

# Contents at a Glance

---

*Introduction*

*xvii*

---

## **Part I Keeping It Simple**

---

- Chapter 1:** The UUI: The Useable User Interface **3**
- Chapter 2:** Modeling the Real World **35**
- Chapter 3:** Laying Out Your Windows and Dialog Boxes **61**
- Chapter 4:** Managing Your Software's Time **93**
- Chapter 5:** Highly Navigable Software **111**
- Chapter 6:** Data, Reports, and Printouts **131**
- Chapter 7:** Adding a Web Interface **155**

---

## **Part II The Lonely Engineer**

---

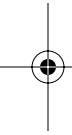
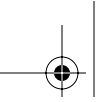
- Chapter 8:** Under the Hood **171**
- Chapter 9:** When Your Software Starts, Stops, or Dies a Quick Death **191**
- Chapter 10:** Modularity and Libraries **209**
- Chapter 11:** Object-Oriented Pontificating **231**

---

## **Part III The Business of It All: It's "Dollars and Sense"**

---

- Chapter 12:** A Very Gross National Product: Business and Software Problems **247**
- Chapter 13:** Testing, Testing, Testing **263**
- Chapter 14:** Installing, Training, and Helping **279**
- Chapter 15:** Book in a Book: A Guide for Programming Bosses **303**
- Appendix A:** Software Design Resources **321**



# Contents

---

*Introduction* *xvii*

<b>Part I</b>	<b>Keeping It Simple</b>	<b>1</b>
<hr/>		
<b>Chapter 1</b>	<b>The UI: The Useable User Interface</b>	<b>3</b>
	It's Intuitive! Trust Me!	6
	Idioms and the Software Experience	9
	That Stupid Desktop Metaphor	11
	The Visual Basic Programmer Has Been Let Out of the Cage!	13
	Giving Users What They Want (Including Respect)	16
	Software Libraries and Required Knowledge	17
	Asking Too Many Questions	18
	Saving User Preferences	24
	Remember the Keyboard?	26
	The Real Risk of Repetitive Motion Injury	31
	Moving Forward	32
<b>Chapter 2</b>	<b>Modeling the Real World</b>	<b>35</b>
	Modeling That's Too Real	36
	The Case for Using Use Cases	38
	Modeling Non-computer Gadgets with a Computer	43
	Defining the Scope of the Virtual Device	45
	Building the Virtual Interface	47
	Real-Time Interaction and Instant Updates	55
	Designing Classes that Model	56
	Moving Forward	59
<b>Chapter 3</b>	<b>Laying Out Your Windows and Dialog Boxes</b>	<b>61</b>
	Organizing and Arranging the Windows	63
	The Usability Differences between Dialogs and Windows	63
	The Great Debate: MDI, Tabs, Separate Windows, and Panes	65

Designing Tabbed Windows	69
Designing Split-Paned Windows without the Pain	71
Let Them Be Free! Custom Layout	73
Avoiding Modal Dialog Boxes When You Can	74
Keyboard Shortcuts and Windows Navigation	76
Laying Out Controls in a Window	76
The Purpose of the Controls	76
Arranging Your Dialog Box Controls	79
Don't Forget the Tab Order and Shortcuts!	83
Hint Windows and Balloon Help	86
Layers of Tabs in a Dialog, and Piles of Dialogs	86
More Window Design Issues	89
Big Fonts, Little Fonts	89
Adding a Status Bar Control	90
Custom Mouse Pointers over Your Window (No!)	90
Moving Forward	91
<b>Chapter 4</b>	
<b>    Managing Your Software's Time</b>	<b>93</b>
World, Meet Hourglass. Hourglass, World.	94
Alternatives to Showing the Hourglass	95
Implementing a Cancel Button	96
Don't Freeze Up the Whole Computer (or the Software).	99
Shortcuts to Success: Bookmarks	101
How Better Software Could Make People More Efficient	105
Don't Reinvent the Wheel!	106
Make Information Accessible	109
Moving Forward	110
<b>Chapter 5</b>	
<b>    Highly Navigable Software</b>	<b>111</b>
Avoiding Hoop-Jumping	112
Building a Minimal Required Usability Vocabulary	116
Navigation through Toolbars and Menus	119
Laying Out a Meaningful Toolbar	119
What to Put on a Toolbar	120
Constructing Sensible Menus	121
When to Use Images in Your Menus	124

Creating Reasonable Pop-up Menus	125	
Dragging and Dropping	128	
Including a Drag-and-Drop Feature	128	
Drag-and-Drop Technologies in Windows	129	
Moving Forward	130	
<b>Chapter 6</b>	<b>Data, Reports, and Printouts</b>	<b>131</b>
Relational Design: Great on the Hard Drive, but...	134	
Data versus Information	137	
Read It the Way a Customer Would	139	
Wasted Paper in Your Mailbox	145	
Spelling, Grammar, and Word Usage	145	
Remember the Usual Gotchas	145	
Avoid Talking Shop	146	
Making Data Available Online and on the Screen	148	
Putting Your Data on the Web	149	
Accessing Up-To-Date Information Online	151	
Including Print Features in Your Software	151	
Choosing the Best Menus	151	
Including a Print Preview	153	
Wasted Paper in Your Printer	154	
Moving Forward	154	
<b>Chapter 7</b>	<b>Adding a Web Interface</b>	<b>155</b>
Accessing Data over the Web	156	
Embedding a Browser (Yawn)	156	
Requiring an Internet Connection?	158	
Retrieving Data on the Web	160	
Auto-updates	163	
Automatically Updating Your Software Itself	163	
Automatically Updating the Data	164	
Synchronizing Data	166	
Moving Forward	168	

