

# Discussions in User Experience

Healthcare for User Frustration

—  
Dave Lull

Apress®

[www.allitebooks.com](http://www.allitebooks.com)

# Discussions in User Experience

Healthcare for User Frustration



Dave Lull

Apress®

## *Discussions in User Experience*

Dave Lull

Glen Ellyn, Illinois, USA

ISBN-13 (pbk): 978-1-4842-3266-8

ISBN-13 (electronic): 978-1-4842-3267-5

<https://doi.org/10.1007/978-1-4842-3267-5>

Library of Congress Control Number: 2017960181

Copyright © 2017 by Dave Lull

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Editorial Director: Todd Green

Acquisitions Editor: Louise Corrigan

Development Editor: James Markham

Coordinating Editor: Nancy Chen

Copy Editor: Lori Lynn Cavanaugh

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/9781484232668](http://www.apress.com/9781484232668). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my sweet wife Edith, who has helped me in so many ways*

*As in my first book, this one is dedicated to all my educators, inspirators, and encouragers, who contributed to my understanding of UX, writing, art, and many other disciplines; to those who need training in correct UX design and design for usability; to the generations of UXers that need to be more certain that UX is being steered properly in business; to those who understand the value of experience; to UX teachers who it seems need to adjust the emphasis in some of their course content; and of course to all those who suffer with poor user interfaces, many of whom don't even know it.*

# Contents

<b>About the Author .....</b>	<b>xi</b>
<b>Foreword .....</b>	<b>xiii</b>
<b>Introduction .....</b>	<b>xvii</b>
<b>■ Chapter 1: UX in the Workplace.....</b>	<b>1</b>
Working in Agencies.....	1
Working in IT Departments.....	2
So Where Does It Fit?.....	2
Personal Interaction .....	2
<b>■ Chapter 2: UX and Society .....</b>	<b>5</b>
<b>■ Chapter 3: Some UX Philosophy and Ethics.....</b>	<b>7</b>
Graphics .....	7
Dark Patterns .....	8
Playing on Users' Emotions.....	9
<b>■ Chapter 4: UX Psychology Basics.....</b>	<b>11</b>
The Parts of UX.....	12
Basic Drives .....	12
Cognition .....	13
Levels of Processing .....	13
Definitions and Thresholds.....	13
Users .....	14
Industry Influence .....	14

■ CONTENTS

Negative Bias .....	15
Communication .....	15
General .....	15
User Communication (and Interpretation).....	17
Bias Yourself.....	18
Getting Ideas .....	18
Empathy .....	18
■ <b>Chapter 5: UX Psychology Tools .....</b>	<b>19</b>
Chromostereopsis .....	19
Curse Of Knowledge.....	19
Decoy Effect .....	19
Depressive Realism.....	20
Dunbar's Number .....	20
Dunning-Kruger Effect.....	20
Eye Tracking .....	21
Gestalt .....	22
Luminance Contrast .....	22
Phenomenology.....	22
Unlearning (to Get Closer to Our Users).....	23
Scotoma .....	23
■ <b>Chapter 6: Various UX Specifics .....</b>	<b>25</b>
Diplomacy and Tact .....	25
UX Levels of Understanding .....	26
Full Stack Unicorn .....	27
Specialization .....	28

Collaboration .....	28
UX vs. CX .....	29
Statistics .....	30
Minimum Viable Product (MVP) .....	31
Code Is Subservient to UX Design .....	31
A Link Is a Promise.....	31
Doing It Because You Can.....	32
What Does/Should the Business (Product Management) Do Concerning UX?.....	32
Best Practices .....	33
<b>■ Chapter 7: Planning.....</b>	<b>35</b>
General Product Planning.....	35
Responsive Design .....	36
Design for the Highest Percentage of Market .....	37
<b>■ Chapter 8: Design Concerns .....</b>	<b>39</b>
Clever vs. Usable .....	39
Trends.....	41
First Impressions.....	42
SEO and UX .....	43
Information Architecture .....	44
Customers vs. Business .....	44
Edge Cases.....	45
OTB (Not Off-Track But Still a Big Gamble).....	46
Policing.....	46
Context .....	46

- The (Quick) Science of Options ..... 47
  - Options in General ..... 47
  - Client Options ..... 48
- Average Person ..... 48
- Delighting Users ..... 49
- Designing the Flow..... 50
  - Flows in General ..... 50
  - Stepwise Flows ..... 50
  - Autoflow ..... 51
  - Ad hoc Navigation..... 51
  - Flow Buttons..... 51
- Overconveniencing..... 52
- Above the Fold..... 52
- Hamburger Menu Icon... Top Right or Left (Mobile)?..... 52
- Simplicity..... 53
- Proximity ..... 54
- Forms Design ..... 54
- Ecommerce Specific Issues ..... 55
  - Checkout Process ..... 55
  - Guest Checkout..... 56
- Chapter 9: Design Tools, Methods ..... 57**
  - General Labels..... 57
  - Column Header Labels ..... 58
  - Page Titles ..... 59
  - Button Labels ..... 59
  - Copywriting for Labels ..... 61
    - Internal and External Consistency ..... 61
    - Wall UI Displays ..... 62



First Idea Visualization (Initial Sketching).....	63
De Facto UI Standards .....	63
Futzers.....	63
<b>Specifics.....</b>	<b>64</b>
Scrolling and Columnization.....	64
Date/Time .....	65
Progress Visibility .....	66
<b>Progressive Disclosure.....</b>	<b>66</b>
Search Tools .....	67
Discovery and Hidden Affordances.....	69
Parallax Scrolling.....	70
Double-Duty Affordances.....	70
Skeuomorphs.....	71
<b>■ Chapter 10: Research and Testing.....</b>	<b>73</b>
<b>Problems with Research .....</b>	<b>73</b>
Time.....	74
Concision .....	74
Research Plan.....	74
Audience Spectrum .....	75
Eye Tracking Results Analysis.....	75
<b>Problems with Testing.....</b>	<b>75</b>
Test Creation.....	77
A/B Testing.....	77
Test Results Analysis .....	77
What Do We Test for Concerning UX? .....	78
Usability Testing Lab Environment.....	78

■ CONTENTS

■ **Chapter 11: Hiring UX** ..... 79

■ **Chapter 12: The Future of UX**..... 83

■ **Chapter 13: UX Tips** ..... 85

■ **Chapter 14: Notes on Online Discussions**..... 89

■ **Chapter 15: Summing Up: What Does All This Point To?** ..... 91

■ **Glossary**..... 93

■ **Bibliography** ..... 99

**Index**..... 101

# About the Author

**Dave Lull** is a UX Architect based in Chicago. He has worked for Rockwell, John Deere, Chase, Sears and Redbox, among others. His background as a software engineer, technical writer, web designer, graphic designer and instructor has helped him to realize the customer's welfare comes first. He has taught web design at college level and is hoping to set up a basic UX course at a first-year college level.

# Foreword

Since writing my first book on user experience design, I have felt a need in the UX realm for some deeper thinking, due to an obvious need for better understanding per the obvious confusion about UX in the workplace. I wondered why this was happening, and I have concluded there are mainly two things behind it: keeping your job (doing what the boss wants regardless of ANYTHING else), and the automatic adherence/acceptance of industry social influence (latest trends, sanctioned practices, talk-the-talk, etc.). Unfortunately much of this has a basis in some human primal issues like ego and pride, and an ongoing lack of definitions (evident everywhere). These obviously go beyond academia, as such analysis is beyond just UX mechanics and grammar and lives with them in the realm of work life. I don't intend to rip apart the current somewhat ragged fabric of UX practice in business (tears made mostly by how the business, too often with little understanding of UX, determines where and how it fits) but rather identify the issues and offer some suggestions to give it more integrity and even up the edges. UX needs some help *now* to better guide it into the future.

But what exactly is behind my fear of UX running off the rails? It is not something unique to UX. Over the last two decades I have noticed a change in thinking in the industry – actually not so much a change as an intensification of the old “first to market” and “fail fast” philosophies, which only work sometimes and for various lengths of time, and always leave litter along the tracks from speeding. Such philosophies tend to influence external ideas, such as absolute adherence to whatever the boss and team think, regardless of their expertise, and rather than trying to guide with what you know (and were hired for) just go along and find another job where hopefully people understand better. A sad state, and no doubt in existence for decades if not centuries.

People tend to think that whatever is generally accepted and becomes a popular word or concept is good – figuring, like the old marketing ads, “a million people can't be wrong” (they need to read up on Copernicus and Galileo). Worse, they tend to think it's actually true and “best practice” to the point of defending it. This may coincidentally occur at times, but how is that determined? Not over time, as we have seen acceptance before we even evaluate the idea, and we've seen best practices get replaced sometimes in just weeks. Not through study or comparison, for the same reason – just through popularity and so-called benchmark sites belonging to high-profit companies which naïve people assume, by virtue of company revenue only, must do everything perfectly, right down to their janitors' broom choice.

But carrying this further, what if some popular concept is wrong? We tend to wait until it is validated or rejected by the larger group. Meanwhile, we work as though it was correct, wasting time and money, maybe not only in the work place but in education, all because we had no real validation of the thing and we indulge the business who want to keep the latest trends until someone else risks a change. Take browsers, for example. How many man-hours across the planet have been wasted due to programming around

browser differences? Most of which didn't even have to happen? If you think about it, it's scary. If even a half day a month - 2.5% - of a developer's time is wasted, after multiplying that by all the developers in all the companies, the result is an astronomical amount of time (collectively millions of hours per year) down the drain. Not something for us to be proud of as business people nor as humans. So it seems that to get truly wise, we 1) have to examine each practice, and not propagate it purely out of social sanction or unwise business decisions, 2) support those that do in fact work, not just seem like they should work or "must" work because the label has become popular in the industry, and 3) pay close and constant attention to the standards of organizations that are attempting to help with this. But these often mean risk, which no one likes to take.

So how do we accomplish this? Thousands of companies testing the same practice just shift the time waste to a different area. I would suggest that some established group (of UX people with decades of experience) does extensive testing of a concept and passes it before everyone jumps on the bandwagon.

