, yet subversive book. Cennydd and James teach you the subtle art of fighting for—and designing for—users in a hostile world.”

—Joshua Porter, co-founder Performable & co-creator of 52WeeksofUX.com

“Cennydd and James’s clever and crafty book will teach you how to make your own rules, play well with others, and create a culture of UX from the ground up.”

—Whitney Hess, user experience design consultant,
author of the blog *Pleasure and Pain*

“Making design matter in your organization is not about titles and talk. It’s about what you get done every day. *Undercover User Experience Design* will show you the way.”

—Luke Wroblewski, author of *Web Form Design & Site-Seeing*

“At Clearleft we pride ourselves on delivering exceptional design even with tight deadlines and budgets. This indispensable guide to guerrilla UX explains how we do it.”

—Andy Budd, co-founder and managing director of Clearleft

★ ACKNOWLEDGMENTS

Cennydd and James thank

Wendy Sharp, Jacqueline Aaron, Tracey Croom and all the staff at Peachpit, for helping a couple of novices to achieve something we still find unimaginable. Your dedication, patience and irreplaceable advice kept us going. We hope we've done justice to your efforts.

Our colleagues past and present at Clearleft—Paul Annett, Sophie Barrett, Andy Budd, Kate Bulpitt, Natalie Downe, Andy Hume, Jeremy Keith, Paul Lloyd, and Richard Rutter—for creating an environment in which we could flourish, and covering for us when deadlines got tight. Special thanks to James Bates for going beyond the call of duty with designerly assistance.

Whitney Hess, Luke Wroblewski and Josh Porter for their flattering endorsements, and Peter Boersma, Dan Brown, Leah Buley, Leisa Reichelt and Brandon Schauer for kindly agreeing to let their ideas be featured. You all exemplify the intelligence and generosity of this fabulous community.

Chris Summerlin (www.honeyisfunny.com) for his fantastic chapter illustrations, and for demonstrating superior knowledge of classic cars.

Francis Storr for his invaluable comments on the early drafts of this book, and helping us to see through our readers' eyes.

Cennydd thanks

The Brighton and London UX communities, for inspiring and tolerating me. Particular respect to Martin Belam, Desigan Chinniah, Danny Hope, Johanna Kollmann and Matthew Solle for their tireless undercover work for the British UX community. Thanks also to Ian Fenn for the sticky note tip, to Amanda Jahn and Sjors Timmer for their offers of support, and to my mentorship partners for asking the smart questions that gave birth to this book.

Friends—Sascha Auerbach, Alex Block, Tom Britten, Chloe Brooks, Isla Buchanan, Sharon Chesworth, Justin Clark, Rebecca Cottrell, Joe Cross, Pippa Dakin, Jo Denyer, The Sharp Foundry, Andy Fox, Kushal Gaya, Sélène Hinton, Kate Howson,

Emily Kawasaki, Sam Lewis, Tom Payne, Jos Roberts, Rejina Juie Sabur, Dan Waddy, Matt Wheeler, James Wilson, Craig Wood, James Woods and many more—for their support and forgiving my enforced social hibernation.

My family, for—well—pretty much everything.

Anna, for her support and compassion. Words aren't enough.

My co-author James. Heartfelt thanks and eternal respect. Although I still prefer the provisional title for the book.

James thanks

The UX community, especially my allies in Brighton and the rest of the British Isles. Thank you for inspiring me. Keep sharing.

All my friends. For remembering me during my somewhat prolonged absence, but more importantly, for reminding me “It’s just a book.”

My family. For their unerring support and encouragement. I hereby promise you cake.

Sarah, Jemima & Charles. Without your extraordinary patience and understanding none of this would be possible. I love you.

And finally, Cennydd. An incredible talent and someone I feel honored to call my co-author and friend.

TABLE OF CONTENTS

CHAPTER 1	Going Undercover	1
	The Undercover Manifesto	3
CHAPTER 2	Exploring the Problem	5
	Hit the Ground Running	6
	Defining the Problem	13
	Culture	19
	User Research Under the Radar	23
	Research Outputs	33
CHAPTER 3	Generating Ideas	41
	Sketching	43
	Collaborative Design	51
	Sharing the Spark	62
	Design Principles	64
CHAPTER 4	Making It Real	69
	Design Deliverables	71
	Fidelity	81
	Better Deliverables	90
CHAPTER 5	Refining Your Solution	97
	The Nature of Iteration	98
	Testing with Users	99
	Critique	113
	After Launch	123

CHAPTER 6 Working with.131

Working with Developers.	132
Working with Visual Designers	136
Working with Content Specialists.	139
Working with Product Owners	141
Working with Marketers	144
Working with SEO Specialists	146
Working with Senior Managers	150
Working with Agile	153

CHAPTER 7 Where Next? 159

UX Leadership	168
Moving On	172
Over to You.	176

Index.	178
----------------	-----

This page intentionally left blank

MAKING IT REAL



However deeply you have investigated the problem and however many creative ideas you have produced, you've little to show for your time until you turn your work into something concrete.

It's time to make things real. Where objectives, requirements, and unrequirements previously took a backseat to exploration, you should now look carefully at which of your ideas meet both business and user needs.

It's natural to have a hunch about which ideas you want to pursue. Now you should combine these diverse concepts into a more definite form. You're moving toward a complete design by adding detail, but there's still room to change your mind and tweak your designs as you go.

THE ROLE OF DELIVERABLES

We don't like jargon, but sometimes we're stuck with it. The word *deliverable* has become part of the UX vocabulary—a way to describe the outputs that UX designers share with stakeholders.

Whether these outputs are personas, sitemaps, or wireframes, creating deliverables is a craft that demands practice and expertise in the tools of the trade. It's not surprising, then, that user experience designers are sometimes guilty of “fetishizing” deliverables, seeing their creation as the *raison d'être* of great UX design. To an extent, this attitude is understandable since designers are often judged on the quality of their deliverables. There's nothing wrong with aiming high, but remember that in our manifesto we pledged allegiance to delivery. Deliverables are a stop on the journey, not the end of the line.

The most important role of deliverables is to document your design choices. Deliverables play an important role in the success of the project, helping you to communicate key concepts and the project's direction. They also reduce project risk by recording decisions made throughout the process.

But deliverables are more than mere documentation. They're also useful design tools. Just as a visual designer uses graphics software to work up an idea into a full page mockup, the UX designer needs a medium for refining ideas. Particularly in the early stages of project, the act of creating the deliverable can be as valuable as the deliverable itself.

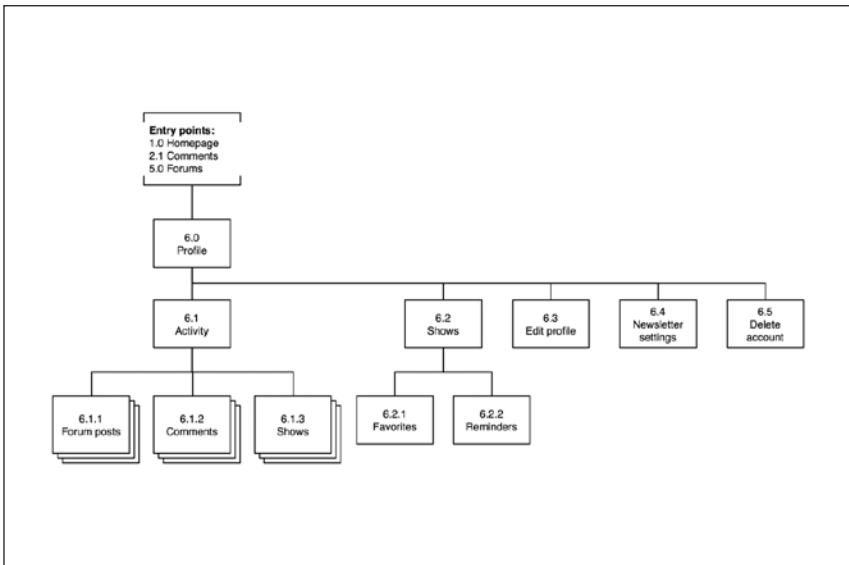
Wireframes, personas, sitemaps, and other deliverables are therefore living documents, steadily getting closer to the final design with time. Some organizations struggle with this suspended uncertainty, believing that if a design is on paper it's final. If you work for this kind of company, be careful not to let people fixate on early designs.

★ DESIGN DELIVERABLES

In Chapter 2 we discussed deliverables that capture research findings. Now we'll focus on deliverables that you can use to document your ongoing design work.

SITEMAP

A sitemap is a visual representation of a site's structure **A**. Usually arranged hierarchically, sitemaps indicate how content and information are organized and, consequently, how users will navigate the system. A sitemap documents the system as a whole, pulling back from interface specifics to look from a broader vantage point.



