

The Art of Building Great User Experience in Software



Free Sampler

Effective UI

O'REILLY®

*Jonathan Anderson,
John McRee, Robb Wilson
& the EffectiveUI Team*

O'Reilly Ebooks—Your bookshelf on your devices!



When you buy an ebook through oreilly.com, you get lifetime access to the book, and whenever possible we provide it to you in four, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, and Android .apk ebook—that you can use on the devices of your choice. Our ebook files are fully searchable and you can cut-and-paste and print them. We also alert you when we've updated the files with corrections and additions.

Learn more at <http://oreilly.com/ebooks/>

You can also purchase O'Reilly ebooks through [iTunes](#), the [Android Marketplace](#), and [Amazon.com](#).

Effective UI

*Jonathan Anderson, John McRee, Robb Wilson,
and the EffectiveUI Team*

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

Effective UI

by Jonathan Anderson, John McRee, Robb Wilson, and the EffectiveUI Team

Copyright © 2010 EffectiveUI. All rights reserved.

Printed in Canada.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor: Steve Weiss

Development Editor: Jeff Riley

Production Editor: Rachel Monaghan

Copyeditor: Genevieve d'Entremont

Proofreader: Nancy Kotary

Indexer: Julie Hawks

Cover Designer: Karen Montgomery

Illustration and Interior Design:

The EffectiveUI Team

Printing History:

February 2010: First Edition.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Effective UI*, the image of a rainbow lorikeet, and related trade dress are trademarks of O'Reilly Media, Inc.

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 O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-0-596-15478-3

[F]

Contents

Preface	ix	3 Effective Planning and Requirements ..	75
1 Building an Effective UI	1	Uncertainty and the Unknown	77
Understanding UX	4	The Humility of Unknowing	78
What Good UX Accomplishes	6	The Weakness of Foresight and Planning	79
Why Engagement and Good UX Matter	10	Friction in a Complex and Peculiar System	81
The Elements of Engaging UX	11	Subjectivity and Change	87
Redefining Two Fundamental Terms	32	Lessons from Uncertainty and the Unknown	89
Design	32	The Further You Are in the Project, the Wiser You Are	89
Development	34	Start Development As Soon As Possible	90
2 Building the Case for Better UX	37	Written Functional Requirements and Specifications Are Inherently Flawed	90
Why Now Is the Moment for UX	40	Commitments to Scope Are Untenable	92
Motive	40	Relish and Respect the Unexpected	92
Means	48	Intolerance of Uncertainty Is Intolerable	93
Opportunity	50	Effective Requirements	94
Winning Support for Better UX	53	How Framework Requirements Are Built	97
Stakeholders	53	Reexamining the Three-Legged Stool	99
Education	57	Commitments You Can Live Up To	101
Quantifying the Business Value	67	Effective Process	102
Materializing and Proving the Concept	67	Development Methodology	103
Other Strategies for Building Support	73		

4 Bringing Together a Team	113	Who Should Be Involved in the Research	182
The Project Leader	116	Finding Research Participants	184
Relationship to the Product	116	Determining the Research Sample Size	185
Relationship to the Stakeholders	117	Making Recordings	188
Relationship to the Project Team	119	Research Through Speaking with Users	190
Who Should Be the Project Leader	119	User Interviews	190
The Stakeholders	121	Structured Interview Techniques	191
Securing Authority	121	Research Through Direct Observation	193
Collaboration and Decision Making	124	Analyzing the Research Observations	196
The Characteristics of a Successful Project Team	125	Discovering Personas	196
Getting Professional Help	127	Weaving User Stories	198
Insourcing Versus Outsourcing	130	Discovering User Priorities	199
5 Getting the Business Perspective	139	Guerilla User Research	200
Defining Success	141	Stakeholder Buy-in Through User Research	202
Creating a Project Mission Statement	142	7 Initial Product Architecture	205
Determining Project Success Criteria	144	The Initial Product Architecture Team	208
Exercising Restraint	145	Contextual Scenarios	210
Applying the Pareto Principle	148	Mapping High-Level Workflows	213
What Not to Restrain	148	Sketching Low-Fi Visual Representations of Requirements	215
Refocusing Product Objectives	149	Examining Key Features and Interactions	216
Omissions Aren't Permanent	150	Setting a Style Vision	217
Describing the Product's Users	151	Developing Nomenclature	221
User Attributes	152	Technical Architecture	222
Exercises to Identify Key User Attributes	153	Getting a Lay of the Land	223
Creating Business Requirements	160	Making Platform and Framework Choices	223
Defining "Requirement"	161	Understanding Data Requirements	224
Exercises to Develop Business Requirements	163	Mapping Interactions with Other Systems	225
Maintaining Stakeholder Buy-in	169	Finding Shortcuts Through Third-Party and Open Source Components	228
6 Getting to Know the User	171	Discovering Business Logic	229
Valuing User Research	173	Software Architecture in Big Design Up Front (BDUF)	230
Combating Pressure to Skip User Research	175	Project Infrastructure Needs	232
Key Concepts in User Research	177	Code Source Control	232
Empathy	177	Graphic Asset Management	233
User Goals Versus Product Features and Tasks	178	Testing Infrastructure and Environments	234
Qualitative Versus Quantitative Research Methods	180		