In any case, I just wanted to write down some thoughts about various ideas I've had concerning UX and IT (and the strange relationship between them) as well as UX in agencies and everywhere else. Not that the thoughts I relate or propose here are perfect ideas but rather to be a launchpad for your thinking. We need to make UX better, which in my opinion is simply finding ways to expel and keep out the wrong and inappropriate concepts that may fit better in other disciplines. In the decades I have been in UX, I have never been comfortable with how it was situated structurally in companies. Often it resides in IT under engineering, which is definitely not the right fit; it has also been under marketing with a similar feel, and product management which didn't seem right either. I asked myself why, and since UX is a psychological pursuit, that is, a "middle brain" thing (yes, I use left/right/middle brain terms as they work well, although they don't necessarily correspond perfectly with parts of the organ). Generally, engineers would like to box it into their strict rules world, artists tend to ignore usability in favor of trendiness and "cool" design. In my opinion, I would say UX doesn't fit under any current business group, but stands on its own. The reason it hasn't so far is because of thinking that says current structure must be right or cannot be changed, and through picture association - "if it is part of product creation, it must be under IT" and "if it doesn't fit under IT, what other department could it be in?" You can replace IT with Marketing in that sentence also. In other words, anything that doesn't fit under current structures must not be real, or at least not acceptable (again, the fears). In the pre-computer past, UX was around (better known then for the most part as human factors engineering), and was part of product design but also part of *every other production area*. It took the software engineering community (and the influence it had on company heads) to disregard this and try to stuff it into an area it didn't really belong. Remember that UX came about as a discipline because engineers couldn't design user interfaces. And, to be fair here, I've known many software engineers, developers, and the like, to openly admit they couldn't. So my suggestion is that it should exist as a department on its own, at least equivalent to engineering, marketing, and product management. An easy way to see this is to stop thinking it's only about product development. It applies to marketing/advertising ideas, branding, documentation, internal (software) tools, signage, and most of all to all company visual rules and general visual consistency throughout.

This book is not intended to discuss every facet of UX. I would say it covers less than half of the entire discipline. UX is a GIANT undertaking, not just another part of the product development process. Building and testing the UI is a different story as the UI is only part of the UX.

I'd like to reiterate that the field of user experience is vast and touches on many things. No one can know all about it and keep up with the nuances discovered so frequently, which in turn provide statistics, some of which are quite valuable to the field in various ways. Nor can we assume all UX concerns fit with new physical products/devices (although the fundamentals certainly do).

But the bottom line is this (and if you gain no other insight from this book, please learn this):

---

### UX IS NOT JUST ABOUT USERS – IT'S ABOUT USER WELFARE

---

No, not what the business wants (at least not directly), nor what your boss wants, nor backing yourself up with socially accepted truisms that fail in application either instantly or over time (especially based on the popularity of some individual or group), nor what *seems* right, nor what the company processes are, nor what your ego says in defense, and so on. And notice I did not say “whatever the user wants”, nor “consideration for the user” – there is an important difference here. Most UXers (and product managers) today believe that whatever the user says they want is what should be provided. Not quite right. It's whatever the user wants that accommodates their welfare. And those wants should be analyzed before diving headlong into designs that aren't really good or will be abandoned soon. The business end of the company (historically a restriction to good UX in my experience) will always opt for the trendy, easy to implement and/or whatever brings in profits in the short term (the frivolous wants, more or less). Therefore the best use of your time as a UXer is to learn true consideration of people. It pains me that this must be said, but I have noticed that particular property of the human waning over the decades. One way to verify this is to notice how uniquely great it feels when someone is sincerely (*sincerely*) considerate of you or others.

# Introduction

Einstein said that if he had an hour to solve a problem he would spend 55 minutes thinking about the problem and 5 minutes thinking about the solution. I think the same holds true about design – spend 55 minutes designing and 5 minutes implementing. Further, in that 55 minutes of design, 10 should be spent thinking about designing per the business requirements and 45 about designing per the user requirements.

This book is a how-to, but not how to build the house – rather 1) how to think about building the house before you even buy the materials, and 2) understanding the underlying anomalies and the deeper thinking needed before just using tools to assemble typical sites based on eye candy, cleverness and trends. UX requires a terrible amount of thought in design (and redesign), and a great understanding of what to do and especially what not to do. In other words, it requires the full realization of the true values of the different tools, methods, ideas, and analyses required to make not an award-winning site, but one that works well and the users like.

The UX room is entered into by many doors: psychology, art, development, and graphics to name a few. But regardless of which door (or doors) you came through, all the outer domains are represented inside, and are communal. UX is a psychological pursuit, and as such requires awareness and/or knowledge of a number of subjects as well as an open mind and a healthy regard for experience.

The ideas I'm presenting here may generally fit into current thinking/practice but some may not, so take them with grains of aspirin. My thoughts have broadened over the years to take into account things like responsible design (ala Victor Papanek) and thinking beyond the norms (ala Buckminster Fuller), all in regard to good aesthetics but more importantly all wrapped in the warm and classy overcoat of good usability. It seems that over the years I have seen parts of good usability tarnished by and even replaced with some popular ideas that should not have happened, such as:

- ☞ Trendiness and general popular UI design
- ☞ Benchmark competitor anything (pedestalizing high-profit competitors by assuming every little thing they do is right and will remain right for years)
- ☞ Decisions made by those in authority that have little UX experience (what *seems* right)
- ☞ In mobile, de facto “understood” controls/affordances (everyone must know them all when born)
- ☞ The idea of “discovery,” meaning the UI can have and hard-to-understand labels and icons and *even hidden parts* so users can work it like a puzzle

- ❖ Assumed universal UX understanding by all project stakeholders/team members regardless of their UX knowledge and experience (with UX decisions made by the group). (Note: I could hardly type this as I shudder to think of what I've seen happen)
- ❖ Listening and adhering, at all costs, to what the users say, as though users know exactly what they want and how to express it (and besides we can make more money on the fixes later)
- ❖ The idea that any of these can be used throughout an application and set of applications, regardless of where they really fit
- ❖ Everyone knows how to build the best testable pages and components. Popularizing the term “best test practices” is unnecessary.
- ❖ A/B tests are all that's ever needed and are the best possible choices (sometimes they are, but how is that determined, and who decides what they consist of?)
- ❖ Co-workers make good user representatives
- ❖ Being too close to the product doesn't matter
- ❖ You must hire UX people that know and have experience in the subject matter, not just in UX
- ❖ Development has its own separate QA department, but UX must test their own work (in most cases)
- ❖ Do whatever the user wants; take them literally and provide according to *exactly* what they say.
- ❖ Force-fit UX into engineering or marketing
- ❖ No need to question research; no need to test testing or question test creation

I hope to make it clear in this text that usability (that is, user *welfare*) is number one; aesthetics, trendiness, cleverness and coolness follow behind. I am a strong advocate of Donald Norman's statement “the best UI is the one that disappears”. And I'll add “disappears *quickly*”. I use his quote not just because of his popularity and great books, but because it stands on its own. I'm going to take a guiding position to help steer UX away from some of the nonsense that has surfaced, and expose the bad practices that have appeared especially since the mobile phone's small screen was created.

The basic overall goal of UX is simply to make it easy for users to accomplish their goals and tasks (if it somehow helps the business, all the better, but the user is first and foremost). This is clearly seen by examining the usability principles set forth by Jakob Nielsen and others. For example, using simple, natural language, minimizing the need to remember things, being consistent, and providing feedback, are all intended for the user, not the designer, not the business, not anyone else. But again I'll emphasize: it's not about just what the user thinks or wants, it's about the user's *welfare*. And if you can design in what they want and what the business wants as well, then you have the best solution. I feel a sad lack in our educational program for UX is the lack of emphasis on how to help users, not just do what they want. And yes, this would take training.



Bottom line: clean up first. Forget about UX specifics the business may want, forget about your own cleverness, and forget about impressing people – that is, the wrong people. Then determine who is important and in what way, and how to support them through general consideration, consideration of their welfare, and good usability.

## Conventions Used in This Book

### Deep Digging



This symbol means going beneath the surface of an idea to examine the foundation. Knowing the deeper meaning or reason of a UX concept can help us make better decisions when confronted with problems that aren't solvable using the standard tools and rules, or where a combination of rules is necessary and must all be considered together even when they seem to conflict.

### Soapbox



I've been in IT and UI/UX for more than three decades, and I'd like to pass on more than just the mechanics and grammar I've learned during that time – one of the reasons for this book. The psychology of activities in the daily experience and the reasons behind some UI concepts seem to be missed or underemphasized in education and even in books to some degree, but they play a very important part in our lives and the lives of our progeny, in the welfare of the companies we work for, and in our daily experience. I hope to encourage other authors to do the same, because I look for the little things, the daily learning, the reasons behind things that I can't find in other writings. We all can learn from the nuances as well as the mechanics.

I also feel a very strong need to comment on the more or less moral side of UX. As it is a psychological pursuit and not a simple science like mathematics, it can fall victim to manipulative efforts such as dark patterns (discussed elsewhere in this book).

But basically these notes are opinions and suggestions that have helped me create better products and benefit from a more pleasant daily experience. They may have also promoted better responsibility in design – at least I hope so.

## Definitions

Some of these definitions are textbook and can be found online. Some are made or augmented by my experience. You'll note some definitions in this book's text as well, which are ones I'm setting forth myself through a need in the industry.

## Affordance

A UI component that is (usually) a control (clickable, tappable, slidable, etc.). These can be textual hyperlinks, buttons, or things that stand out differently from their surrounding components in some way, and should be visible (there are some designs that have tappable areas with no visual clue of the affordance – see discoverability). See CTA.