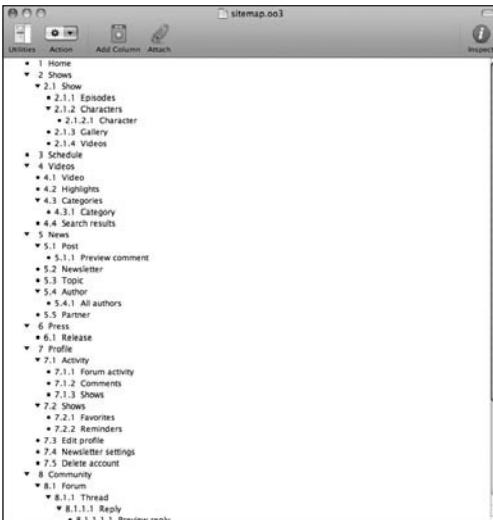
A This sitemap uses a top-to-bottom hierarchy to show site structure.

A sitemap is a familiar tool to every UX designer; you'll use one for most sites you design, particularly large, information-heavy sites. Use what you learned in your research (card sorts, interviews, and so on) to draft a structure that holds the site's content in a way that's meaningful to your users. It's worth starting this process as early as you can, so that you can fully appreciate the components of the system and what work lies ahead. It's not unusual to discover branches reaching farther than you expected.

Creating a sitemap will help you understand not only scope but also how sections and pages should be arranged and labeled. By the end of the exercise you should have a fair idea of how your site navigation will work, and what labels you should give to site sections.

Documenting a sitemap is a relatively simple task, thanks to Jesse James Garrett's visual vocabulary (www.jjg.net/ia/visvocab), in which you'll find a standardized set of shapes and connectors for use in your sitemaps and other deliverables. But don't let a sitemap's simplicity fool you: it will play an important role in the project. It might be the first deliverable you share with your team, so don't be surprised if it sparks debate. We'll talk more about feedback and critique in Chapter 5.

Since a sitemap is all about structure and hierarchy, starting with a text outline **B** can be useful. Tools like Microsoft Excel and the Omni Group's OmniOutliner will help you shuffle pages around before you start creating all those boxes and arrows.




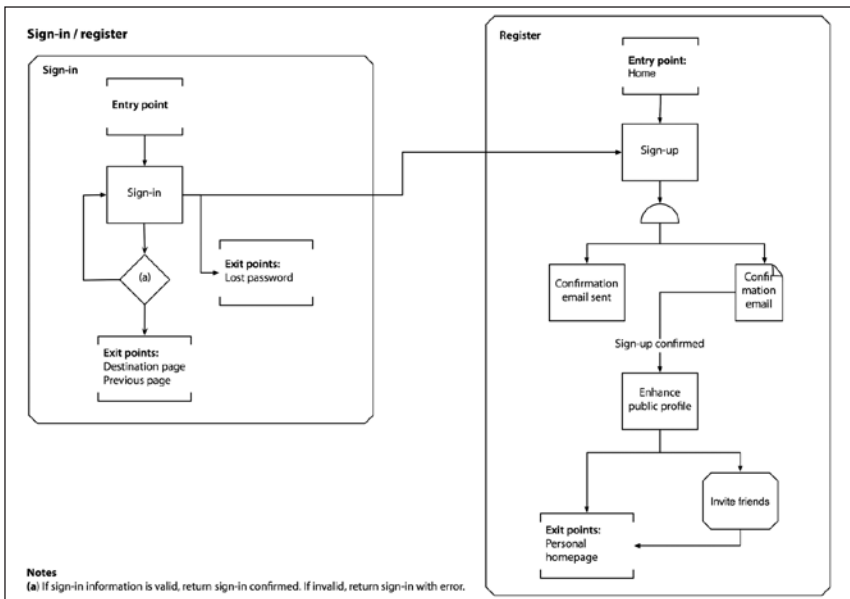
B Text outlines help you plan the site's structure before you translate it into a visual sitemap.

Sitemaps get big quickly, so make your drawing canvas as large as possible. Don't worry too much about laying the structure out beautifully; your sitemap will change a great deal over the course of the project. However, do try to use your software's connection tools properly, so that when you move a section, its connections move with it.

If your sitemap is becoming unwieldy, use the visual vocabulary's continuation-point symbol to break the sitemap into smaller subsections. Finally, add a simple numerical identifier to each page you depict. This will help you cross-reference pages with other deliverables.

USER FLOW

A user flow  shows the steps a user can take to complete a task or activity.



 This user flow captures a sign-in/register process using Jesse James Garrett's visual vocabulary.

The simple hierarchical nature of a sitemap can be too limiting on sites with complex navigation, large amounts of user-generated content, or process-heavy applications that branch according to user input.

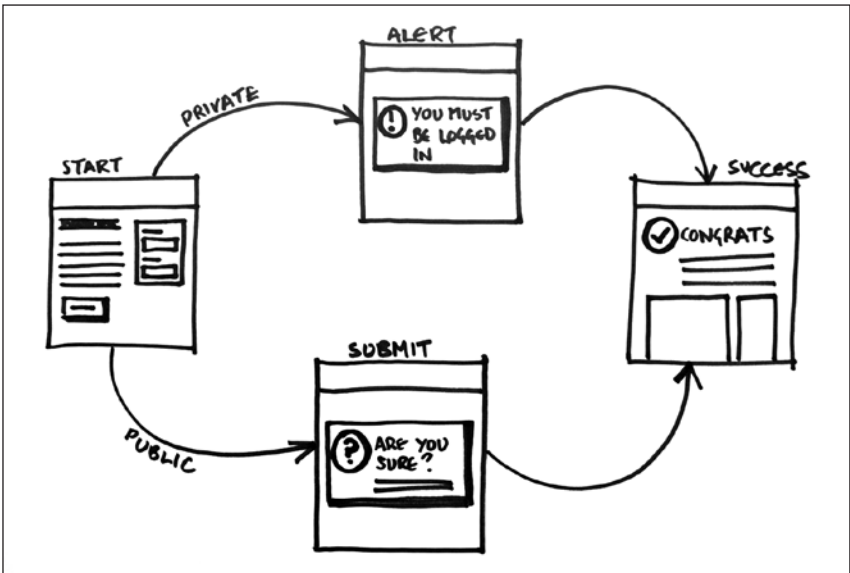
Under these circumstances a user flow may be more suitable. This deliverable describes how a user moves through a process, rather than capturing a hierarchical information structure. User flows are particularly suitable for task-driven sections of your site, such as creating a new account or adding an item to a shopping cart.

If only a few sections of your site are process-led, you can create a user flow in tandem with a sitemap; simply cross-reference the two using your numbering scheme.

User flows are great at explaining complex logic, but don't let them become swamped with detail. It's easy to overwhelm user flows with irrelevant minutiae. Instead, pay attention to the high-level interactions. You might also need to consult with your developers if your proposed solution involves tricky technical logic.

Wireflow

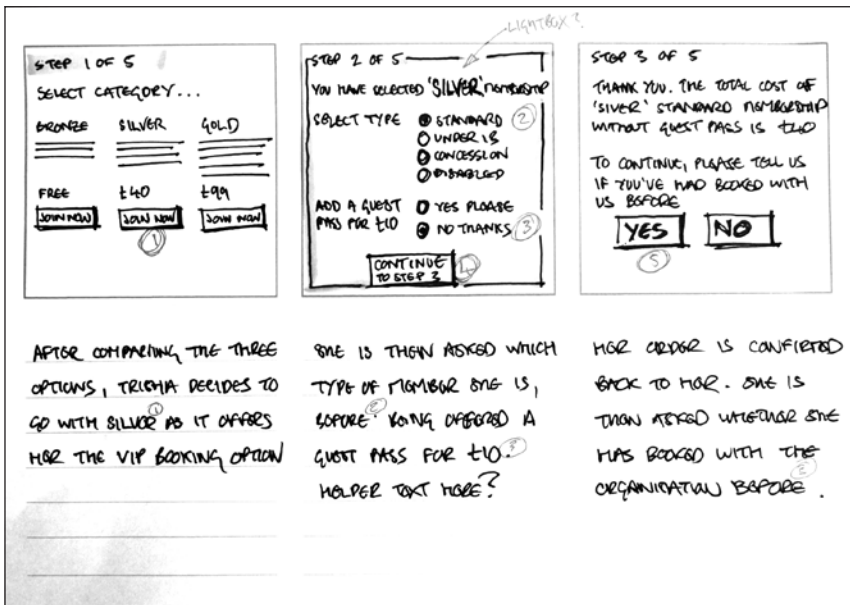
Like a user flow, a wireflow **D** shows how a user moves through the site, but replaces labels with rough representations of the relevant interfaces. Wireflows are particularly effective when the behavior of the interface is unconventional and demands visual explanation. Again, a wireflow and a sitemap can work well together.