8 The Iterative Development Process . . .	235	9 Release and Post-Release	263
Regarding “Process”	239	Managing Expectations	265
Iterations and Feedback	239	The Alpha and Beta Releases	266
The Scope of Iterations	243	Receiving Orderly Feedback	268
Prioritizing the Subjects of Iterations	245	Last-Minute Housekeeping	269
Finishing Iterations with Something Complete	246	User Documentation	270
Estimating Iterations	247	And Champagne Corks Fly...	271
Basic Iterative Process	248	Adoption	272
Mapping Progress and Feedback Across Multiple Cycles	252	Post-Release	273
Increasing the Amount of Feedback	254	Review	274
Iteration in Sub-Ideal Project Approaches	256	Measurement and Tracking	277
Strict Waterfall Process	257	Afterword	281
Iteration in a Big Design Up Front (BDUF) Process	261	Index	287

specialized in one of these domains, you won't find yourself writing code, designing interfaces, or conducting user research, but understanding what to expect, what to avoid, and how all of the professional domains contribute to the forward momentum of a project will help you ensure its success.

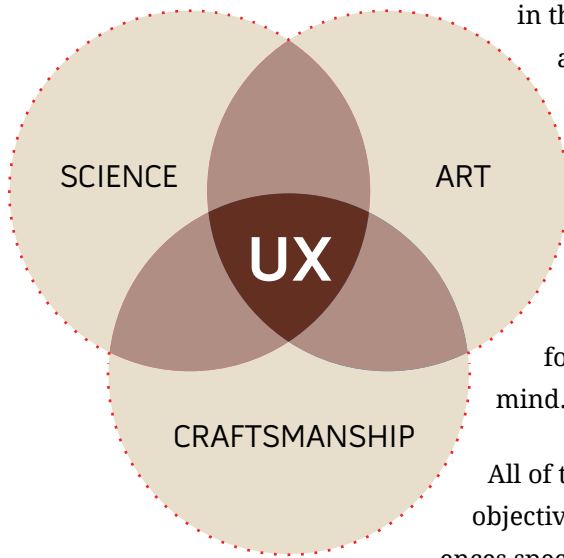
Understanding UX

Good and bad UX is typically easy to identify but difficult to define in generalities since the medium of UX is individual, subjective human experience. But in order to understand whether your company's products or internal systems have successful UX design and to convince skeptical executives of the value of UX, it helps to have a clear explanation of UX design and what makes its contribution valuable.

User experience is, as the name suggests, the experience a user has when interacting with software. Just as is the case with music, a software product's UX falls somewhere along a range between subjectively good and subjectively bad. This is obvious enough, but in that simple analogy are a number of truths that are often misunderstood or overlooked in software development. The process of creating good music involves a combination of the underlying mathematical principles of music that govern how we interpret sound, the technical skill required to write and play the music, and the artistic sense required to know how to make it all come together pleasingly

in the subjective consciousness of the intended audience. Take away any of those elements, and you make it impossible to bring new music into being. Also, the quality of music is not an objective one, but is specific to the subjective experience of the individual listener. A group of people might love techno and hate country, but that doesn't mean that techno is objectively good and country is objectively bad; it just means that if you're making music for that group, you need to bear their subjective needs in mind.

All of that is also the case in software UX. There's no such thing as objectively bad or good UX, only subjectively bad or good experiences specific to the user. The process of creating great UX involves some combination of quasi-scientific disciplines such as human factors



engineering, usability, and information architecture; the technical skills to produce not only great UX and user interface design but also the working software itself; and the artistic sense required to intuit and design for how the different subjective perspectives of different users will experience any given aspect of the software. Briefly, building great UX requires the combination of science, skilled craftsmanship, and art to address a subjective need.