## Cognitive Biases

These are biases we have that we are not necessarily aware of having but cause us to think, act and respond in certain ways that are not only subjective but subconscious. As UX designers and architects we can use these to guide users in certain directions, ***however, these can be powerful tools and using them must be done ethically. One reason for that, other than the obvious moral issue, is that some users can see through their use which destroys any trust they initially had in your company or site. The following link lists 67 different cognitive biases. If the link doesn't work just search for "cognitive biases".***

<http://www.neurosciencemarketing.com/blog/articles/cognitive-biases-cro.htm>

## Cognitive Hysteresis

This term is defined as the difficulty to change focus from one way of seeing or doing things to another, especially to the opposite, even in the face of contradictory evidence. Also called cognitive narrowing or tunnel vision, this phenomenon is often reinforced by ego – the reluctance to admit when we're wrong or unknowledgeable in some area.

## Cognitive Influence

Akin to social influence this phenomenon is defined as the subconscious need to conform to a way of thinking based on repetitive experiences, often fueled by social influence or seemingly correct ideas or methods, but influenced by more than that. For example, the more we use and hear about some concept the more we accept it as not only "good" but "the best" if we hear of nothing else comparatively "better." The waterfall method of software design was accepted as the chosen model for some time before being replaced with the iterative method. I saw that the iterative method was actually being used by myself and my colleagues before it was fully accepted.

## Confirmation Bias

Our predisposition to notice information with which we agree and ignore information with which we disagree. A good thing to remember when user testing.

## CTA or Call To Action

A term for those screen components with which we can interact. A UX jargon term, I feel the word affordance is a slightly better choice, as CTA suggests that any clickable/tappable visual is in the user's mind something to achieve a goal or continue a flow, when sometimes it is just a way out or an option choice. Yes these are all actions, but users generally don't consider a radio button for example a "call" but rather a choice - and since nearly no UX terms are kept only among UXers, I suggest we don't ask for trouble with terms non-UXers may misinterpret. See Affordance.

## Experiential and Reflective Cognition

Experiential cognition is the perception of phenomena understood intrinsically based on logic, obvious empirical data and/or experience knowledge and requires no conscious mental effort to interpret or understand. Reflective cognition requires thinking (reflection).

## HFE

Human factors engineering. The International Ergonomics Association defines ergonomics (or human factors) as "the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance."

## Innovation

The root word of innovation is the Latin "nova" which means new, in the purest sense. Any product or service that is truly innovative must be completely new, meaning there is risk involved as to its acceptance. Innovation is NOT something copied from another company - they had the innovation, so no one else can. It is also NOT something that requires society's approval - an idea is innovative or it isn't, regardless of social acceptance. It is also NOT something that is just cool unless of course it is in fact new.

## Negative Bias

Humans are negatively biased. What that means is we tend to think of the bad, embarrassing, negative things automatically without reflection, and that most good things require conscious effort to be brought to mind.

## Skeuomorph

A skeuomorph is a design or affordance that copies structures from real life counterparts, graphically representing their necessary visual and contextual (e.g., any obvious movement) characteristics in order for the user recognize the physical representation immediately.

## Usable, Usability

The degree to which a user interface is easy to use by all users or at least by most, or by a determined target audience.

Usability is a word that requires a descriptor, which, like “doctor” used alone, is usually understood as “good.” But anything is usable to some degree (such as in the case of something physical, as a paperweight). Even Don Norman’s teapot with the spout right over the handle is usable, just not very. One might go further to say its *handle* is unusable (but if you’re strong enough, and don’t mind scalding, you could use it to pour). Poor UIs are usable to some degree (unless the software is broken, which would not be the fault of the UI), and are described as having poor usability. Usability is a measure. You can’t say whether something is usable or not without saying to what degree, at least with a general adjective (“high usability,” “poor usability”).

## User Journey

This is the entire experience that takes place, from inception to completion (or to a premature end in some cases). Completion can mean the terminus of various paths. This is regardless of tools – or maybe a better way to say it is with all tools and devices involved. In other words, there are experiences that never touch an online device (or even an offline one), or may take place entirely in the mind (for example fiction writing). In this book of course I’ll limit this definition to the experiences that involve computer devices and/or online access.

An example of a user journey might be something like this. Ann wants to shop for (with intent to purchase) a new dress. This thought actually began, in this case, with the fact that she has been invited to a dinner with friends and “has nothing to wear.” So the inception was in this case the invitation. Her next steps are how, when, and where to shop. She decides to shop online at three of her favorite stores, but this was determined by the fact that she has no time to physically visit all three of these stores. She does her online shopping and decides Store A has the best chance to have something she likes. She also cannot spend the time having a dress shipped to her and then hoping it fits, and would like to see it on herself, so she goes to Store A. There she finds one she likes according to all of her requirements and buys it. At home she tries it on again and also tries her accessories with it. She finally decides to keep it and wear it to the dinner. But the journey isn’t over yet. She wears it to the dinner and her choice of clothing seems to go over well with the other guests. At home later she realizes it’s a keeper, and the journey, in essence, is over. All the parts of the journey need to be considered in the design of the website that 1) made her remember the store as one of the options, 2) helped her determine which store to visit, 3) helped her find a dress she liked, 4) encouraged her enough to spend time and gas to visit the store, 5) gained approval from her peers, and finally 6) decided to keep the dress.

Notice that there were a number of options Ann could have taken. She may have selected more stores; she may have visited more stores; she may have bought more than one dress to try on at home intending to either keep both or take one back; the dress she took home could have been defective in a way she didn't notice at the store, she may not have cared to keep it as others said it looked awful. The journey needs to consider all possible routes from all possible decisions, and this can get very sizable and/or complex. Avoid ruling out options that your design group considers limited in occurrence. A good test for this is to ask users about a similar set of initial conditions, and guide them against skipping over possible avenues they wouldn't explore.

## UX

Though decades old, a definition for UX has not to the date of this writing been clearly established. I'd like to offer a definition of this term based on decades of experience in the field including the online and offline discussions in which I've taken part. It is worth noting here also that UX is not just about digital presentation, but was around long before computers; in fact, it has been with us since Adam.

User Experience (UX) is an umbrella term that includes many areas. You cannot "do" UX, in the same way you cannot "do" Art or Information Technology. This is more clearly seen when we consider some of the functions of UX:

- User Experience Research
- User Experience Architecture
- (Visual) Information Design
- Usability
- User Cognition
- User Interface (UI) / User Interface Design (UID) / Interactive Design (IXD) / User Interaction and even User Experience Design (UXD, UED)
- User Experience Testing / User Acceptance Testing (UAT) / User Experience Acceptance Testing (UXAT) / Heuristic Evaluation
- UX Strategy
- Human factors engineering

UX, meaning "User Experience," sometimes referred to as "UE," obviously (I hope) has many facets and various ways to accomplish them. I have never felt that UX resides under HFE (Human Factors Engineering), but rather that UX sits on top, with HFE and HCI (Human-Computer Interaction) as subordinates.

Note that there is nothing in this list about development or programming (no HTML, CSS, Javascript, JSP, etc.). Although these affect the end experience in that they help to produce it, they are tools to build the site/app product, and their effect on the experience is usually second-hand or consequential (good design should not be limited by tools - the tools should accommodate good design - but granted sometimes a compromise is required). Similarly, graphic design is not included, although images (and all graphics)

play a very important part in UX. If we were to include everything that contributes to the experience, it would encompass every function of the business, and everything that touches the user's senses. Granted that UX should have the final eye on the product, but I would recommend a collaborative effort with development, marketing, and the like if for no other reason than to answer questions and guide the company's understanding of UX in the right direction.

Note also that UXD isn't here specifically, but it is by representation in other specified areas (UID/IxD, Research, etc. under HCI). This becomes easier to see with the realization that UX is a psychological pursuit, regardless of its nonpsychological components and regardless of its wrongly accepted place in the IT realm. You may use art elements and principles to create a web page, but your choices are based on psychology (or should be), not pure art alone. The same applies to the engineering ideas used to build back end access from a page. Anyway, the true idea of UX should not be confused with pure art or engineering disciplines; art and engineering principles, elements and tools are used in creating an experience of course, but UX is essentially applied psychology.

These branch into more specific areas, such as user satisfaction (through good usability and good aesthetics). Also, there are other subordinate concepts to consider, such as interpretation of what users really want (and need and expect). And, just as important as what *is* part of UX is what *is not* part of UX. UX is not programming or markup languages (although markup may be helpful to know, and it is good to know some programming concepts that can affect the experience, like Ajax calls which may cost a user time).

As far as what is needed to know in order to practice UX, besides the mechanics, grammar and psychology involved, I would say is as follows: UX is about really knowing how to put yourself in someone else's place. The emphasis in this sentence is the phrase REALLY KNOWING. Don't just assume you know, find out just how to really know what that means. Hopefully this book will help.

# CHAPTER 1



# UX in the Workplace

The type of work your company does can, in some ways, affect the UX you perform. This is not limited to agency versus IT, but strategy, architecture, design, responsibilities, and so on, can color your daily work experience differently from your experience in another company. Even permanent versus contract can make a difference.

Most of my work experience took place within the information technology section of companies whose existence was to create products or perform services that required software either directly for what was purchased, for internal departments that helped make parts or software for them, or for marketing them. I have also worked at agencies building images, web pages, mockups, wireframes, and so on. The two worlds are somewhat different, although what I produced was similar. Both required designing for usability, aesthetics, and business purpose. These similarities make sense because users are universal and don't care what environment the page they are viewing was made.

## Working in Agencies

My time in agencies was spent mostly building mockups, wireframes, and image assets for web pages. In working in IT departments I noticed stricter adherence to unwritten norms about design. Default locations for things were assumed and any purposeful veering from them was viewed as risk-taking mixed with a suspicion of a lack of experience. Apparently designers are supposed to wait until the grand high exalted mystic designers (from the supposed "benchmark" companies) step off the sanctified path first. I guess that's like designing women's clothing. I couldn't believe the fears and cowardice I found concerning the UI. The interesting part was when I found that something I tried to do was implemented just six months later. I realized I was ahead of my time, a lot less cowardly, and had a little better view into the future than most. Why are companies so fearful of that? You would think they would encourage it. Unfortunately design ideas must pass through stodgy, stubborn product managers (often with no UX training and/or no guts) causing the UX design area to circumvent that lack of understanding and either produce research such managers like (they never question that - interesting) or pre-test the idea and present that. The overriding sad part here is that the UX contingent is not trusted to know their own expertise. It seems that UX design moves forward about as fast as rust.