D This sketched wireflow combines high-level user flow with basic screen layout.

Storyboard

Storyboards **E** are another powerful tool for illustrating user flow. We covered storyboards as research deliverables in Chapter 2, but you can also use them at this stage to show how the interface behaves and any offline activity or thought processes required of the user.



E Storyboards let you add instructional detail to complex interactions.

Scenarios, if created in your research phase, make excellent starting points for illustrating user journeys. You can show users in their context of use (home, office, and so on) and any interactions or emotional responses that take place off the screen.

Storyboards excel when an interaction is difficult to communicate using an abstract deliverable like a sitemap or user flow. A storyboard's visual and narrative approach gives an additional layer of context.

It's impractical to storyboard your entire site, but presented alongside other deliverables, storyboards can offer the extra level of detail required for crucial interactions.

To save time, use a single frame to show related interactions, labeled with numbers to show the order in which they occur. Thought and speech bubbles can be a useful adjunct to textual descriptions, adding character and helping to humanize the storyboard. Keep your output simple in case you want to start over.

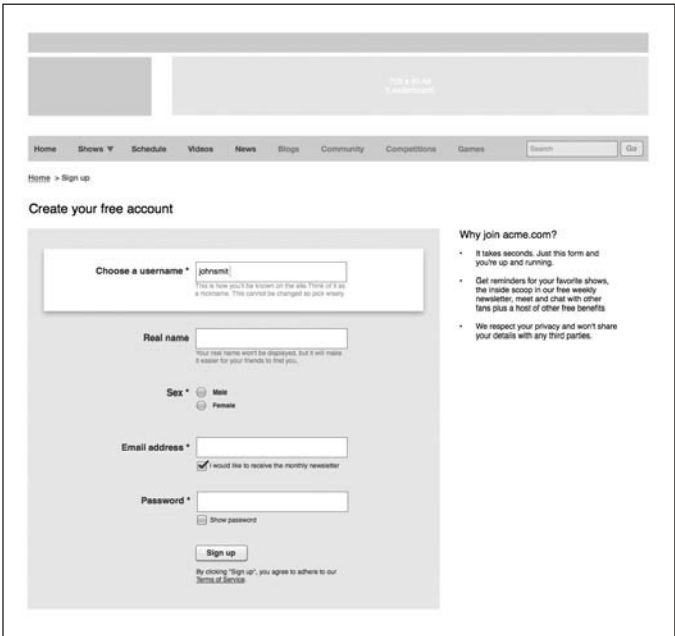
TIP

Save time by using a storyboard template, available for free online:

- Konigi's OmniGraffle template (www.konigi.com/tools/omnigraffle-ux-template) and Graph Paper (www.konigi.com/tools/graph-paper)
- Comic book-style templates from the Sun.com customer experience team (http://blogs.sun.com/MartinHardee/entry/design_comics_templates_1_o)

WIREFRAME

A wireframe **F** is a low-detail representation of an interface. It omits color, image detail, and other visual design specifics, providing instead a simple inventory of what's on the page and how it should be laid out.



F This wireframe shows the first screen of a sign-up process.

A wireframe addresses several design issues:

- **Information organization.** Which items should be grouped and where? Are there any particular relationships that need to be made more evident than others? How should these groups be prioritized?
- **Content.** What content needs to be present on the page? Will it be prose alone or does the page need to accommodate images and video?
- **Functionality and controls.** What can users do on the page? How will users navigate the site? Is there a search function? A log-in control? Are there any inputs such as forms?
- **States.** What are the various states of the page? How do forms handle errors? Does the page vary depending on the user's status—for example, logged in versus logged out?
- **Behavior.** Are there interactions that happen without a full page refresh? How does the page respond to input?
- **Metadata.** What page is this? How does it relate to the sitemap? What project does it belong to? Who is the author? What version is it?
- **Annotation.** Nuanced interactions or complex points may need further explanation. Use annotation callouts to highlight these areas.


NOTE

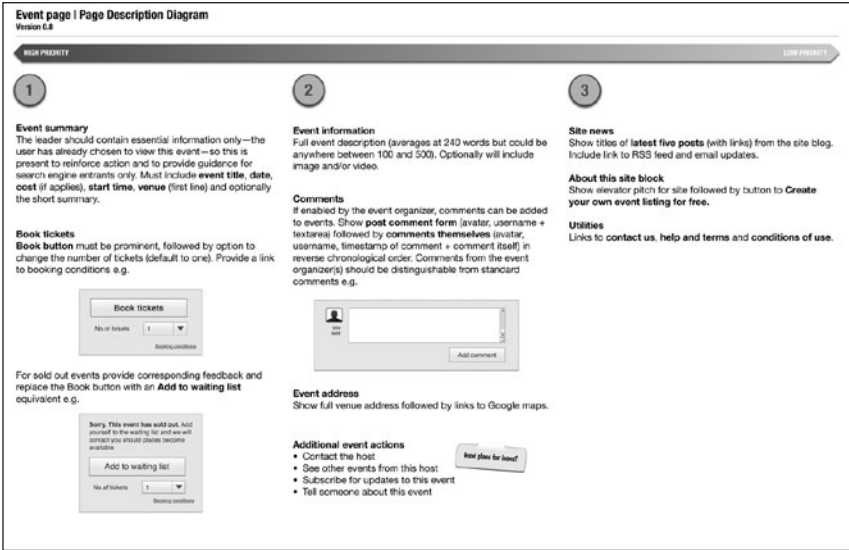
It can be hard to capture page behavior with static wireframes. If your design requires rich interaction, then storyboards or a prototype may be more suitable ways to explain site behavior.

The wireframe has long been the bread and butter of user experience design, and will undoubtedly be one of your most commonly made deliverables. With practice you should be able to produce a wireframe quickly, freeing up more time to discuss it with others and explain your design choices.

Try to ensure that your wireframes show consistency between pages; it can be easy to focus on an individual page and lose sight of how other pages are designed. If you are confident in your HTML skills, you may also want to include some suggestions about semantic markup on your wireframe, to help communicate its structure to your developer colleagues.

PAGE DESCRIPTION DIAGRAM

A page description diagram (PDD)  is a written inventory of every element on your web page. Elements are categorized into high, medium, and low priority and ordered horizontally.



 This page description diagram shows the contents of a page in priority order.

Wireframes may be the most commonly used tool in the UX designer's kit, but they aren't always suitable. If visual or graphic designers are currently in charge of deciding layout—what goes where on the page—they may complain that your wireframe relegates them to the role of “coloring the boxes.” A PDD might be a better approach in this scenario.

Dan Brown of EightShapes invented the PDD as a way for UX designers to explain content hierarchy without dictating the specifics of layout. The PDD describes what's important to a UX designer—the contents of the page and which are the most important for common user tasks—without stepping too far into the time-consuming world of layout.

PDDs leave the door open for visual designers to interpret page layout, while allowing stakeholders to focus on the important issues of content and priority. This makes PDDs ideal if you're trying to squeeze a user experience focus into an existing design process.

If an element within your PDD needs more visual detail, you can supplement its text description with a simple diagram.

PROTOTYPE

A prototype **H** is a simplified but functional model of a system, used to explore, communicate, and test a design. Users can move directly through a prototype, with pages following each other in chronological order.



H An HTML prototype being used for usability testing.

The previous deliverables are all in some way abstracted from the final product. Stakeholders must imagine how these documents will translate into a finished website. However, stakeholders who aren't familiar with user experience may not be convinced by skeletal deliverables. Instead, they will want to see solutions.

Prototypes excel at presenting solutions, since they involve the smallest leap of imagination between deliverable and end product. They let people experience the real flow of a system, meaning your stakeholders can feel firsthand the benefit of good user experience. It's better to test-drive a car than to look at it in the showroom.

An effective prototype will not only communicate your design to stakeholders, but also enthuse them in a way no other deliverable can. Get a senior manager excited by a prototype, and your battle to integrate UX is already half won.

Prototypes also have the great advantage that you can run usability tests with them. We'll cover that in the next chapter, but for now, get buy-in for the concept by explaining that a prototype allows you to gather insight into what works for real customers and what doesn't. Couple this with the message that a prototype will help you incorporate customer feedback before costs become too high, and you'll have a compelling argument for your case.

The most obvious downside of a prototype is the effort required. A prototype takes longer to build and calls for more specialized skill than other deliverables, and if your assumptions and decisions are wrong, you could waste time.

To minimize the time you spend on your prototype, remember that you're not looking to create the real thing. Don't worry about polish—quality of code, beautiful aesthetics—you're only after a platform to demonstrate and test your ideas. Rough edges are a necessary part of the bargain.

A prototype will typically change beyond recognition as the project progresses. Make it modular and disposable. It's not a good idea to use the code of a prototype as a basis for the actual website; you're more interested in making it work than making it perfect.