In the way your company has approached the development or improvement of its software products, has it demonstrated an understanding of these concepts? Evidence of failure is easy to perceive in hindsight. If you've neglected the scientific aspects of building software, you've built products that are confusing, hard to use, cumbersome, poorly organized, and frustrating. Undervaluing the technical need on the engineering side usually means you've produced gorgeous UI designs but a disappointing, hacked, utterly compromised final product that performs poorly. The technical need on the UX design side—and yes, design for software is highly technical and not just subjective artistry—is also often overlooked or misunderstood. This leads to product UIs designed in ways that are graphically interesting but that cause undue difficulty in how the software will actually work and be developed. And finally, if you haven't recognized the subjective nature of UX, it's likely that, despite all the best of intentions and efforts, you've built products that users hate or reject. It also means you've worked with team members who narrowly focused on their own disciplines and deliverables without being constructively mindful of how their work assembles into a larger whole.

This entire book is dedicated to ways you can avoid those bad outcomes, but it's important at the outset to point out explicitly that delivering on the promise of great UX requires that you and your company's view of and approach to software development is sensible and correct. Just having some talented team members won't lead to success if your general approach to the endeavor is wrongheaded. And it's not enough to have just one person on the team who understands how things need to be done; this is knowledge that needs to be shared and needs to become part of a broader organizational competency. And so you'll find that most of the insight you'll gain in this book isn't specific to innovation, design, technique, or artistry; it's about how you can clear the way for innovation, design, technique, and artistry to come together successfully.

What Good UX Accomplishes

Having a strong UX in your software product is a good goal to have, but high-quality UX isn't in and of itself the real goal. It's the means to another, more important end that, though it's easy to appreciate firsthand, is incredibly hard to describe. Good UX enhances *user engagement*, and UX design is the art of creating and maintaining user engagement in software. Whereas UX is an abstract concept and UX design is a professional discipline, user engagement is the all-important subjective experience.

This naturally begs the question, what is engagement? This is best explained through analogies.

Engagement as immersion

The easiest, most intuitively obvious example of engagement in software is the experience of playing a great video game. Video games—particularly those of the first-person variety—aim to create a high degree of immersion for players.



Deep immersion occurs when the player becomes less and less aware of his surroundings, and his perception of the space separating him and the screen starts to fade. His experience of the game becomes one of being the character rather than just being a guy in a chair manipulating the controller. If you've ever seen someone leaning his body to one side to try to steer a car in a game or

dodge an incoming missile, you've seen someone who's heavily immersed in the game. Robbie Cooper produced a wonderful video for the *New York Times Magazine* showing just how immersed kids get in the game play experience: <http://video.nytimes.com/video/2008/11/21/magazine/1194833565213/immersion.html>.

Creating that deep immersion is an art form, and many things must be controlled lest they diminish or entirely break the immersive experience. A player

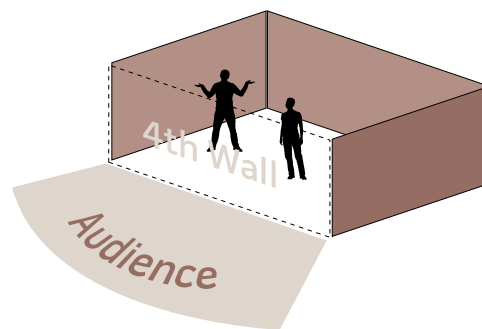
can be snapped out of immersion and the game play experience can be destroyed by simple problems like controllers that are difficult to operate, jarring inconsistencies in the game’s physics or rules, badly delivered lines by voiceover actors, or any jumping and skipping in the video or audio.

The example of immersion in gaming may seem quite remote from what you’re trying to accomplish. If you’re building a new Customer Relationship Management (CRM) tool for internal use at your company, for example, your goal in focusing on the UX of the product isn’t to make your sales team so enthralled by the experience of managing their customer interactions that they forget where they are, mentally merge with the application, and stay up until 4 a.m. trying to reach the next level of enterprise marketing automation efficiency. Well, maybe that wouldn’t be so bad. But certainly most software products are meant to be useful—not entertaining.

Deep immersion is, however, just an extreme example of user engagement. In the case of games, the goal is to bring the player’s focus away from manipulating the controls or comprehending the game dynamics, and even away from being aware of playing a game, and to put it squarely and deeply on goals internal to the game: winning the race, killing the aliens, solving the puzzle, and so on.