## Working in IT Departments

It became apparent to me after a few years in IT that nobody knew where UX should fit into an IT organization (and this may never happen!). In marketing, I had problems convincing others about usability when it went against business desires, such as what the business wanted to sell versus what the customers wanted to buy, and when I opted for good usability against less-usable UI trends. It worked pretty well in product management, but again the product managers were constrained by marketing ideas (at least where I was). The worst was engineering, where UX does not belong at all, partly for the same reasons as above, and partly because engineers will fight against (supposed) low-ranking UXers when their UI ideas cause too much development effort.

## So Where Does It Fit?

UX should, very simply and somewhat obviously, stand alone. It is not just a part of marketing, product management, or (gasp) engineering. There should be a UX director over all UX in a company, preferably a CXO (which some companies do have, I was glad to discover), if for no other reason than to have the power to make decisions for the users at the highest level and have them applied, within the satisfaction of the business. This UX area should be involved and have a say in all products/materials that go to users' or potential users' eyes (or ears, and so on). In a software production company, this means all documentation (even internal), marketing materials, signage, web sites, apps, kiosk signs and UI, branding, radio commercials, and even office labels should be designed by/with UX and have a final approval by UX. UX should be thought of as HAI (Human-*Anything* Interface), not just HCI. One added advantage is that simply by using a single area to oversee all such materials there will be consistency throughout the company and in promoting the brand in the eyes of the users and potential customers.

## Personal Interaction

Academia and books all prepare us for the fact that UID requires a very good understanding and effort in diplomacy and tact. I feel there is enough discussion and instruction needed on this topic to warrant an entire book, but what I'd like to point out here are the keys to this ability. I've found it easy to deal with many social situations in my work as a designer and architect by remembering the following things (and no, it's not always easy – *you have to practice these to make them work*):

- Trade emotion for thoughtfulness without being stuffy
- Choose your words wisely
- See the problem from everyone's angle



- People are various ages and maturity levels – be an example for those with less experience
- People have bad days – consider shelving the problem
- If you know the boss is wrong, get what is said in writing
- Clients don't know design (if they do why did they come to you?) – guide them carefully and be considerate and helpful
- Diplomacy and tact only work with sincerity

## CHAPTER 2



# UX and Society

UX, like any other discipline, relies to some degree on social involvement and understanding. Any social involvement is a give and take – that is, we as designers can't dictate, nor can users just rub the UX lamp for a genie. It means that we must defend this all-important discipline against the IT juggernaut rife with too much business-first thinking, assumption, haste, and misunderstanding. It is up to us to adjust designs according to a mix of users' needs and wants and what we, through our experience, know will not harm them, not bore them, and last longer than trends.

I have noticed over the decades that people, who thought everything was up to them, (or at least a collective social agreement) were labeled as arrogant or know-it-alls. Today the idea that everything is up to what we say and that definitions can change with current social thinking seems to be the norm, and that you are arrogant if you think otherwise! A case in point is “best practices”, which really means “whatever is popular right now”. Best practices today are not what they were (in some cases) even three months prior. I figure “best” things should last longer than that. But in our marketing-influenced society of planned obsolescence the kids of recent generations are brought up to believe that nothing lasts so reality is just the current time segment (what is that laughter coming from the bank?). While some things should change, others shouldn't, especially something like UX which is based on psychology. I've been listening for decades, and I never heard the sound of a paradigm shift there. Aesthetic trends, of course, come and go, but the basics of experience architecture (such as navigation and flow) aren't really subject to trend.

As mentioned elsewhere in this book, UX has not been universally understood as yet. The UI design part of UX is seen as just another step in the production process, and rightly so, but UX overall is about design of the experience and journey and therefore doesn't nestle snugly into a production process flow. It seems (business) people have only seen the UI side (understandably through UX slowly becoming known), and unfortunately think that is all there is to it.

So what can be done? I feel we have two options. One is to patiently wait while people gain an understanding and put UX in its proper place. Of course, that could take decades. The other is for UXers to take responsibility and help that movement along by offering ways to show the advantages of an overarching UX strategy. See the section on UX in the workplace for more thoughts on this.

Regardless of how we roll into the future, UX is here to stay which is not a small feat considering the odds against it. Thanks to pioneers like Norman, Neilsen, Cooper, Krug, and others we have successfully planted it firmly into the IT and agency worlds as the final competitive edge. Now all we have to do is make people aware that it is not just an edge.

## CHAPTER 3



# Some UX Philosophy and Ethics

What? Philosophy in UX? Sure - any psychological pursuit (in fact any human pursuit) has some level of moral involvement in its design or usage, sometimes by just adding or removing something. In any design there are ethics involved, as there are in any business discipline. Imagine graphics that are questionable or even those that use certain colors (colors can be used to manipulate the viewer to a degree). Manipulating the user mind, even in a minor way, has its ethical questions.

Decades ago, when the mail, door-to-door salesman, and the telephone were the main methods for a company to interact with customers other than store visits, it was quickly understood that prospective customers could be manipulated by smooth talking (and other forms of lying), which was surreptitiously labeled as “salesmanship”, a label that had the false overtone of being a good thing. This was the dawn of the silent war between the business and the customer, during which, in the initial skirmishes especially, many innocents fell (well, their bank balances did anyway). Thus the question of ethics surfaced. In UX, the user interface has become the main connection between a company and its customers (with brick-and-mortar store visits a close second) and the same ethical problems still apply. But now, quick retorts, word games, and body language have given way to manipulative visuals and flows on a screen. Often this manipulation is on a subconscious level, such as using certain colors and/or verbiage.

## Graphics

It has been long understood that images can evoke emotions and be used to sway a viewer’s opinion or make the viewer feel happy, safe, warm, hungry, and so on. One of the most obvious is using pictures of platters of food (not only on menus but in ads as well) with rich colors and maybe steaming, and so on, to affect appetite. Another quite obvious idea is pictures of beautiful slim girls in clothing products to suggest to customers what they might look like wearing them, because we all like to think (subconsciously if not consciously) that we look as attractive as the models.

Beyond the actual image, there are shapes and colors that also affect our interpretation. For example, a triangle or square on its edge or a trapezoid or rectangle on its long edge conveys steadfastness or sturdiness; a circle conveys movement, especially if touching a slanted line; an ellipse conveys movement with indecision. Lines provide separation, inclusion, direction, even movement with direction. Odd shaped polygons (depending on how close they appear as regular ones) may convey nothing or possibly confusion. Different shapes situated near or next to each other can convey more complex meanings.

Table 3-1 some colors and their general provocations.

**Table 3-1.** *General color psychology*

Red	Danger, attention, thought-invoking
White	Clean, virgin, new
Blue	Healthy, cool
Green	Natural, calming, flowing
Yellow	Competence, happiness
Orange	Warmth
Purple	Regal, authoritative, powerful
Black	Dark, male, classy, expensive

It is important to note that some colors evoke different feelings in different cultures, for example, white in Japan is associated with death. However, that doesn't mean white web sites are life-threatening there. Don't get confused with the cultural attributes of colors which can be different from the psychological and commercial ones.

Also notable is the juxtaposition of colors, especially with the same visual strength or amount, for an example red and yellow which are very often used for fast food signs (ketchup and mustard).

I encourage you to search "color theory" or "color psychology" on the web.

## Dark Patterns

Dark patterns are a great example of manipulation by the business. Dark patterns are interface gimmicks designed to trick a user into doing something, such as subscribing or donating, without offering options, without an easy way (or any way at all) to back out of the screen, or even just selecting the most expensive option as the default so fast users may mistakenly click/tap past the question inadvertently answering it to the company's benefit. Even bad ecommerce marketing schemes are not as malicious as dark patterns (and other such devices) because they don't manipulate from within the very tool one uses to shop. People of questionable integrity, whom you would think would not care (for themselves I mean), are included in the group who does not care for the burden of being used, having to think, and having their time taken up with nonsense. Worse, many of any

moral standing remember the negative attitude of the experience, and if there are many such devices, also remember the company that employed their usage.

Wouldn't it be better to accommodate the user, so they can remember the "good" sites? There are so many evil tricks being played on users in so many sites that those not using such techniques tend to stand out with a positive tone and be more remembered.

## Playing on Users' Emotions

Some sites use emotional leverage to presumably retain users or make them return. This could be considered a dark practice, as playing on anyone's emotions is a cheat. It also suggests that the company is insecure and/or greedy, since using such practices can tell a user that the company needs your business so badly, probably in view of the competition, that they are willing to use emotional tricks to keep it (while at the same time professing customer concern). This involves things like "sorry to see you go" and even worse "next time we'll do better to gain your loyalty" (suggesting you are not loyal by leaving).

The best way to gain and retain loyalty (in ecommerce, beyond good pricing and availability) is to provide a good experience, making the site both pleasant and easy to use, and putting user considerations first.

## CHAPTER 4



# UX Psychology Basics

UX is of course a psychological pursuit, but this is not easy to understand by looking at the tools and rules to build an experience and their association to the SDLC (software development life cycle). Many people get confused by the mistaken view out there that UX is simply another step in the process – an easy confusion if one considers the UI, just part of the overall experience, to be the entire UX. The full psychology of UX is at least a few books in itself, if not a full shelf of volumes. The following are those concepts that I feel are important to know as tools, ones that I've kept on top of my workbench in easy reach.