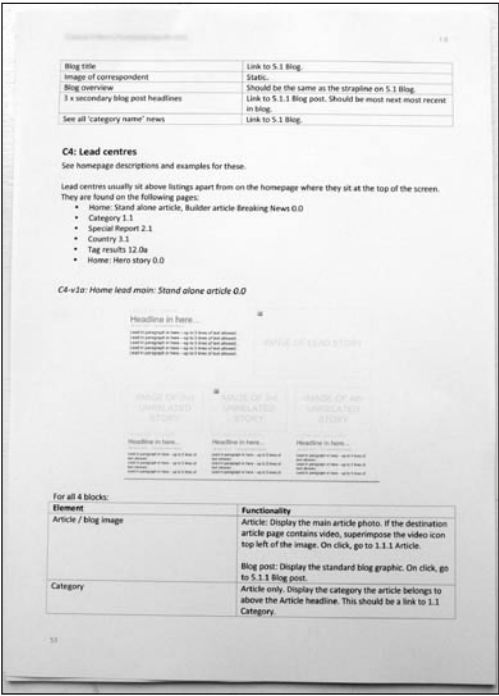
FUNCTIONAL SPECIFICATION

A functional specification **■** is a detailed document describing in full the behavior of your site, its functionality, and how it responds to user input.

Due to the effort involved, the functional specification has fallen out of favor in modern web design. However, some organizations—such as those in high-risk sectors that employ a “waterfall” model where design fully precedes development—still embrace this deliverable.

A functional specification poses a headache to any user experience designer, particularly an undercover one. Once created, it demands that the design be locked down, since the work required to change it later is so great. As such, UX designers often dissuade their businesses from demanding a functional specification, and instead try to sell the benefits of testing and iteration.

■ This functional specification documents the design and functionality of a news website.



If you have no other option than to create a functional specification, investigate whether this would be your responsibility or whether a product owner or business analyst might be able to generate one from a prototype. If the responsibility falls to you, set aside a lot of time and prepare for some painstaking work.

★ FIDELITY

The term *fidelity* describes the level of detail of your deliverables. *Low-fidelity* deliverables are a rough approximation of the intended user experience. They're quick and dirty, capturing the essential characteristics of the site. *High-fidelity* deliverables give a more accurate representation of the site, including not just core features but also specifics of the site's behavior and even aspects of the final visual design.

The right fidelity for your deliverables will depend on your project. Before talking about that decision, let's look at how fidelity affects the tools and techniques you use (and vice versa), and which fidelity is most suited to which deliverables.

LOW FIDELITY

Low-fidelity tools are increasingly popular in UX design. With no technical knowledge required, you can create lo-fi deliverables with very little fuss.

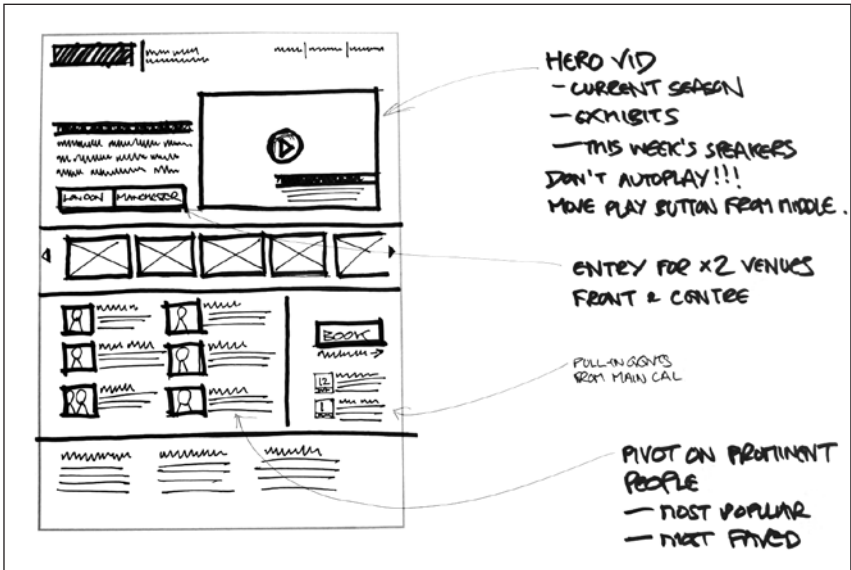
LOW-FIDELITY TOOLS

- *Paper.* Save time by making printed templates containing boxes, browser windows, and so on.
- *Pens.* As before, fat-tip markers for broad strokes, fine-liners for detail.
- *Scissors and sticky tape.* For cutting up and recombining your paper components.
- *Sticky notes, index cards, overhead-projector transparencies.* To mimic interactions like pop-ups and dialog boxes.

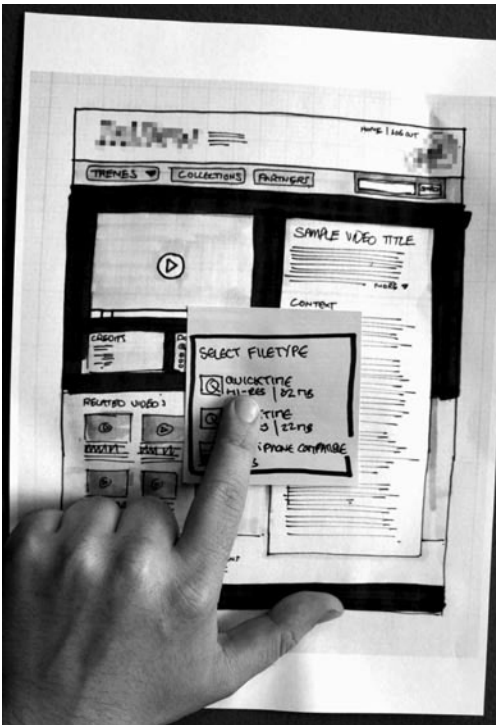
Pen and paper deliverables are extremely versatile: you can use them to produce almost any deliverable, from sitemaps to wireframes and even paper prototypes. Hand-drawn deliverables are very quick to produce and easy to improve upon, but production time will quickly mount up if you need to redraw the same elements over and over again.

Low-fidelity deliverables don't make for good formal documentation, but they're ideal when you need to convey decisions quickly. In Chapter 3 we explained how to use a detailed sketch to convey an idea; the hand-drawn wireframe ■ uses the same techniques and tools but offers more detail, proposing a solution rather than exploring a concept. It includes annotations that explain the major components of the page and how they work together.

Paper prototypes ■ are an easy way to test complex systems without the time and expense of making a digital prototype. Use sticky notes to represent panels that open and close, use overhead-projector transparencies for pop-up windows, and cut out your navigation and tape it to a background “master” so you don't have to recreate it on every page. Paper prototypes need someone “behind the scenes” to operate the system based on the actions of the user: opening drop-down menus, entering data, and so on.



I This low-fidelity wireframe, although hand-drawn, includes more detail than an exploratory sketch.



K This paper prototype uses sticky notes to mimic user interaction.

MEDIUM FIDELITY

When you need more detail than a sketch, it's time to pull out the professional tools. Although they demand more investment of time and money than a simple sketch, they provide professional output and the flexibility to make changes without having to redraw everything from scratch.

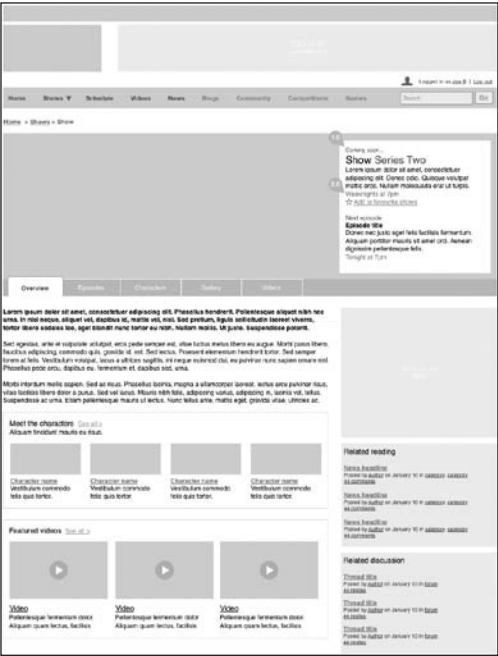
MEDIUM-FIDELITY TOOLS

- *The Omni Group OmniGraffle (for Mac, from \$100; for iPad, \$50; www.omnigroup.com/products/omnigraffle)*
- *Microsoft Visio (for PC, from \$260; <http://office.microsoft.com/en-us/visio>)*
- *Adobe Fireworks (for PC or Mac, \$299; www.adobe.com/products/fireworks)*
- *Adobe Illustrator (for PC or Mac, \$599; www.adobe.com/products/illustrator)*
- *Apple Keynote (for Mac, \$79; www.apple.com/iwork/keynote)*
- *Microsoft PowerPoint (for PC or Mac, \$230; <http://office.microsoft.com/en-us/powerpoint>)*
- *Balsamiq Mockups (for PC or Mac, \$79; www.balsamiq.com/products/mockups)*
- *Mockingbird (web-based, from \$9/month; <http://gomockingbird.com>)*
- *HTML image maps. Clickable “hotspots” can be positioned over any image in a web browser—a simple way to add hyperlinks to flat graphics files.*

If you want to create a family of deliverables with shared elements, you'll immediately benefit from being able to copy and paste elements between documents. Many of these tools offer sophisticated functionality such as master layers and shared libraries, allowing you to create your own collection of templates and stencils to reuse across projects. If you're producing lots of deliverables, this efficiency alone is probably a good enough reason to adopt medium-fidelity tools.