Engagement as the fourth wall

The *fourth wall* is a term from theater that is often used in filmmaking. The action on the stage is bounded by three walls, one in the back and two at the sides, but there is no fourth wall between the action and the audience. The audience members watching an engaging play infer and build that fourth wall in their minds, ignoring its absence. Just as the gamer loses awareness of the space between the screen and himself, and of the screen itself, the audience members become so engrossed in the action that the theater around them fades away. If an actor flubs a line, or a baby starts crying in the back of the theater, that fourth wall is “broken,” detracting from the experiential quality of the play. Rather than being engrossed in the plot and action, the audience members are suddenly reminded that they’re in a theater and have been sitting in their chairs for an uncomfortably long time.



Most filmmakers pay a tremendous amount of attention to the fourth wall. They attempt to keep the audience in a constant state of high engagement through the art of good filmmaking. The art of filmmaking helps them build and maintain engagement, and ensures that they avoid the simple little problems that break the fourth wall and remind the audience they're in a theater watching a film—like when the boom mic briefly appears at the top of the frame, or when actors or extras look straight at the camera, or when the special effects are noticeably fake or overdone. The filmmaker wants to keep the audience immersed in what's going on in the movie, and not on anything else outside it.

Engagement as frictionless accomplishment of goals

We're beginning to arrive at the heart of what engagement is: an undistracted, unencumbered focus on the ultimate goal of the activity a person's engaged in. In movies, as in video games, that goal is to be engrossed and entertained, to be carried away by a story and an experience. The point of software isn't necessarily to engross your users in the experience of using the software, it is to keep them focused on the ultimate goals they're trying to accomplish in using the software, rather than on the actual use of the software itself. Software is, after all, just a tool people use to accomplish certain goals. To be truly and unobtrusively useful, software must clear the straightest, most frictionless path to the accomplishment of the user's goals.

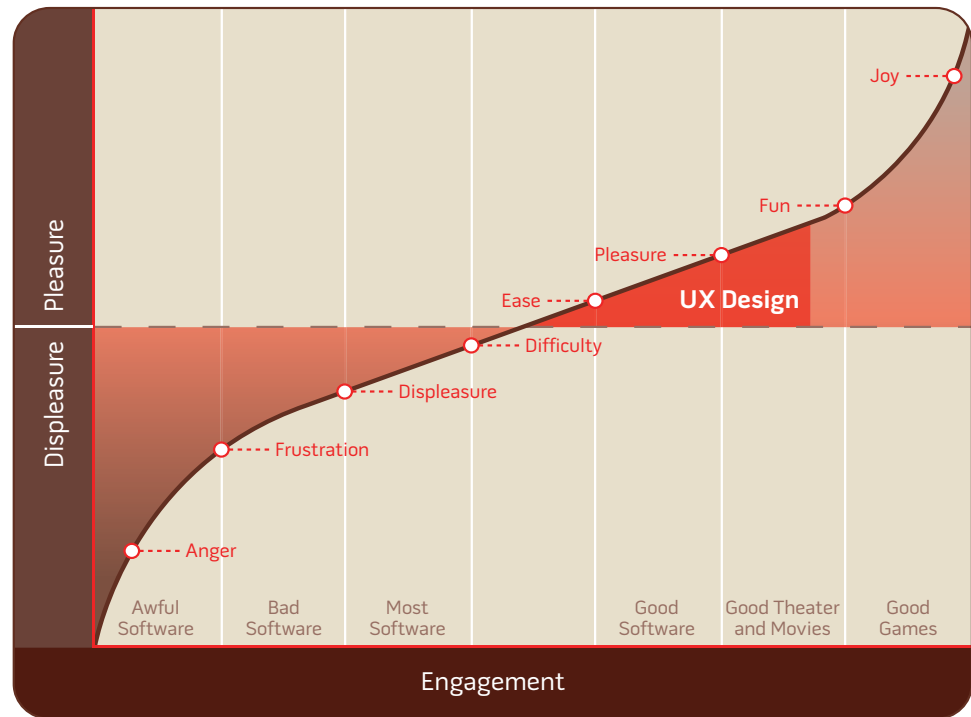
One of the most common instances of frictionless user experience that people encounter comes while driving a familiar route, such as from work to home at the end of each weekday. Almost everyone has had the experience of arriving in their garage or driveway with no memory whatsoever of the drive. In this case, rather than the product being software, it's the car, and instead of a keyboard and a mouse, the user is operating pedals and a steering wheel. The high degree of familiarity people have with the operation of the car allows for such a frictionless experience that their awareness of all the little tasks involved in driving slips away. On leaving work, the driver decides on the goal of returning home; the more familiar the route and the more skilled the driver, the less attention is required to accomplish the goal.