Through the years I've questioned what tools and understanding are really necessary to build a good user experience. The obvious requirements such as knowledge of psychology, design, understanding of interfaces, user types, and so on, are all needed, but what about those not so obvious? These would include empathy, experience with experiences, full user market inclusion (if not purposely market segment aimed), assessment of trends and their usage, weighing internal and external consistency usage, blind trust in research, and others; in short, wisdom in design as opposed to unquestioned conformity to ideas that are supported through simple popularity (such support not necessarily having any real substance). People wanting jobs will want to appear knowledgeable, and the simplest way to do that is to study up on current trends and drop buzzwords at interviews. This silently promotes bad ideas as well as good, as shown by concepts of dubious value like discoverability, over-animation, first impression eye candy, passé ideas like carousels, poor or incomplete research, and so on. Even something as obviously unethical as dark patterns is considered, through industry talk, to be an advantage of the applicant by less scrupulous businesses. Consider also how we all make quick overall judgments based on initial views, without considering our return to that page, possibly many times.

# The Parts of UX

The knowledge, skills and talents involved in UX are various. Not all items in the list below are required knowledge for all facets of UX, but it is good to know at least something about them.

- General psychology
- Cognitive psychology
- Social and organizational psychology
- Ergonomics and human factors
- Engineering (concepts)
- Some development concepts that affect the user experience (such as Javascript server calls within a page)
- Design, art, including history of same
- Anthropology
- Sociology
- Philosophy
- Semiotics
- Linguistics
- Artificial intelligence
- Computer science

I hasten to add items that don't quite fit into this list of specific disciplines, such as honesty, integrity, and sincerity. Such concepts can help in making good decisions.

## Basic Drives

I've learned that there are many connections between interactive activities and basic human needs. My take on our fundamental physiological drives (bottom layer in Maslow's hierarchy of needs) are 1) water, 2) food, 3) shelter, and 4) warmth. Some will argue that 5 would be social interaction, or that social interaction would be on the cusp between the first and other Maslow levels. After this bottom layer comes safety, belonging, esteem, and self-actualization. But throughout these, or maybe as a different but associated list, are pride, lust, and greed. They are often as powerful as the fundamental drives in some individuals.

The reason behind any UX interaction or design component can be boiled down to one or more of these items. For example, dark patterns are based on greed, colors and shapes can manipulate sensuality, even something as simple and innocent as designing for fewer clicks can be attributed to competitiveness and therefore possibly (but not always) to greed. Fascinating design ideas that ignore good usability would be attributed

to pride. Using sensuous pictures of course is manipulating through lust. There is a fine though a bit jagged line between being competitive and being too competitive. There are also underlying symptoms in the industrial strata, such as pleasing the boss (when you know better what should be done), and over-pleasing the customer (and higher-ups) based on quick or insubstantial research analysis.

## Cognition

**Experiential cognition** deals with things that are for the most part obvious, such as what is clickable and how to get to another page; things that are in our experience. In the larger realm of human factors an example could be an opening being a window or doorway. There is no thinking involved, no conscious effort expended. This can also cover the area of learned knowledge. To a remote, uncivilized group of natives a light switch on a wall has no correlation to bringing light (or anything else really), but to the rest of us we know what it is meant to control. This is something learned, however, not apparent or intrinsic.

**Reflective cognition** is that type which requires thinking. In Krug's book "Don't Make Me Think" this is exactly what is being discussed. I personally believe, like Krug and others, that reflection should be designed *out* of sites/apps completely (unless of course that's what the site is in some way about). There will nearly always be some ease of use issues without purposely ignoring that ease. If a user has to think, remember, or search the screen more than two or three seconds *then the UI has failed*.

## Levels of Processing

Our minds process information on three levels:

- Visceral. Subconscious, determined by biological heritage
- Behavioral. Mostly subconscious, basically learned skills
- Reflective. Conscious, self-aware part of the brain, home of self, self-image, analysis

The first two are experiential – experience from within our minds and experience we've learned in life. These would apply to shelter from the elements and other basic needs, response to stimuli, etc. It also accounts for visual grouping and mapping, tactile response, and color and grayscale differentiation, among other things.

## Definitions and Thresholds

Here is a topic that could fill a least one book by itself. So much depends on definitions and thresholds, and yet we avoid them in written and spoken conversation, assuming everyone knows exactly what is meant. This stems from concision (talk fast and avoid boredom) and pride (boasting of word/acronym knowledge; never apologizing). A good example is the term "user experience" which will be explained many different ways by many people, even just those in the IT or agency worlds, with varying component



inclusion. Components like testing, network knowledge, markup language knowledge or ability, programming knowledge, and more. The same is true of “usability” (see Definitions section).

The threshold problem appears when even if everyone agrees on a term’s definition they assume everyone agrees to what extent the idea extends, like “knowing html”, which could mean anything from the knowledge of what it is or does to the ability to create web pages with it along with CSS and even Javascript. Another example is the limit of de-facto mobile app standards we expect everyone to know, like swipe to delete and stylized menu icons (the “hamburger”). How many of these can we expect users to know before there are too many? The answer to that, especially when considering those new to the device, is one, because reflection is required which, as mentioned before in this book, means the UI has failed.

An example of a threshold issue is how far to go with a UI design. UI designers and graphic artists (as well as most non-UX people and departments) tend to push for more – more content, more graphics, more trend copying, and so on. UX architects and designers know to limit this because they know user psychology and because they understand that the product (UI) must be used long after the honeymoon is over.

## Users

As I mentioned before, and I don’t think I can overstress this, it is user *welfare* we are concerned about first and foremost. In business, the customer is most important; however, you can’t just ask them what they want as they’ll focus only on certain things, may not know how to state what they want, and may want something else by the time you implement the first want. User needs and wants is an interpretation of a discussion over time, not a quick laundry list. Even if 100% of those polled want a certain thing, that doesn’t mean it’s good for them (or good at all), nor that it will last any length of time. In one company I noticed that quick decisions were made on rather dubious grounds, and because there was no thought to the future the additions and fixes became a serious backlog, to the point where the user community was forced to experience the problems that were being shelved. Granted the business should have been aware of these, or made more aware of these, but if more thought was given to each issue in terms of forecast, much of the amassed overall problem could have been avoided. New ideas are important, but not at the risk of leaving old problems at large and piling up. See also the section on User Communication.

Many UX failures stem from blindly trusting what users say. This excludes (or precludes) whether they understood the questions correctly, are able to effectively communicate their points, if that communication is complete, if the communication says what it should and nothing more, if they are biased in any way, if their current context colors their thinking, if they missed anything, and a few other pitfalls. It would be great if there weren’t so many problems in human communication, but there are.

## Industry Influence

There are many external influences on UX work at all levels. I have noticed that some are insubstantial or regarded by false values. One example is to trust (without research) in “benchmark” sites based entirely on market share of those companies, not on sound

research of, in our case, the UI. I have seen this deter UX design from better usability in some cases (easier to sell to management and others because it's simpler to understand and explain). It has also caused contaminated business and UX processes. I suggest trusting in UIs that work, not what other company characteristics influence your thinking, and not who said something – don't avoid research just because of false elevation by a big industry name. To copy a business because of their popularity or profit numbers often goes *against* good UI practice, as well as innovation. Many businesses subscribe to the safe approach of seeing what others did first, and questioning (and avoiding) the very premise if not found. That's okay, as long as you don't think you are or want to be an industry leader. And forget the word innovative. Or maybe your entire charter is to play follow the leader and do nothing new. Based on what I've seen, not every risk will pay off well, but most non-risks won't.

## Negative Bias

We as humans tend to think and recall negative experiences much easier than positive ones. We tend to think badly of ourselves, others, and of situations without effort. We can recall bad experiences without trying. To think positively usually requires some level of effort. Because of this bias and its experiential nature we as UX designers and builders must take into account that first impressions must avoid negativity – the first thing a user encounters must not be negative because it will be remembered, and that can color the entire experience reinforcing that mood.

An odd anomaly of this is that in trying to be positive we can end up presenting as negative. For example, in the attempt to capture the users' eyes we may over-design something on the landing page that looks and feels good on first impression, but quickly goes sour, such as a wiki site that takes too long to load each beautiful animated page.

## Communication

Here is another topic that could be a book by itself, (and I believe there are some out there) communication between UX and users.

### General

The general process of communication is fraught with possible errors, whether written, aural, visual, verbal, and probably even mental (we would have to ask Spock). It is a continuum begun by an idea, translated to vehicle such as language (spoken or written), translated again from the vehicle, and finally understood by the reader/listener. Along the path are various pitfalls, such as volume, accent, age, colloquialisms, context, body language, noise, and more. Emails are especially susceptible, since spelling and proper grammar are not highly respected by all. Add to this the intended emphases and vocal inflections the writer hears, but the reader may not, and all of it being excused by things like “oh they know what I mean”.

How do we avoid such problems? Not easily (and that’s the reason they flourish). It requires reading and re-reading your text, reading it to someone else, using your audience’s ears and not your own, and so on. One suggestion is to wait at least an hour after finishing your writing and then reviewing it again. A better suggestion is to have someone else read it if possible.

“Projects are invariably composed of 10 percent technical issues and 90 percent human and communication issues” (Branaghan, 2001 - Jeff J. Rubin essay). The emphasis on communication is made apparent here. Rubin goes on to say that the disregard for this truth is why so many technology projects fail. We could go further here and say that the more left-brained people in management generally place too much emphasis on the mechanics, product, and methods, having less regard for the people and their daily experience. They tend to see people as just another component with all the expectations of that component outlined on their box (job description and tools knowledge) – a kind of “you’re all cattle with money” perception. No wonder companies that are “voted best to work for” emphasize human needs and desires, understanding that when these are addressed people will do their best work. I have personally experienced both extremes, the negative usually the fault of poor upper management, who see people and equipment as equivalent company components.

There is an important concept that isn’t hard to understand about human communication (see also Cobley and Jansz, 2005). In any spoken communication there are at least five basic parts: two on the transmission side, the actual conveyance, and two more on the reception side (Figure 4-1).

On the transmission side are the interpretation of the idea in the mind and the interpretation of that into a communicable thing. In the middle is the sending. The receiving side has the capture (hearing) and the interpretation of what’s heard into an idea and the understanding of the idea. Mistakes can occur at any node in this process because there is so much interpretation going on and because of the use of a spoken or written language (not to mention background noise, volume, distance, and so on).



**Figure 4-1.** Typical steps in any communication. Consider “language” as any type of implementation (like vocal sounds, sign language, or symbols).