Professional tools let you export to a wide range of formats including graphics files, PDF, and even HTML, making it easy to share your deliverables with stakeholders across the world.

The medium-fidelity wireframe **L**, created using the Omni Group's OmniGraffle software, shows a page with placeholder images and annotations.



L A medium-fidelity wireframe conveys more detail than its low-fidelity counterpart.

If you want to demonstrate basic interactivity, you can also use medium-fidelity tools to create simple clickable wireframes **M**



M Clickable wireframes let you mock up basic interactions such as adding an item to a shopping cart.

Presentation tools like Keynote and PowerPoint, although not specifically designed for creating design mockups, make this surprisingly easy. Keynote in particular has several built-in motion transitions that mimic the visual behavior of interactive web applications.

Dedicated tools like OmniGraffle, Visio, and Fireworks include basic prototyping features and can easily be extended with shared libraries and stencils.

HIGH FIDELITY

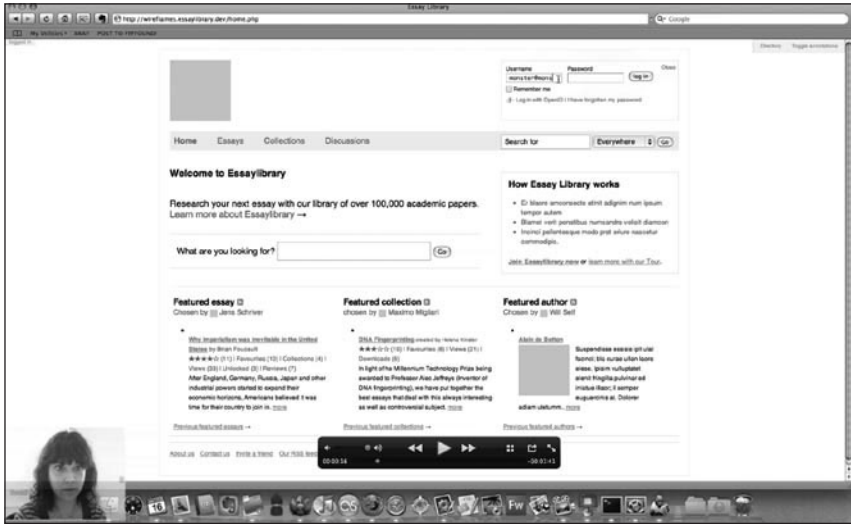
If your product includes particularly nuanced interactions, or you need to include significant visual detail, you should create high-fidelity deliverables. In the UX world, this high level of fidelity usually means creating a prototype.

HIGH-FIDELITY TOOLS

- *Adobe Dreamweaver (for PC or Mac, \$399; www.adobe.com/products/dreamweaver)*
- *Adobe Flash Professional (for PC or Mac, \$699; www.adobe.com/products/flash)*
- *Adobe Flash Catalyst (for PC or Mac, \$399; www.adobe.com/products/flashcatalyst)*
- *Axure (for PC or Mac, \$589; www.axure.com)*
- *HTML, CSS, and JavaScript. There's little or no cost attached to using these technologies for prototypes—all you need is a text editor and a browser.*

Specialist prototyping tools such as Axure can reduce the steep learning curve often associated with high-fidelity prototyping. Like some of the medium-fidelity tools, these specialist tools provide a WYSIWYG interface and generate functional HTML to allow your prototype to run in a browser.

Alternatively, you can get your hands dirty with HTML, CSS, and JavaScript yourself **N**. This can be daunting, but fortunately there are a number of libraries and frameworks that can reduce the legwork involved in layout, cross-browser compatibility, and interaction.



N Screenshot showing a usability test conducted on a high-fidelity HTML prototype.

NOTE

- Popular CSS frameworks include Blueprint (www.blueprintcss.org) and the 960 Grid System (www.960.gs).
- To prototype interactions, use JavaScript libraries such as jQuery (www.jquery.com) and MooTools (www.mootools.net). If you're not well versed in JavaScript, the IxEdit tool (www.ixedit.com) can help generate the code required.
- Basic PHP can be an excellent addition to your prototyping palette. PHP lets you reuse code efficiently across pages, and using basic string functions, you can mock up surprisingly complex search components.

CHOOSING FIDELITY

A useful principle for undercover work is to adopt the lowest fidelity necessary to get the job done. What you lose in detail you'll gain back in time and effort. However, the most appropriate level of fidelity for your deliverables will depend on several factors.

Audience

Your deliverables should be tailored for your stakeholders, so that you don't underdeliver or waste time that you could use elsewhere.

Stakeholders will have differing needs from your deliverables; specifically, they'll need enough information to do their jobs. For example, product owners will be eager to see whether solutions meet business objectives. Content specialists will want to know what copy needs to be created and where it should go. Developers will want to learn about required functionality.

In some instances, you'll be able to provide further detail and clear up ambiguity with a discussion. There's no point creating glossy deliverables for the person sitting next to you; sometimes, a simple picture and a conversation is all you need to get your message across.

Also consider what you learned in your cultural analysis of the organization. Some stakeholders will respond better to high-fidelity deliverables, whereas others will be happy with a sketch of the essentials.

You may be able to create a single deliverable that covers these diverse needs, or you may need to create a couple of versions to explain relevant items in more detail. Just remember that extra deliverables take extra time to create and maintain.

Finally, if you plan to conduct usability testing with these deliverables, you should use a higher-fidelity method, although this will also depend on the goal of your testing. Medium-fidelity deliverables might suffice for simple tests of basic elements, but for subtle interactions or those that will depend on particular content, a high-fidelity prototype is ideal.

Time

As an undercover UX designer, it's better to work on the design than create beautiful deliverables. While this clearly implies that low fidelity is desirable, you must find the balance. Too much haste will lead to ambiguity or confusion, meaning wasted time revisiting your deliverables or clarifying the situation. If in doubt, use the tools with which you are most comfortable.

You can reduce time pressures if you can take advantage of other resources. Could you convince the business to let a willing developer convert your sketches into a high-fidelity prototype?

Life span

Most deliverables are made obsolete when the site goes live, but until that point remember that every deliverable you create will have to be kept up-to-date. This can be especially time-consuming if you're creating several types of deliverable.

Scope

How much of the site experience do your deliverables need to capture? A sitemap and handful of wireframes that capture a single user journey may provide enough detail for your team, or you may need to depict lots of journeys and permutations. Be careful not to overcommit. Let your scope and fidelity be guided by what you can realistically achieve.

Innie or outtie

Outties—external agency staff or consultants—typically have to produce higher-fidelity deliverables than innies. Deliverables will usually form part of an agency contract, and can be the only way for the client company to assess the quality of thought and work they are getting for their money.

Agencies may have agreed formats for deliverables; be sure to match these to reduce the risk of wasted work.

★ BETTER DELIVERABLES

There's no perfect way to make deliverables; the best approach depends on the project and your personal style. However, since an undercover designer's deliverables will be central to the case for UX, here are some tips on how to get the most out of them.


TRY EVERYTHING, MASTER SOMETHING

Try out various tools to develop your proficiency and appreciate which of them are best suited to which deliverables. Most of the software listed in this chapter has a free trial version, so get downloading and experimenting.

You'll probably find that one tool suits you better than the others, or perhaps you'll be limited to one application since that's all that your organization can offer. Take the time to learn the nuances of that tool and become a real expert in it. Keyboard shortcuts will save you hours, as will learning how to create master templates and stencil libraries.

The largest risk with this approach is that you'll fixate on your chosen tool as the only appropriate answer to every problem. Be sure to choose your weapon based on the problem at hand, not just on whatever's most comfortable.

CONSULT DESIGN PATTERN LIBRARIES

A pattern library  documents effective solutions to a common problem. Each pattern explains the problem, the proposed solution, and the context in which the problem occurs.

As users have become more familiar with the web, standardized approaches to targeted problems have begun to emerge. Pattern libraries document these designs, and detail the rationale behind them.

Design patterns won't make your design decisions for you, but they can be a shortcut past some of the legwork you'd otherwise have to do yourself. They can save you time and reinforce your decisions with the knowledge that others have successfully tested these approaches over the years.



