

Extracted from:

Designed for Use

Usable Interfaces for Applications and the Web

This PDF file contains pages extracted from *Designed for Use*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2011 The Pragmatic Programmers, LLC.

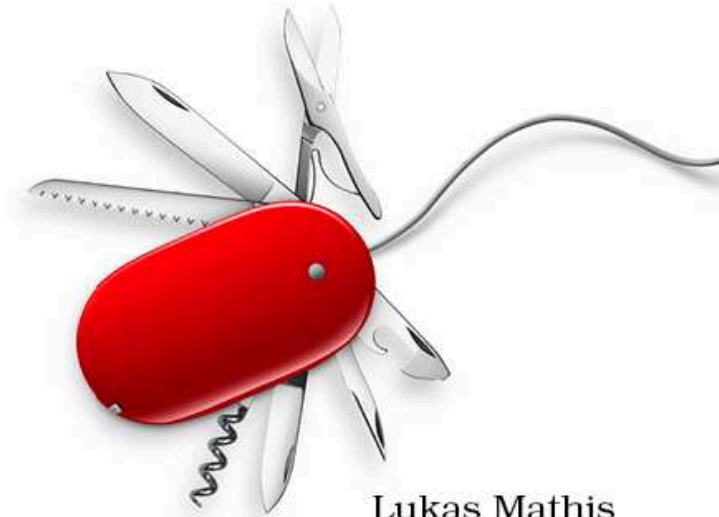
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The
Pragmatic
Programmers

Designed for Use

*Create Usable Interfaces
for Applications and the Web*



Lukas Mathis
edited by Jill Steinberg

Contents

Before We Start, a Word	14
Technique Chapters	14
Idea Chapters	15
How the Book Is Organized	17
Just One More Thing	18
I Research	19
1 User Research	21
2 Job Shadowing and Contextual Interviews	25
2.1 Observing Your Audience	26
2.2 Job Shadowing	26
2.3 Contextual Interviews	27
2.4 Remote Shadowing	28
2.5 Limitations of Contextual Interviews	28
3 Personas	32
3.1 Problems with Personas	33
3.2 Creating Personas	34
3.3 Working with Personas	35
3.4 Personas Do Not Replace User Research	36
4 Activity-Centered Design	39
5 Time to Start Working on Documentation	42
5.1 The Manual	43
5.2 Blog Posts	43
5.3 Screencasts	44
5.4 Press Releases	44
5.5 Talk About Tasks	45

6	Text Usability	48
6.1	Why Words Matter	48
6.2	People Don't Want to Read	49
6.3	Say Less	50
6.4	Make Text Scannable	51
6.5	No Fluff	51
6.6	Sentences Should Have One Obvious Interpretation	52
6.7	Talk Like a Human, Not Like a Company	53
6.8	Illustrate Your Points	54
6.9	Use Words People Understand	55
6.10	Test Your Text	56
6.11	Display Legible Text	57
7	Hierarchies in User Interface Design	60
7.1	Creating Hierarchical Structure Visually	61
8	Card Sorting	65
8.1	Designing Hierarchies	66
8.2	Preparing for a Card Sort	67
8.3	Participants	68
8.4	Running a Card Sort	69
8.5	Running a Remote Card Sort	71
8.6	Evaluating the Results	72
8.7	Guidelines for Creating Usable Hierarchies	73
9	The Mental Model	79
9.1	What People Think	79
9.2	Three Different Models	81
9.3	Hiding Implementation Details	81
9.4	Leaky Abstractions	84
9.5	Designing for Mental Models	85
II	Design	95
10	Sketching and Prototyping	97
10.1	Designing the Structure	98
10.2	Flow Diagrams	98
10.3	Storyboards	99
10.4	Sketching	99
10.5	Wireframes	101
10.6	Mock-ups	101
10.7	Tools	103