Notice there is no guarantee that the original transmitted idea is the same as the received one, partly because of the steps in between, partly due to the intelligence and eloquence of the people involved, and partly because of possible problems inherent to the language used.

These steps aren't all conscious ones, but they are performed every time we communicate. This diagram is basic – it could be argued that the conversion to and from language may be further compromised by the speaking of the language (accent, dialect, and speech impediments) or the writing and reading of it. There is also the subconscious level of communication regarding word inflection, body language, colloquialisms, and so on. But the basics shown are enough to prove this point: many mistakes can be made at a mechanical level in the simplest messages. I'm sure you've had email experiences where something was misunderstood – email is one of the worst communication devices we have, not because of the concept, but because of how we abuse it by not taking the time to include the “feel” into our writing. This abuse comes from the desire to do things easier and faster.

The importance of communication in UI design is manifold. Interpreting users' requirements, explaining design to non-designers, and building an interface between systems and users are all examples of the varied uses of communication and the complexity of it.

## User Communication (and Interpretation)

Often we take what the users say they want or need and make decisions based on that, cold. This would be great if the users knew the following:

- What they wanted/needed at all times
- Exactly how to articulate what they want/need
- That the communication is clearly understood by the researchers, and so on.

In other words there are many issues with communication and finite, scheduled research times.

Customers:

...know much about what they think they want

...know much about what they don't think they want

...know much about what they think they need

...always assume what they want is a good idea

...know little about what they haven't thought of or been shown yet

So what can we do? First, understand initial research is only part of a first step, not the defining criteria. If you think about it, the real definition of what users want and need isn't clearly understood until the product has been used by users in their own environment, and for some time.

## Bias Yourself

As a UX designer, if I can't un-bias myself from my demographic, I need to over-bias myself into everyone else's in order to design for most users.

## Getting Ideas

Sometimes we just get stuck, either in trying to get an idea or in trying to resolve a problem. Some suggest surrounding yourself with inspiration, but I feel this can bias your design without you realizing it. Refer to other ideas, but avoid just copying what you like. I find the best way is to start with only the ideas and references in your mind, then later look beyond. Even this can be an influence, since your past experiences can taint an otherwise blank drawing board. That's not to say you shouldn't read and use your experience, just don't let it over-influence.

## Empathy

Empathy with your users is most keenly understood for accessibility considerations (such as for the color challenged), because it is easier to empathize with users who have only the challenges (if any) that we as the experience designers have. However, designers must still consider their users in a deeper sense of empathy, such as with concerns for time, usage conditions (like multitasking), environment, and other conditions (like noise) specific to the type of site or tool. Often we think only about usage in a quiet, warm, well-lit comfortable place, with a beverage handy, and we're wide awake and fed. Further, we assume we have the same or similar knowledge of the subject, the same background, and the same memory type and size. This is where personas come in handy, but of course only to a point. Still, better to have some clues into the users' minds than none at all. Empathy is basically walking a few miles (not just one) in the users' shoes.

The main difficulty I've seen with empathy is that the experience designer needs to empathize with a broad range of users and user types. This is what makes a great designer – the ability to design for the range without leaving anyone out. It is not easy, but it can be done. Often too broad a spectrum results in customization – allowing the user to make selections for his/her own type or level. Sometimes this can cost time and effort, but in my experience such costs have been worth it.

## CHAPTER 5



# UX Psychology Tools

There are many reasons people think and act the way they do. This section discusses some of the psychological tools that can be used in UX design. Some are necessary and some are just good to have in your toolbox. Like any tool, some of these can be used unethically. Don't.

## Chromostereopsis

Chromostereopsis is the visual effect of juxtaposed colors at the extreme ends of the spectrum “dancing” against each other. Avoid putting primary blue components on top of a primary red component (such as blue text on a red background). This actually hurts some users' eyes, but at any rate it defeats the purpose (as in text) and is annoying and therefore should be strictly avoided.

## Curse Of Knowledge

Curse of knowledge is making the assumption that others you are communicating with have the same background in the subject that you do or perhaps more. This can even go as far as having the expectation of the user understanding acronyms. This happens daily in any social contact. It flies directly in the face of UX design, as (at least less experienced) UX designers should always use empathy, with enough research to avoid wrong assumptions about users. They must learn how to communicate without judgment of user knowledge and at the same time without sounding condescending. In testing, it's never wrong to ask if the user has any questions about what was said.

## Decoy Effect

The decoy effect, or asymmetric dominance, is a psychological phenomenon (and a cognitive bias) that we often experience without knowing it. Businesses use this to bias the user into purchasing (or opting for) generally the most expensive of three choices or the choice the business most desires. The ethicality of such a practice is arguable when all

the options are real, and in some cases (where this translates to prices) may be prices set quite honestly with no biased intent. However, the purposeful use of this idea obviously has the best intentions of the company in mind and not the customer.

Consider these options:            A-\$25    B-\$45    C-\$50

In a nutshell, if there is one option that is inexpensive and two that are more expensive but not having the same difference in price as the difference in the first and others (thus the asymmetry), people will generally opt for the most expensive option because 1) the last two make the first less desirable, and 2) the highest is only a bit more than the middle one and therefore generally more attractive. The middle one is the decoy. It is a real option, but its purpose is to persuade the user to more strongly consider the second or third. For those who don't want to spend the money, but don't want the lowest price, the second is close enough to the third to give the company good revenue. For more information search "decoy effect".

## Depressive Realism

Depressive realism is a concept that suggests that depressed individuals make more realistic inferences than do non-depressed ones – that is, non-depressed individuals' judgments are positively biased and therefore usually less realistic (we are by nature negatively biased). I don't suggest that you depress your test users to achieve more candid results. Rather I suggest that if you are concerned about user honesty, a guide might be to list a user's demeanor along with other personal criteria you may record, and judge his/her comments and answers accordingly.

## Dunbar's Number

Dunbar's number is a suggested cognitive limit to the number of social relationships a person can retain – defined as those relationships where the person knows the others and their interrelation. This number of course varies by individual, but it has been found to be generally 100 to 230. Its importance in UX is for social media concepts (good "connections" per person).

## Dunning-Kruger Effect

Some individuals assess their abilities as better than they really are – an inability of the unskilled to recognize their mistakes. This is the Dunning-Kruger effect. A corollary to this is the idea that some high-ability people think that things they do easily are relatively easy for most everyone else (refer to [https://en.wikipedia.org/wiki/Dunning-Kruger\\_effect](https://en.wikipedia.org/wiki/Dunning-Kruger_effect) or search "Dunning-Kruger wiki"). This in turn suggests that such individuals feel that others should know what they are talking about in a given context. Since this effect is hard to recognize in the individual who has it, any conflict of knowledge can become

a battle of egos, or at least the D-K person's ego against a frustrated know-better. This is exacerbated by difference of rank – the bosses usually think they know or think they must look like they know.

So how do we deal with this? Exercise in tact and diplomacy, coupled with consideration without condescension – altogether not an easy thing to do. However, there are some tips in the section called Personal Interaction elsewhere in this book.

These things are good to know about users, as any one-on-one discussion may reveal such a bias and therefore color your test results. The experienced tester knows how to overcome this at the test – otherwise some adjustment must be made afterward (which is more often the case because you can see the issue in retrospect more easily than at the moment).

## Eye Tracking

After viewing many results of eye tracking on various pages, I've found the following attractions and their order fairly apparent. The eye will automatically go to these items on initial scan, assuming the page does not have many of the same item type. Note that this doesn't make a simple list. Note too, that some of these occur simultaneously or come on each other so fast that they can seem to occur almost simultaneously.

1. Animated items
2. Obvious high contrast, high luminance contrast or isolated items. For example, a white and gray page with only one colored component, or a page full of desaturated or small colored items with one saturated and/or larger colored or high contrast item. See also the luminance contrast topic.
3. Items that appear to have (obviously or inadvertently) some sexual significance or eye interpretation.
4. Universally recognized images.
5. Large high contrast colored areas
6. Images over text (especially colored images, large colored text will test higher than grayscale images). However, this must be within the context of #3.
7. Large high contrast text over small images and other text.

Testing with eye tracking is of course generally viable, but there may be other tests where your money would be better spent once you have eye tracking and heat map analysis experience. It pained me to see eye tracking expenditure in one company when experienced UXers could easily have guessed where the eye would go. When the business requires eye-tracking data in lieu of their UX experience, they should learn to trust the experts first to determine if and when tracking is necessary. Beyond this, I've also seen that analysis of results is assumed to be clearly understood by everyone without any training whatsoever – that we all know exactly what to do when we see a heat map of a page. There are many ways to maximize where the eye goes by manipulating page



components to guide it, but in general I haven't seen any great improvement in the site purpose (knowledge, purchase, info finding, and so on). However, there are some benefits, such as the user's ability to accomplish their goal a little more quickly and with a little less frustration – and every little bit helps.

## Gestalt

Gestalt basically means a whole that is perceived from disconnected parts through, in our case, a visual association. Our brains want to make something complete and understandable to fit within our experience. Four right angles, for example, aligned to look like corners of a box, will be perceived as a rectangle with parts of its sides missing.

Where this comes into play for UI is somewhat obvious. Most people will see figures and polygons using screen components that have no real correlation other than their shapes and placements. Even colors can cause correlation to complete an image. You probably don't want that to happen, when you are purposely trying to group what you want seen as a group. This is just one more reason to have someone else look at your designs before continuing with production.

## Luminance Contrast

Luminance is the intensity or perceived brightness of a light. Luminance contrast is the difference in luminance between a screen component and its surroundings. There are three concerns here: the component's immediate or nearby surroundings, its overall surroundings over the page, and the ambient light in the room or area. The main concern though is in the immediate surroundings, such as a button's text to its background. Of course, the button background is visually compared to *its* immediate surroundings, usually the area the button resides in which may be some page part or possibly the entire page.