Popular pattern libraries include Yahoo Pattern Library (<http://developer.yahoo.com/ypatterns/>), Quince (<http://quince.infragistics.com/>), and Martijn van Welie's collection (www.welie.com).

However, be careful not to apply patterns blindly. A design that works elsewhere isn't guaranteed to work on your site, and a subtle difference in requirements can render an approach invalid.

BUILD A PERSONAL LIBRARY

As you gain experience, you'll inevitably build up a set of stencils, shapes, and templates that suits your style. By continually tweaking the set and making it your own, you'll end up with a set of tools that can act as your own artist's palette.

Similarly, you'll subconsciously build your own library of design patterns that work well for specific problems. Take the time to build these into a design pattern library of your own, and you'll find yourself becoming more efficient, focusing on your design rather than the tool you're using.

If you're on a team, share these stencils, templates, and design snippets on a shared drive or intranet. If you're prototyping in HTML, ask whether your developers have a library of code snippets that can save you coding time.

LEARN HTML AND CSS

While we're on the topic, since HTML and CSS are the media you work in, learn them. Just as a composer must know the range of each instrument she writes for, learning HTML and CSS will help you truly understand the possibilities and limitations of web technology.

The fundamentals are surprisingly simple and take only a couple of weeks to pick up. Don't worry about advanced tricks, but learn enough so that you would know how to start coding your wireframes. Not only will HTML and CSS give you another viable prototyping option, but also they will improve the quality of your designs and help you communicate better with your developer colleagues.

TIP

- There are dozens of online tutorials to help you learn HTML and CSS, such as www.w3schools.com; and books such as *Designing with Web Standards*, (3rd edition) by Jeffrey Zeldman (New Riders, 2009), which give an excellent grounding in modern standards-based web development.
- The WaSP InterAct curriculum can be a useful reference (<http://interact.webstandards.org/curriculum/front-end-development>).

IDENTIFY KEY PAGES

Each site has a couple of pages that lie at the heart of the user experience. For example, the photo-sharing site Flickr centers around just two key pages: the photo page and a user's photo stream. Without these, the hundreds of other pages on the site could not exist.

Make key pages your top priority since their structure and design will affect the rest of the site. You should already have a hunch about which pages are key—your sitemap and design principles will often help to highlight them. Give key pages the attention they deserve, and your site's user experience is off to a great start.

It's rare that the homepage is a key page. Although it's an irresistible draw for stakeholders, the homepage's main job is usually to direct visitors to the key pages. Designing a good homepage is easier once you've defined the core essence of the site; as such, consider leaving it until later.

GET THE MAIN FUNCTION RIGHT

Similar logic applies within a page. It's better to get a page's main function absolutely right and forget about the rest, than to spread your effort thinly. It's easy to lose sight of what really matters when your head's full of competing requirements. If you find yourself struggling, break the problem down. What is the one thing this page needs to accomplish? Forget the secondary content—sidebars, footers, and so on—and focus on the core.

EXPLORE AS YOU GO

Although you're starting to converge toward a final design, don't be afraid to explore new alternatives as they crop up. You can use any of the techniques discussed in Chapter 3.

The line between generating ideas and refining them is blurry. It's OK to switch between the two modes, although in later stages you should focus on polishing the details rather than generating additional ideas.

CONSIDER THE 'FIRST-RUN' EXPERIENCE

Pay special attention to the user's first experience with your site. Is the site clear about what it does and how it should be used? It's easy to neglect this state if you're short on time, but even though a user may only see this state once, the first experience forms the important first impression.

If you're designing a site that relies on user-generated content, remember to design the "blank slate": how the page will look without this content. A design that seems active and useful once it's full of content can appear imbalanced and deserted without that content. Provide direction and incentives for users to contribute so that the site's emptiness doesn't appear intimidating.

USE REAL CONTENT

For years, UX designers created wireframes and prototypes with “lorem ipsum,” dummy text indicating that the real content would follow later.

Designing around dummy text means that you have no way of knowing if the main visual focus of the page matches the main written focus of the page. Without knowing what’s on the page, you may leave too much space, or too little. Dummy text also means you can’t run valid usability tests on the system, since real users will want to read the text and look for trigger words.

Use real content wherever possible. If the content is unavailable, outdated, or being rewritten, talk with your content providers about the contents of the page, what its features are, how long it is, and what the main focus and calls to action should be.

As an undercover designer, you may have to use lorem ipsum if you don’t have access to content providers. However, aim to replace it with real content—even a first draft—as soon as you’re able. Only then will you be able to judge whether the design and content together meet your users’ goals.

NOTE

If you must use lorem ipsum, online tools such as www.lorem2.com can generate the text for you in a variety of formats.

COME UP FOR AIR

Creating deliverables can be enjoyable, but be careful not to drown in detail.

A simple and effective way to avoid losing focus is to build regular peer review into your routine. We’ll cover formal critique in the next chapter, but before that point, consider whether you can share work in progress with close coworkers who are sympathetic to the UX cause. This could be something as simple as an impromptu discussion: simply print out your designs and take them to your colleagues for their thoughts. If you find yourself stuck on a problem, a few minutes spent describing it to colleagues can force you to focus on the heart of the issue.

A more advanced form of peer review is pair design, where you and a colleague work on a design simultaneously at one desk. Pair programming, the developer equivalent, is common within Agile environments (see Chapter 6, “Working

with...”), where it’s used to improve quality and create easily maintainable code. Applying the same technique to UX design can drive up the quality of your work and help colleagues buy in to your recommendations.

Although it’s common for UX designers to conduct pair design together, you can partner with anyone involved in the project: developers, visual designers, content specialists, and so on. If you’re using software while pair designing, let one person “drive,” sitting at the computer, while the other provides input and suggestions. Swap every hour or so to keep things fresh.

Of course, pair design takes two people, so it can be tricky to arrange. However, it will teach you a great deal and can help build strong relationships.

MAKING IT REAL: FURTHER READING

- *Web Anatomy: Interaction Design Frameworks that Work*, by Robert Hoekman, Jr. & Jared Spool (New Riders, 2010)
- *Prototyping: A Practitioner’s Guide*, by Todd Zaki Warfel (Rosenfeld Media, 2009)
- *Search Patterns*, by Peter Morville & Jeffery Callender (O’Reilly Media, 2010)
- *A Project Guide to UX Design: For User Experience Designers in the Field or in the Making*, by Russ Unger and Carolyn Chandler (New Riders, 2009)
- *Web Form Design: Filling in the Blanks*, by Luke Wroblewski (Rosenfeld Media, 2008)
- *Designing Interfaces: Patterns for Effective Interaction Design*, by Jenifer Tidwell (O’Reilly Media, 2006)
- *Communicating Design: Developing Web Site Documentation for Design and Planning*, by Daniel M Brown (New Riders, 2007)
- *Information Architecture: Blueprints for the Web, 2nd Edition*, by Christina Wodtke and Austin Govella (New Riders, 2009)

This page intentionally left blank

INDEX

101 Things I Learned in Architecture School, 67
960 Grid System, 87

A

A/B testing, 128–129, 146
accessibility, 7, 152
Adaptive Path, 59, 63, 167
Adlin, Tamara, 40
Adobe, 84, 86, 109
adoption ladder, UX, 160–161
Agile, 94–95, 153–157
analytics data, 34
analytics snapshot, 8–11
Anderson, Stephen, 61
annotation callouts, 77
Apple Keynote, 84, 86
Art of Looking Sideways, The, 67
associations, 166
Audacity, 27
authority, sign-off, 21
automated remote tests, 111–112
Axure, 86

B

BabyNameWizard.com, 37
Back of the Napkin, The, 67
Balsamiq Mockups, 84
black-hat SEO, 148
blogs, UX, 31, 166
Blueprint, 87
Boersma, Peter, 167, 177
Bolt, Nate, 130
bounce rate, 9
brands, 144, 145
briefs, 16
Brooks, Kevin, 177
Brown, Dan, 40, 78, 95
Brown, Sunni, 67
browser data, 10
bug reports, 124
Building Findable Websites, 157
Buley, Leah, 59
business
 benefits, 152–153
 objectives, 14, 16
 research, 16–19
Butler, Jill, 67
Buxton, Bill, 67
buzz, building, 12–13