<b>Part II</b>	<b>The Lonely Engineer</b>	<b>169</b>
<b>Chapter 8</b>	<b>Under the Hood</b>	<b>171</b>
	Dealing with Dynamic Allocation	173
	Watch Those Pointer Variables	174
	Mucking with the System Directory: Keep Out!	178
	All Kinds of Mistakes Your Coworkers Make	179
	Tight Loops and Burning Up the User's Processor	179
	Invisible Programs Running in the Background	181
	Programs that Run upon Startup	183
	Writing Overly Complex Code	184
	Implementing an Undo System	185
	Moving Forward	189
<b>Chapter 9</b>	<b>When Your Software Starts, Stops, or Dies a Quick Death</b>	<b>191</b>
	Starting Up	192
	Minimizing Startup Time	192
	Reading the Command-Line Arguments Even in a Windowing System	193
	Allowing Multiple Instances of Your Program (or Not)	195
	Accepting Files and Opening Them While Your Program Is Running	196
	Shutting Down	198
	Correctly Using Exception Handlers	199
	Handle All Exceptions	199
	Clean Up Resources Appropriately	200
	What about Exceptions in Constructors and Destructors in C++?	203
	Specifying Exceptions in Function Headers in C++	205
	Moving Forward	207
<b>Chapter 10</b>	<b>Modularity and Libraries</b>	<b>209</b>
	Dividing Your Product into Libraries	210
	Static versus Dynamic?	210
	The Trouble with Dynamic Libraries	212
	What Makes for a Highly Useable Library?	214
	A Unique Naming Convention	214
	Loading the Dynamic Library Now or Later?	215
	Proper Handling of Versions	216

	Placing Libraries in the Correct Locations on Windows	224
	Properly Using Resources in a Multitasking Environment (Think: Mutexes!)	225
	Creating a Useable Library for Other Developers	227
	Moving Forward	230
<b>Chapter 11</b>	<b>Object-Oriented Pontificating</b>	<b>231</b>
	The Final Answer: What Is OOP?	232
	Avoid Letting Object-Oriented Programming Get out of Control	236
	Object-Oriented Usability	238
	Object-Oriented End Users	239
	Object-Oriented System Administrators	241
	Object-Oriented Developers	242
	Moving Forward	244
<b>Part III</b>	<b>The Business of It All: It's "Dollars and Sense"</b>	<b>245</b>
<b>Chapter 12</b>	<b>A Very Gross National Product: Business and Software Problems</b>	<b>247</b>
	Building the "Killer App"—Forget It!	248
	How Software Affects Businesses	253
	Technology Has Improved, but Has the Usability?	256
	Choose Ease and Usability, Not a Gazillion Features	257
	Being Aware of Industry Standards	259
	Ways You Can Defend Your Business against Problems	260
	Moving Forward	262
<b>Chapter 13</b>	<b>Testing, Testing, Testing</b>	<b>263</b>
	Why You Need a QA Team (and What They Should Do)	264
	Why Your QA Department Should <i>Not</i> Be a Stepping-stone to Development	265
	Who Should Work in QA?	266
	QA and Showstoppers	268
	Usability Testing Strategies	271
	Searching for Usability Bugs	271
	Doing a "Full Pass" Test on Your Software	274
	Performing a Consumer Usability Test on Your Software	275
	And Finally, Remaining Open-Minded to the Test Results	276
	Moving Forward	278

<b>Chapter 14</b>	<b>Installing, Training, and Helping</b>	<b>279</b>
	Installing Your Software	280
	What an Installer Does	280
	The Standard Choices	282
	Reinstalling, Repairing, and Removing	284
	Gathering Custom Options	285
	Gathering User Information	287
	Offering Online Help	288
	What about PDF?	291
	Training with Ease	293
	Choosing Types of Training	294
	Building a Highly Useable Training Course	295
	Setting Up (and Being Ready for) a Training Session	297
	Running a Training Session	299
	Moving Forward	301
<b>Chapter 15</b>	<b>Book in a Book: A Guide for Programming Bosses</b>	<b>303</b>
	Programmers, Decision Makers, and Egos	305
	The Kinds of Programmers	305
	Checks and Balances	311
	Placing New Hires	312
	He Said, She Said	314
	Making a Decision, and Staying Up-To-Date	314
	Keeping Up with the Technological Joneses	315
	Running Rampant: Managing Feature Creep	315
	Running Rampant 2.0: Avoiding the Latest Whiz-Bang Technologies	317
	Moving Forward	319
<b>Appendix A</b>	<b>Software Design Resources</b>	<b>321</b>
	Groups and Organizations	322
	Newsletters and Magazines	322
	Websites	323
	Books	324
<b>Index</b>		<b>327</b>