It's easy to imagine ways in which friction could be increased and attention drawn to the tasks involved in driving. Swapping the positions of the accelerator and brake pedals, for example, would shatter the driver's acquired easy familiarity with driving and would force her to pay very careful attention to working the pedals for the entire drive home. By changing the goal from going home to going to a restaurant in an unfamiliar part of town, the driver must focus her attention on navigation. And if something important in the car is malfunctioning—say, one of the tires is running flat—the driver will need to focus on controlling the steering wheel. Each of these will make for a more memorable experience of driving because the driver's attention will be on managing the little tasks involved in driving.

Engagement in software

The goal of UX design in building engagement in software is to help people be more focused on and effective at the accomplishment of their goals. This involves expert combination of the science, technique, craft, and art of UX design to create user experiences that effectively engage their target users. It also involves avoiding or smoothing over things that tend to create friction and diminish or break engagement. Breaking engagement, like breaking the fourth wall, is crossing the line where the user must focus on operating the software instead of achieving her goals. Broken engagement both causes and indicates difficulty for the user, which in turn causes displeasure. Strong engagement, on the other hand, both causes and indicates ease for the user, which in turn brings about pleasure.

The aim of UX design, with its principal goal of creating and maintaining engagement, is therefore to bring software past the point of frustration, difficulty, and displeasure, to first create engagement and then to deepen it according to the needs of the user and the aims of the product. UX design tries to reduce the friction that diminishes from engagement and that interferes with a user's ability to focus on accomplishing his goals. UX design works to apply a certain artistry that helps elevate simple engagement to higher levels of ease and pleasure, which are what make exceptional software.



Why Engagement and Good UX Matter

If you understand that positive engagement leads to greater pleasure and effectiveness for the user, and negative engagement leads to difficulty, displeasure, and wasted time, it's easy to imagine why engagement and good UX are important in customer-facing products and internal information systems. To ask whether good UX should be a priority for an organization is essentially to ask whether assisting and pleasing customers and helping employees to be happy and effective are important goals in business. If a software product has been well conceived such that helping users accomplish their goals is directly connected to an important business goal, then reducing the friction experienced in achieving the users' goals should be the same as reducing resistance against the accomplishment of business goals.

With the growth of the customer experience (CX) trend, there's been an increased recognition in business that every aspect of a company's interaction with its customers ("touch points") is an effective, rewarding experience. There's also an increased understanding of the importance of experience

quality over just service delivery. Simply having a well-stocked, conveniently located grocery store is not enough; the store must be visually appealing and clean, the checkout process must be quick and easy, and the store must have ample and accessible parking. The corollary to this in software is that it isn't sufficient to simply provide the user with a complete range of features; a good experience in using those features to accomplish one's goals is also required. The grocer doesn't want to waste his customers' time by not having enough checkout stands, or to trouble and confuse them by not organizing and labeling the shelves properly, or to deter potential customers by being hard to access or appearing unprofessional and untrustworthy. Likewise, companies with customer-facing products should avoid wasting their customers' time, confusing them or insulting their intelligence, or pushing them away. The linkage between acquiring and satisfying customers and business success is uncontroversial, but the direct relationship between UX quality and those goals is underappreciated.

The value of good UX and engagement extends to internal information systems and isn't limited to customer-facing applications. The goals change, but the means of accomplishing them remain the same. In the case of internal applications, exceptional UX has the ability to increase productivity, improve the timeliness and relevance of business data flowing to decision makers, increases adoption of the product and therefore the reach of its benefits, improves employee satisfaction, and generally reduces cost and increases opportunity.

The Elements of Engaging UX

EffectiveUI has spent a long time trying to define, in concrete and measurable terms, the substance of engaging UX. Since good UX is something that's measured subjectively and is dependent on the individual needs of the specific users of a given product, there's no 100-point checklist of good UX design; nevertheless, it's important to have a structure and lexicon for expressing problems and opportunities related to UX that otherwise can be recognized only at a gut level. There are a number of concepts that are focal points of good UX design, or can be fault points for bad UX. This list of elements of engaging UX can serve as an evaluation tool for assessing the UX quality of your company's current applications, understanding where past efforts have missed the mark, and identifying where investments are needed.

Want to read more?

You can find this book at oreilly.com
in print or ebook format.

It's also available at your favorite book retailer,
including [iTunes](#), [the Android Market](#), [Amazon](#),
and [Barnes & Noble](#).



O'REILLY[®]

Spreading the knowledge of innovators

oreilly.com