C

Callender, Jeffery, 95
Camtasia Studio, 103
card sorting, 30–31, 107
Card Sorting: Designing Usable Categories, 40

career opportunities, 172–174
case studies, 20, 163, 166
cash cows, 20, 127
cash incentives, 26, 102
cast shadows, 48
Chalkmark, 111
Chandler, Carolyn, 40
Chisnell, Dana, 130
Cialdini, Robert, 163, 177
Cisco WebEx, 109
Clearleft, 103
cluster analysis, 31
CMS, 22, 140
collaborative design, 42, 51–61
collaborative design workshops, 52–54, 62
collaborative mindset, 22–23
Comic Life, 39
comics, 38–39
CommandShift3, 111
Communicating Design, 40, 95
competitive analysis, 8
compliance, regulatory, 152
concept maps, 39
conferences, 166
confidentiality agreements, 31, 101
ConnectNow, 109
consent forms, 104
constraints, 15
content
 audits, 11–12
 bottlenecks, 22
 confusion, 22
 hierarchy, 78
 inventories, 11–12
 real vs. dummy, 94
 specialists, 139–140, 149
 strategies, 139–140
content-creation process, 139, 149
content management systems, 22, 140
Content Strategy for the Web, 40, 157
contextual inquiries, 28–29
conversion rate, 10
Cooper, Alan, 36, 40
corridor tests, 26–27, 112
cost savings, 152–153
Craigslist, 25
CrazyEgg, 11
creative prompts, 61
critiques, 113–123
 common problems with, 119–120
 defined, 113
 following up on, 123
 planning, 113–114
 running the session, 115–116
 translating stakeholder comments, 116–118
 winning debates during, 120–122

CSS, 86, 87, 92
 cultural factors, 19–23
 customer feedback. *See* user feedback
 customer research. *See* user research
 customer satisfaction, 14
 customer-service application, 124
 customer-service representatives, 30
 customer surveys, 29

D

deadlines, 21–22
 deliverables, 70–95
 defined, 70
 fidelity considerations, 81–89
 functional specifications, 80–81
 making better, 90–95
 page description diagrams, 78–79
 prototypes, 77, 79–80, 82, 86
 recommended books on, 95
 role of, 70–71
 sitemaps, 71–73
 tailoring for audience, 88
 updating, 89, 98
 user flows, 73–76
 wireframes, 76–77
 demographic information, 36
 design by committee, 21
 Design Consequences game, 58–59
 design critiques. *See* critiques
 design deliverables. *See* deliverables
 design disinterest, 20
 design games, 54–61
Designing for the Digital Age, 40
Designing Interfaces, 95
Designing with Web Standards, 92, 157
 design leadership, 169
 design mockups, 86
 design objectives, 14
 design pattern libraries, 90–91
 design principles, 64–66
 design problems, 13–23
 and cultural factors, 19–23
 defining, 13
 framing, 14–15
 researching the business, 16–19
 subjectivity of, 15
 design process. *See also* UX design
 collaborating on, 51–61
 complexity of, 13
 fluid nature of, 132
 generating ideas for, 42–43, 61
 recommended books on, 67
 role of critiques in, 113
 role of deliverables in, 70–71 (*See also* deliverables)
 sustaining momentum for, 62
 design research, 143
 design reviews, 113–114. *See also* critiques
 Design the Box game, 54–56
 Design with Intent Toolkit, 61

design workshops, 52–54, 62
 desktop software, 4
 developers, 132–136
 diary studies, 32
 direct observation, 29
 divergent thinking, 42–43, 61
 drawing tools, 46–47
 Dreamweaver, 86
 drop shadows, 48
 dummy text, 94
 dynamic pages, 149

E

EightShapes, 78
 emergent UX phase, 160
 entry pages, 9
Essential Persona Lifecycle, The, 40
 Ethnio, 109
 Excel, 72
 exclusions, 15
 executive summaries, 33–34
 expectations, 21
 expert reviews, 6–7, 8

F

face-to-face interviews, 27–28
 feature requests, 15, 30, 160
 feedback, 120, 124, 127. *See also* user feedback
 Feedback Army, 111
 fidelity, 81–89
 figures, drawing, 49–50
 Fireworks, 84, 86
 “first-run” experience, 93
 Fitts’ Law, 122
 FiveSecondTest, 111
 Flash Catalyst, 86
 Flash Professional, 86
 Fletcher, Alan, 61, 67
 Flickr, 92
 fly-on-the-wall technique, 29
 focus groups, 28, 30
 formative testing, 100
 form shadows, 48
 Fried, Jason, 157
 functional specifications, 80–81
 FY Threshold, 122

G

Gabriel-Petit, Pabini, 177
 games, design, 54–61
 Gamestorming, 67
 GarageBand, 27
 Garrett, Jesse James, 72
 generalists, user experience, 167, 170, 177
 Get Satisfaction, 124
Getting Real: The Smarter, Faster, Easier Way to Build a Successful Web Application, 157
 gift certificates, 110
 Godin, Seth, 173, 177
 Goodwin, Kim, 40

Google

- Alerts, 124
- Analytics, 9
- top corporate principle, 147
- Webmaster Central help pages, 157
- Website Optimizer, 128

GoToMeeting, 109

Govella, Austin, 40, 95

Graph Paper, 76

Gray, Dave, 67

group interviews, 28

guest observers, 104

H

Halvorson, Kristina, 40, 157

Handbook of Usability Testing, 130

Hansson, David Heinemeier, 157

heat maps, 11

heuristic evaluation, 6–7

high-fidelity deliverables, 81, 86–87

Hoekman, Robert, 95

Hohmann, Luke, 67

Holden, Kritina, 67

homepages, 9, 25, 92, 146

hotspots, 84

How Designers Think, 67

HTML, 84, 86, 87, 92

human-computer interaction experts, 6

I

Illustrator, 84

image maps, 84

incentives, 26, 102, 110, 124

industry experts, 141

Influence: The Psychology of Persuasion, 177

information architecture, 11

Information Architecture: Blueprints for the Web, 40, 95*Information Architecture for the World Wide Web*, 177

Information Architecture Institute, 166

information organization, 77

Inmates Are Running the Asylum, The, 36, 40

innies vs. outties, 16, 19, 89

Innovation Games, 67

inspirational triggers, 61

integrated UX phase, 161

Interaction Design Association, 166

interviews, 16–19, 27–28

IntuitionHQ, 111

iteration, 98–99

IxDit tool, 87

J

JavaScript, 86, 87

jQuery, 87

K

Kawakita, Jiro, 56

Keynote, 84, 86

key pages, 92

keywords, 9, 148, 149

KJ Method, 56–57

Konigi, 76

Krug, Steve, 130

Kübler-Ross model, 124–125

Kunlavsky, Mike, 40

L

landing pages, 149

launch decisions, 123–124

Lawson, Bryan, 67

leadership, UX, 168–172

lettering, 50

libraries, pattern, 90–91

Lidwell, Will, 67

Likert scales, 29

Linderman, Matthew, 157

line drawings, 47–48

LiveLook, 109

local maximum, 129

Lockton, Dan, 61

Loop11, 111

“lorem ipsum” text, 94

low-fidelity deliverables, 81–83

M

Macanauf, James, 67

mailing lists, UX, 31

managers, 150–153, 169, 171–172

manifesto, undercover, 3, 160, 162

Marcotte, Ethan, 157

markers, 46–47, 52–53

marketers, 144–146

market reports, 31

market research, 23, 28, 144, 145. *See also* user research

maturing UX phase, 161

medium-fidelity deliverables, 84–86

mental models, 7, 39, 40, 125

Mental Models: Aligning Design Strategy with Human Behavior, 40

Mental Notes, 61

mentors, 166

metadata, 77

metrics, 14, 126

microcopy, 128

Microsoft, 72, 84, 86

Mint, 9

mobile applications, 4

Moblify game, 57–58

Mockingbird, 84

mockups, design, 84, 86

Molich, Rolf, 6

MooTools, 87

Morae, 103

Morville, Peter, 95, 177

Mulder, Steve, 40

multivariate testing, 128–129

N

name charts, 37
 Navflow, 111
 navigation paths, 10
 NDAs, 31, 101
 negative feedback, 127
 Net Promoter Score, 14, 162
 Neumeier, Marty, 157
 newsletters, 163
 Nielsen, Jakob, 6
 nondisclosure agreements, 31, 101
 NPS, 14. *See also* Net Promoter Score
 numerical data, 34–35, 126

O

objectives, 14
 observers, guest, 104
Observing the User Experience, 40
 OmniGraffle, 76, 84, 85
 Omni Group, 72, 84
 OmniOutliner, 72
 one-on-one interviews, 27–28
 “one-up” template, 60
 online tutorials, 92
 open design, 125
 OpenHallway, 111
 OptimalSort, 30
 organizational culture, 19–23
 outlines, 72
 outputs
 user design, 70–71 (*See also* deliverables)
 user research, 33–39
 outsourced developers, 135–136
 outties vs. innies, 16, 19, 89