It's important to note here that luminance contrast is not the same as simple dark/light contrast. Luminance contrast deals with color hue values. For example, the luminance of a primary saturated green background may not contrast as well with black text as it would with white or a saturated light green text. NASA has an excellent article on this topic at: [https://colorusage.arc.nasa.gov/luminance\\_cont.php](https://colorusage.arc.nasa.gov/luminance_cont.php) and [https://colorusage.arc.nasa.gov/guidelines\\_lum\\_cont.php](https://colorusage.arc.nasa.gov/guidelines_lum_cont.php).

This tells us not to depend on size: often luminance contrast is higher with a smaller button that is the same color as a larger one because of their respective surroundings.

## Phenomenology

Phenomenology is the objective study of those topics usually regarded as subjective, or the study of appearances. The definition of phenomenology is somewhat ambiguous, as it has been discussed and segmented (logical and philosophical for example) since its inception (late 1700's). Supporting Edmund Husserl, known as the father of phenomenology, this quote sums up the definition issue: “[any] unique and final

definition of phenomenology is dangerous and perhaps even paradoxical” due to the nature of the study (see [https://en.wikipedia.org/wiki/Phenomenology\\_\(philosophy\)](https://en.wikipedia.org/wiki/Phenomenology_(philosophy))).

In UX, phenomenology brings to light that to most people things are what they seem to be, or perhaps *should be* what they seem to be, without hidden meaning. This falls right into our dumpster, as simplicity and ease of use are paramount to us. We have to consider what users think about what they see and what they can do, which is most often, if not always, what it *seems* they can see and do. This is as opposed to what we as designers and architects understand the interpretation of and know can be done – we often can’t help but think more deeply than our users. The most difficult part, in my estimation, of UX, is the true ability to see through others’ eyes without one’s own internal biases slanting the importance of some interpretations. It’s definitely not an easy thing to do – it is hard to unlearn things, especially those deeply embedded in our psyche. In fact I would be tempted to call it a talent rather than a skill. But it is not impossible. In any case, we need to think in terms of what is in our experience and disregard anything deeper.

## Unlearning (to Get Closer to Our Users)

In order to design visuals and flows, we need to strip away what we know and make ourselves think more as our users would. This includes the grammar (our understanding of navigations, standards, page design, and so on) as well as the content in some cases. It also includes the responsibility we have of the site design. In other words, disavowing all understanding that a typical user would not have.

But this is not an easy thing to do and not easy to gauge. How do you know when you have unlearned enough? One answer to this question is to realize and record your feelings, knowledge bits, and lack of knowledge when visiting an unfamiliar and dissimilar site.

An idea I’ve used for years to help accomplish unlearning is clearing my mind. I use this in steps. Step one is to free myself from any memory concerns for the day. I do this by writing down all the things I want to do that day (work and home), including things I want to discuss with someone or even just think about before the day is out. Then I do a simple prioritization, first taking care of any items that are quick and/or need to be done right away. Step two is to start the work with a focus on the tasks that need design (redesign, completion, review, and so on) just as a user would be focused, keeping myself intent on the task from start to end, and anticipating interruptions, but quickly regaining focus if interruptions occur. Within this step I keep a piece of paper handy to jot any mental interruptions, including epiphanies, ideas about the design itself that may require further research, or some other task requiring some future work that might be too involved for or not necessary to the task at hand.

Whatever method you use, a good test is to see if you have a somewhat similar feeling entering your site as you had entering someone else’s.

## Scotoma

A physical scotoma is a blind spot in the eye. A psychological scotoma is a consequential or manufactured blind spot in seeing or understanding something – in most cases because we see only what we want to see. This applies to many things, but as far as UX is

concerned there are some concerns we should be aware of: 1) we experience our physical (eye) blind spot when viewing a web page (the location depending on distance from the eye to the screen), which may cause us to miss something on the page, 2) our eyes may be challenged concerning certain colors, and 3) we may be purposely or inadvertently (over time) blind to certain colors or images (such as ads that always appear on the right or banners we tend to ignore because we're not interested in advertising at the moment).

The business and design areas in a company should be constantly aware of this phenomenon. Besides having their ads ignored, such a practice can cause users to remember the ubiquitous annoyance and add fuel to ignoring parts of a page or even to site skipping.

## CHAPTER 6



# Various UX Specifics

Decades in the trenches have taught me to realize that definitions often have a depth not realized by all and that topics may have a breadth of knowledge and detail. This section presents specific UX concepts that are either necessary or good to know.

## Diplomacy and Tact

Academia and books all prepare us for the fact that UID requires a very good understanding and effort in diplomacy and tact. I feel there is enough discussion and instruction needed on this topic to warrant an entire book. But what I'd like to point out here are the keys to this ability. I've found it easier to deal with just about any social condition in my work as a designer and architect by remembering these things:

- Trade emotion for thoughtfulness without being stuffy
- Choose your words wisely
- See the problem from everyone's angle
- People are various ages and maturity levels – be an example for those with less experience
- People have bad days – consider shelving the problem
- Clients don't know design (if they do why did they come to you?) – guide them carefully
- Diplomacy and tact really only work with sincerity

It is very important to understand that, with using tact just as with anything else, you can go overboard. Your entire argument can be lost, for example, by overplaying tact to the point of sounding too knowledgeable or condescending.

## UX Levels of Understanding

I noticed over the past decades that my understanding of UX – how it works, how it functions in the workplace, what I learn, and so on – has changed. It may be that *I've* changed, or just that I've learned more from my work, or maybe it's a combination of these. The idea of the levels I'd like to discuss may certainly apply to more than just UX, but I want to share some of the things I've observed about this.

I have determined at least four different levels, shown in Table 6-1, characterized by labels that I hope concisely identify and differentiate them. There may be more, but my guess is these cover the majority of possible categories.

**Table 6-1.** *UX Levels*

Label	Level	UX Exp (years)	Explanation
Dangerous	Surface	0	Applies to anyone and means to them that UX is simple, not as difficult or important as software engineering or business needs/wants. What seems right is right.
Educated	Educated	4	Applies to those who have a UX-based degree and/or have had some courses and read the books.
Immersed	Resident	10	Applies to those who are not only educated but have worked in UX for some years, preferably in different organizations (for a well-rounded education of the industry, having learned how people think).
Sagacious	Professional	20	Applies to those who have learned to go beyond the teachings and accepted practices, who understand the psychology of the practices, how they compare and conflict (and how to resolve the conflicts), why they do what they do, which ones are or aren't understood in business, how UX concepts are related to basic human drives, and how to get the point across to non-UXers.

Note that the difference between the levels is basically experience. It's not education, because most will relate that to academia and degrees, which really only applies to attaining the second level (with the possible exception of a PhD, which requires enough hands-on work and study to gain experience).

Just because everyone agrees to something doesn't mean it is better or the most correct. I think Copernicus and Galileo were the best historical examples of this idea, as well as a great example of dedication to one's convictions. Galileo was put under

“house arrest” (commuted from formal imprisonment) because he dared to think, like Copernicus, that the heavens did not revolve around the earth. After further proof and the support of others (the Age of Enlightenment) people slowly accepted the idea. This slow acceptance was a paradigm shift which took many decades before the populace realized they were all wrong.

## Full Stack Unicorn

A full stack unicorn is a term that basically means a person that can do many things, and in the case of UX, things that are UX and some things that are not. An example would be a UX architect that is also a graphic designer and a developer. While not impossible to find, many people know that this requires all sides of the brain (engineer, artist, and psychologist), which *usually* means the individual will not excel in all segments but rather, through preference for one discipline, will excel in only one and have less interest and therefore do a comparatively less quality job with the others.

The term came about from naïve and/or thrifty business people that either didn't know what UX and development were (it's all front end, right?), or simply felt they couldn't afford three people to do such “less important” stuff. After all, “front end” entails everything to do with what the user sees, right? Of course not, and it took decades for business people to realize that it doesn't (and some still don't see it).

It is perfectly possible for a person to do various things and well, but those individuals probably like one thing a bit better. This is referred to as “how we're wired”. However people like that are very hard to find, so make sure to test the horn – most of them are painted cardboard – pretty and maybe sturdy but not real. Also consider: should a position actually be responsible for all that even if the person in that position can do it all and do it all well? Will there be too much work? And won't he/she be hard to replace?

Hiring managers have varying thoughts on this. What is difficult for some to understand is that the list shown here can mean four different positions: UXA, graphic designer, front end developer, and back end developer. And while some individuals may be able to do all this, our brains are not wired to excel in all these areas at once.

**Full stack**, as it applies to UX, *generally* means having knowledge and experience with:

- UX and UI design
- UX architecture
- UX strategy
- wireframing
- graphic design
- research
- testing
- front end coding (HTML, CSS, Javascript, and JS libraries)
- possibly (but rarer) back end coding (like Java)
- possibly (but rarer) connection speeds & hardware anomalies

You may notice that I avoided “left brain” and “right brain” references here. This is because many believe the notion is wrong, basically because they associate these types with actual brain lobes (left lobe, right lobe, Michelob). I myself don’t want to confuse the ways we think with brain parts. After 40 years in the business (IT and agency) world, and through basic life experience as well, it is glaringly obvious that there are at least three distinct basic ways of how we think and do: call them the engineering part (but I don’t just mean engineers here – it’s an umbrella term), the artistic part, and the psychology part (EAP). It was convenient to associate these with brain parts, but in the end it was confusing to some because they’re not a perfect fit, and it was too easy to end up letting the tail wag the dog. Maybe these three parts are too fundamental, but this idea has lasted against years of usage. I recommend using these brain part references, or my EAP reference, and to not confuse the terminology with the concept. As I say about many such terms, *as long as everyone knows what you’re talking about, it’s not a big deal.*

## Specialization

Decades ago, businesses shifted their requirements from focused, single-function positions to generalized ones. While the upside (to the business) is paying for two or more positions with one salary, the downside is the thinning of each expertise as reduced by time allowance. The result is what you see prevalent on the web – compromised designs done by trends and cleverness and quick thinking. Isn’t it obvious that people doing the part of their jobs they don’t care for will cut corners there in favor of doing what they like? And how much more effort and thinking will they put into that side? But to be fair, often such individuals feel that having experience in different disciplines will help them keep and land jobs in the future, which is true. But unfortunately this is building the wrong structure – one that promotes the multi-talents (whether they excel at all parts or not) which are harder to replace, rather than those in focused positions that, as I have seen, can create better products without compromising talent. It may be nothing more than the concentration necessary to do better in all the areas, but that takes time. Another concern is that design may suffer against the desire to keep all parts simple, such as foregoing some design idea in favor of simpler code. The opposite is true also – making the code complex and taking less time on the design.