11	Paper Prototype Testing	106
11.1	Guerilla Paper Prototype Testing	107
11.2	Running Full Usability Tests with Paper Prototypes	109
12	Realism	122
12.1	Symbols	123
12.2	Virtual Versions of Real-World Objects	125
12.3	Replicating Physical Constraints in Digital Products	128
13	Natural User Interfaces	132
13.1	Avoid Gesture Magic	133
13.2	Recognizing Gestures	134
13.3	Accidental Input	136
13.4	Conventions	137
14	Fitts's Law	140
14.1	Screen Edges Have Infinite Size	141
14.2	Radial Context Menus Decrease Average Distance .	142
14.3	Small Targets Need Margins	145
14.4	Sometimes, Smaller Is Better	145
15	Animations	147
15.1	Explaining State Changes	147
15.2	Directing User Attention	148
15.3	Avoid Unimportant Animations	150
15.4	Help Users Form Suitable Mental Models	150
15.5	Learning from Cartoons	152
16	Consistency	157
16.1	Identifying Archetypes	157
16.2	Behavioral Consistency	158
17	Discoverability	161
17.1	What to Make Discoverable	161
17.2	When to Make Things Discoverable	163
17.3	How to Make Things Discoverable	164
18	Don't Interrupt	167
18.1	Make Decisions for Your User	168
18.2	Front Load Decisions	169
18.3	Interrupt Users Only For Truly Urgent Decisions . .	170

19	Instead of Interrupting, Offer Undo	173
19.1	Let Users Undo Their Actions	174
19.2	Temporary Undo	175
20	Modes	177
20.1	Nonobvious Modes	178
20.2	Unexpected Modes	182
20.3	Sticky Modes	182
20.4	Modes Are Not Always Bad	183
20.5	Quasimodes	183
21	Have Opinions Instead of Preferences	185
21.1	Why Preferences Are Bad	187
21.2	How to Avoid Preferences	188
21.3	If You Can't Avoid Preferences	189
22	Hierarchies, Space, Time, and How We Think About the World	191
22.1	Hierarchies	192
22.2	Space	193
22.3	Time	195
22.4	A Better Hierarchical System	196
23	Speed	200
23.1	Responsiveness	201
23.2	Progress Feedback	201
23.3	Perceived Speed	203
23.4	Slowing Down	204
24	Avoiding Features	207
24.1	Remember the User's Goals	208
24.2	The Five Whys	208
24.3	Instead of Adding a New Feature, Make an Existing Feature More Usable	210
24.4	Solve Several Problems with One Change	210
24.5	Consider the Cost	211
24.6	Make It Invisible	211
24.7	Provide an API and a Plug-in Architecture	211
24.8	Listen to Your Users	212
24.9	But Don't Listen to Your Users Too Much	213
24.10	Not All Users Need to Be Your Users	214

25	Removing Features	217
25.1	Do the Research	218
25.2	Inform Your Users	219
25.3	Provide Alternatives	219
25.4	It's Your Product	220
26	Learning from Video Games	222
26.1	What's Fun?	222
26.2	Why Your Product Is Not Like a Game	224
26.3	What We Can Learn from Games	227
26.4	Fun vs. Usability	233
III	Implementation	235
27	Guerilla Usability Testing	237
27.1	How Often to Test	238
27.2	Preparing for the Test	239
27.3	How Do You Find Testers?	239
27.4	How Many Testers	239
27.5	Running the Test	240
27.6	The Results	240
28	Usability Testing	242
28.1	Usability Tests Don't Have to Be Expensive	243
28.2	How Often to Test	244
28.3	How Many Testers	245
28.4	Who Should Test Your Product?	246
28.5	How to Find Testers	248
28.6	Different Types of Tests	248
28.7	Preparing for the Test	249
28.8	Running the Test	250
29	Testing in Person	252
29.1	Running the Test	252
30	Remote Testing	259
30.1	Moderated Remote Testing	260
30.2	Unmoderated Remote Testing	268

31	How Not to Test: Common Mistakes	270
31.1	Don't Use Words That Appear in the User Interface	270
31.2	Don't Influence the Tester	271
31.3	Avoid Stressful Situations	272
32	User Error Is Design Error	274
32.1	Don't Blame Your Users in Your Error Messages . .	275
32.2	No Error, No Blame	277
33	A/B Testing	281
33.1	When to Do A/B Testing	283
33.2	What's Success?	283
33.3	Preparing for the Test	284
33.4	Running the Test	284
33.5	Interpreting the Results	285
33.6	Keep in Mind	285
34	Collecting Usage Data	289
34.1	Measure Speed	289
34.2	Exit Points	290
34.3	Measure Failure	291
34.4	User Behavior	291
35	Dealing with User Feedback	294
35.1	Unexpected Uses	294
35.2	Bad Feedback	295
36	You're Not Done	297
A	Acknowledgments	298
B	Bibliography	301
	Index	304

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Homepage for Designed for Use

<http://pragprog.com/titles/lmuse>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/lmuse.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)