P

page description diagrams, 78–79
 page titles, 149
 page views, 10
 pair design, 94–95
 pair programming, 155
 pattern libraries, 90–91
 paying customers, 10
 PDDs, 78–79
 peer reviews, 94
 pen-and-paper deliverables, 82
 pens, 46–47
 people, drawing, 49–50
 Performable, 128
 personas, 35–38, 61, 63, 71
 persuasion, psychology of, 163–164, 177
 phases, undercover UX, 160–161
 phone interviews, 27–28
 photo journals, 32
 PHP, 87
 Plasq’s Comic Life, 39
 portfolios, 174–175
 PowerPoint, 84, 86
 power structures, organization, 21
 presentation tools, 86

principles, design, 64–66
 process-heavy organizations, 22
 product owners, 141–143
 professional organizations, 166
Project Guide to UX Design, A, 40, 95
 project objectives, 14
 prototypes, 77, 79–80, 82, 86
Prototyping: A Practitioner’s Guide, 95
 Pruitt, John, 40
 psychographic information, 36

Q

quantitative data, 34–35, 129
 Quesenberry, Whitney, 177
 Quince pattern library, 91

R

Rapid Iterative Testing and Evaluation, 107
 rebranding, 145–146
 recording usability tests, 103
 recruiters, research, 24
 recruiting
 usability test candidates, 101–102, 109–110
 UX staff, 171
 red flags, 20–22
 referring sites, 9
 registered users, 10
 regulatory compliance, 152
 Reichelt, Leisa, 58
 relationships, building, 22–23
Remote Research, 130
 remote testing, 108–112
 Request for Proposal, 16
 requirements, 14, 16
 research
 business, 16–19
 design, 143
 outputs, 33–39
 recording/analyzing, 32
 recruiting services, 24
 third-party, 31
 user, 23–33 (*See also* user research)
 résumés, 174–175
 return on investment, 126, 147, 164–165
 RFPs, 16
 Rhodes, John, 157
 risk, 152
 RITE, 107
 Roam, Dan, 13, 40, 67
Rocket Surgery Made Easy, 130
 ROI, 126, 147, 164–165
 Rosenfeld, Louis, 177
 Rubin, Jeffrey, 130

S

Santa Maria, Jason, 43
 scenarios, 38, 75
 Schauer, Brandon, 59
 Screencast-o-matic, 103
 screen-sharing software, 109

search engine marketing, 150
 search engine optimization, 146
 search engines, 9
 search logs, 10–11
Search Patterns, 95
Selling Usability, 157
 SEM, 150
 semantic markup, 77
 senior managers, 150–153
 SEO specialists, 146–150
 shadows, 48–49
 shapes, 47–50
 sign-off authority, 21
 sign-up forms, 146
 Silverback, 103
 site analytics, 8–11, 112, 126
 sitemaps, 11–12, 71–73
 site reviews, 6. *See also* critiques
 Six-to-One game, 59–60
 “six-up” template, 59
 sketchboards, 63
 sketchbooks, 46
 sketching, 43–50
 purpose of, 43
 sharing results of, 62, 63
 techniques, 47–50
 tools, 46–47
 types of, 43–46
Sketching User Experiences, 67
 Skype, 28, 109
 sliding scales, 117
 SMART objectives, 14
 SMEs, 31–32
 Snapz Pro, 103
 social media activity, 11, 124
 social networking sites, 141
 social networks, 124
 specialists
 content, 139–140, 149
 SEO, 146–150
 user experience, 4, 167, 168, 170, 177
 Spencer, Donna, 31, 40, 54
 Spool, Jared, 31, 95
 stakeholder interviews, 16–19
 stakeholder review sessions, 99
 standards-based web development, 92
 states, page, 77
 static pages, 149
 statistics, 14, 34, 126, 175
 sticky notes, 53, 56–57, 82
 stories, user, 155–156, 166
 storyboards, 38–39, 48, 75–76, 77
Storytelling for User Experience, 177
 subject-matter experts, 31–32
 summative tests, 99–100
 Sun.com, 76
 SurveyMonkey, 29
 surveys, 29, 35
 synchronous remote tests, 109–110

T
 tacit knowledge, 15
 team members, 170–171
 TechSmith, 103
 templates
 card sort analysis, 31
 consent form, 104
 “one-up,” 60
 “six-up,” 59
 storyboard, 76
 testing methods, 107–112, 128–129. *See also*
 usability testing
 text generators, 94
 text outlines, 72
 textures, 48–49
 third-party research, 31
 Tidwell, Jenifer, 95
 time constraints, 21–22
 time savings, 152–153
 T-model, 167–168, 177
 Tognazzini, Bruce, 6
 Treejack, 111
 tree tests, 107–108
 trigger word analysis, 32
 trustworthiness, 7
 TryMyUI, 111
 Tulathimutte, Tony, 130
 tutorials, 92
 Twitter, 124

U
 undercover manifesto, 3, 160, 162
 undercover UX techniques, 6–13. *See also* UX
Unfolding the Napkin, 13, 40
 Ungur, Russ, 40
 unique visitors, 9
Universal Principles of Design, 67
 unrequirements, 15, 16
 URLs, 148–149
 usability testing, 99–112
 choosing venue for, 102
 combining with other methods, 107–108
 deciding what to test, 100–101
 do's and don'ts, 105–106
 following up after, 105, 106–107
 guest observers for, 104
 moderating, 102, 106
 presenting findings from, 112
 with prototypes, 79, 80
 recommended books on, 130
 recording, 103, 104
 recruiting candidates for, 101–102, 109–110
 taking notes during, 102–103
 time required for, 101
 types of, 99–100
 Usabilla, 111
 user-centered design, 3, 161
 user experience. *See also* UX
 generalists, 167, 170, 177
 specialists, 4, 167, 168, 170, 177

user feedback, 30, 80, 100, 124–125
 user flows, 73–74
 user-friendly language, 7
 user interface controls, 49
User Is Always Right, The, 40
 user knowledge, 152
 Userlytics, 111
 user personas. *See* personas
 user requirements, 14
 user research, 23–40
 determining need for, 23–24
 finding participants for, 24–26
 getting permission for, 24
 methods, 26–32
 outputs, 33–39
 recommended books on, 40
 recording/analyzing, 32
 user-scented design, 3
 user stories, 155–156, 166
 user surveys, 29
 UserTesting, 111
 UserZoom, 111
 UX
 adoption ladder, 160–161
 blogs, 31, 166
 design (*See* UX design)
 leadership, 168–172
 mailing lists, 31
 as mindset vs. process, 3
 newsletters, 163
 research (*See* research)
 résumés/portfolios, 174–175
 showing value of, 6
 starting conversation about, 6
 stories, 166
 undercover techniques, 6–13
 UX design
 and Agile, 153–157
 balancing business/user needs, 23
 building skills in, 166–168
 collaborating on, 51–61
 and content specialists, 139–140
 and developers, 132–136
 fluid nature of, 132
 fundamentals of, 2
 generating ideas for, 42–43
 getting support for, 160–168
 human angle in, 132
 importance of “first-run” experience, 93
 launch phase, 123–124
 making changes in, 127
 manifesto, 3
 and marketers, 144–146
 measuring happiness with, 14
 overcoming challenges of, 2
 and product owners, 141–143
 recommended books on, 67
 return on investment of, 126, 147, 164–165
 and senior managers, 150–153
 and SEO specialists, 146–150
 sketching, 43–50

testing, 99–112 (*See also* usability testing)
 and visual designers, 136–138
UX Leadership Journal, The, 177

V

Validation Stack, 121–122
 VALS segmentation tool, 36
 verticals, 141
 Visio, 84, 86
 visitors, site, 9
 visual communication, 44–45
 visual designers, 136–138
 visual vocabulary, 72, 73
 VoIP software, 27–28

W

W3Schools.com, 92
 Walter, Aaron, 157
 Warfel, Todd Zaki, 95
 WaSP InterAct curriculum, 92
 waterfall projects, 153–154
 weapons of influence, 163–164
Web Anatomy, 95
 webcams, 103
 web development, standards-based, 92
 WebEx, 109
Web Form Design, 95
 Webnographer, 111
 Website Optimizer, Google, 128
 websites
 characteristics of good, 7
 dynamic vs. static, 149
 identifying key pages for, 92
 mapping structure of, 11–12, 71–73
 revisiting/changing, 98–99, 127
 setting design principles for, 64–66
 WebSort, 30
 web UX design, 4. *See also* UX design
 Welie, Martijn van, 91
 WhatUsersDo, 111
 whiteboards, 45, 46
 white-hat SEO, 148
 wireflows, 74
 wireframes, 71, 76–77, 78, 83, 85
 Wodtke, Christina, 40, 95
 workshops, collaborative design, 52–54, 62
 writers, 149
 Wroblewski, Luke, 95
 WYSIWYG interface, 86

Y

Yaar, Ziv, 40
 Yahoo Pattern Library, 91
 Young, Indi, 39, 40

Z

Zeldman, Jeffrey, 92, 157
Zog: The Number One Strategy of High-Performance Brands, 157