Bottom line to managers/chiefs: don’t sacrifice quality in a discipline on the assumption of saving cost – in the long run nothing will be saved, unless you’re satisfied with low- to mid-quality, uncompetitive products in constant need of update.

## Collaboration

The definition of collaboration is for a group to work together to realize or achieve something successfully. It doesn’t mean cross-function, control, compromise, or delegation. It requires the ability for all in the group to understand everyone’s part and to try to accommodate them while displaying trust in each others’ expertise. For two team members with the same expertise of course this is different – they share responsibility and work.

The members of a project team need to collaborate with each other. However, we can't assume everyone knows what that means. In my experience, this involves making sure all parties that should know something do know it, while understanding and remembering who is responsible for what; and making sure those of the conversation group all understand the definition of the term.

## UX vs. CX

UX has evolved quickly, I would say too quickly, and as with any fast-moving concept we have made ourselves suffer by ignoring the clarity of its nomenclature. We need to get the definitions solid (and stop making new words all the time without clarifying the existing ones). Customer eXperience (CX) is one of these. All customers are users at least within our digital products. All users have a User eXperience, therefore all customers are UX users. That said, the CX term pertains mostly to commerce (ecommerce for digital concerns), and therefore is a simple shorthand to mean those users that are customers. But, is it really necessary to make a distinction between all users and ecommerce users? If that's true, why not have WX for wiki users and DX for document users (like for government docs)? There are differences in these as there are specific user goals that differ in all these types. And all are users.

The typical semantic and more universally understood definitions need to hold unless we can re-educate everyone quickly, which would require that the UX world is in agreement first. We all know the best way to solve this is to test it in public, which of course costs time and money. From my experience I would have to say that most will consider UX as the umbrella and anything with "customer" in it as an ecommerce-oriented subordinate.

However, I see that UX to some is HCI (Human-Computer Interface) only. The interaction with a device is not necessarily the entire experience or journey, but is it universally understood that "customer" could apply to a full journey of any kind with any goals? Those in ecommerce consider both "customer" and "ecommerce" to have understood goals like shop, compare, purchase, and so on, which are not goals that apply to every site and app in existence. Using "customer" broadly comes from marketing thinking (which has brought us such dubious terms as "free for" and "marketplace").

Let's look at the words. There seem to be two ways of thinking about these based on definitions: if any access to any kind of site/app, even non-ecommerce ones, means the person is a "customer" of it, then CX seems to apply the same as UX; if only ecommerce sites have customers, meaning they shop, compare, purchase, and so on, and which I believe is the more generally accepted definition, then UX is the umbrella and CX a subordinate. I think we're also confusing the HCI part of the journey with the entire journey, in which case I can see why some think the digital "user" is only a part. My contention is that we need a term for the user journey from inception to conclusion, and I doubt it should be "customer". That term is too restrictive and marketing oriented and can further confuse some.

When personal computers were in the larval stage, there was no such thing as ecommerce, but there was commerce and there were customers. When computers matured and the internet era began, ecommerce was born and users also became customers of the sites that offered products and services. A better way to say this is that store customers had a new way to shop for and purchase store offerings besides visiting or phoning.



Further, consider this: in ecommerce, users are always considered customers or potential customers, but in non-ecommerce, the best word so far to describe users is “visitors”. We can’t think of visitors as customers because of the definition of the word – a visitor is not necessarily a customer. A visitor might be a customer of the site in a sense, but not of the store. Until that visitor purchases something, he/she is only a potential customer. It depends on how you define the terms. The term customer generally implies shopping, comparing, buying, and so on, not things one can do on every site. The word visitor does not imply a purchase was made by the individual. You may have visited a store (physically), even many times, and never bought anything – would you consider yourself a customer?

Then there is the lack of definition of the word customer at the point of when a potential customer becomes an actual customer. Is anyone that visits your ecommerce site a customer, regardless of whether they bought anything, or do visitors graduate to customer status after a purchase? Note that your development team or back end information architects may have set up your database to make this distinction. Wikipedia and dictionary.com both have definitions of the word “customer” that include the word “purchase”. This suggests that prior to being a customer you are only a potential customer. But someone who is a customer, is a potential customer, or is just a visitor (not a customer at all) is always a user because he/she uses sites, apps, stores, cars, pencils, and things in general. I think the problem here is twofold: 1) “user” doesn’t just apply to HCI, and 2) the broader definition of “customer” is a marketing term - to marketing people everyone is a customer as long as you set foot in the store (or set fingers on the site). But the real point here is not just the definition, but *whether the entire public knows the definition*. Some business people shoot themselves in the foot by being clever and smugly forcing definitions on the public.

Good UX does not require our users or customers to look up the definitions of words.

## Statistics

Some of the discussions I’ve taken part in about statistics suggest that we don’t put too much faith in them, with which I heartily agree. Sometimes, because people put a false trust in them, statistics tend to take over and bulldoze better thinking. We don’t need to just know the numbers, we need to know what the numbers really mean, if they were achieved correctly, and what if anything to do about them.

For example, it was stated around 2010 that users only stay on the home/landing page 15 seconds. Why do they only stay 15 seconds? Is this for all types of sites and pages? Is this for first time and return users? I think we need to do more thoughtful interpretation (as well as thoughtful process or product change) before following some general statistics and building assumptions from them. That said, they may be a good guide, especially when compared against opinions that may carry too much bias consciously or unconsciously, through individual experience or assumption.

If you think about it, the way statistics are gathered can be biased from the start (just like testing can be). Consider gathering opinions on two or three images that represent some concept. First, there are probably many images that can make that representation. Second, what components are in (or out of) the images shown? Does one have a pretty, slim 20-year-old girl in it while the others don’t, and are you gathering from an all-male or mostly male group? Isn’t it obvious which will win (depending on what similar tricks

are being played in the other options)? That’s actually a form of non-monetary bribery, especially when the slant is obvious. If you don’t believe this can happen you would be wrong, as I’ve worked in companies that do it.

## Minimum Viable Product (MVP)

Recently the term “Minimum Viable Product” has been thrown around as widely accepted advice for startups. Is it just another fly-by-night trend? Like everything else, it depends on how individual companies define the term. “Quick to market” (and fix/finish it later) has been around a lot longer, and is essentially the same thing. Obviously the key to defining an MVP is in wisely determining the initial features of your product and whether all of them are really necessary to build something close to a viable product without taking too much time. Some companies I’ve worked for opt to wait a bit longer in the hope of surpassing the competition with a more complete and finished (i.e., less-buggy) product - requiring competitive research and some street smarts. Of course this is always a gamble, but maybe the bet can be hedged. As I mentioned elsewhere in this book, it often only takes a little more thought and effort to create a better product and reach (with ease of use) more of your market.

## Code Is Subservient to UX Design

What is to be built comes first, how you build it comes next, and, seemingly, incorporating ease of use comes later. However, there may be some adjustment back and forth. While I agree that the how can influence the what, the what should happen at inception and be altered if necessary by the how later.

It was said in one online discussion: “...code is trivial and above all, subservient to design”. I would agree that code, just the control of the data and markup of the pages, is trivial - but how the code is designed is not. The “subservient” part of the statement is quite true, and sadly, after decades, still only a minority really understands this fact. I believe this realization alone, if taken to heart by all agency and IT management, would greatly increase the quality of all software-based products. The fact that UX is hidden in (and in some ways controlled by) marketing, product management, and even (gasp!) engineering shows that a paradigm needs to shift. An illustration of this is that if we design for two boards to be attached to each other, we shouldn’t worry about what the hammer can do until we’ve made the nail, screw, or glue decision.

## A Link Is a Promise

The user assumes 1) that links will work and 2) that they will take them somewhere that corresponds to the link text. They are disappointed when these simple things don’t happen. We are not here to disappoint. We (UX and the company) are here to take care of our users. Although this isn’t something that fits only into the UX realm, I (as the UX designer) would feel responsible to some degree if a link fails. This also behooves us to check the links to a page or page part for every change made on the page, and possibly a timely check of links, especially external ones. Everything created requires maintenance.

## Doing It Because You Can

Just because the latest cool graphics tidbit allows for detailed super realistic 3D for example, doesn't mean that it should be automatically applied to everything. This happens in companies, with the thought that if it's cool and the latest it will help visits, retention, or sales. I would guess no company is thinking "just how realistic should we get for this product?" or "how long will this be a trend until we have to spend money changing it?" Nor is anyone assuming there is more than one way to depict, say, a game character. As one person said in a discussion, the realistic Mario can easily look creepy (one example did because of the shading – the artist's pride doesn't reflect good usability or even a well-accepted aesthetic). If you task 10 different designers to build a realistic character you'll get 10 different looks, giving the viewer probably 10 different feelings, and not by intention. Thought must go into using the latest thing before just applying it. As I mentioned elsewhere, if you're okay with changing the UI every two months then do whatever you want with trends – remain aware that such changes mean taking money and resources away from some other pursuit.

## What Does/Should the Business (Product Management) Do Concerning UX?

PM (product management) should not be designing the how of the UX. The extent of their UX involvement is to make sure the UX supports the business requirements and that's all. Sometimes of course, there may be issues that cross disciplines and that is where adjustment may be needed. But product managers (and other managers, directors, and Quality Assurance) should let the UX people do their job - they're the ones who know and who are responsible. I haven't been in many companies that practice this, but in the few that do, it sure works well.

If the PMs are doing their jobs right, and all management understands the definitions of department/discipline responsibility areas, UX should almost never need to persuade anyone of anything. If you think they should, then developers should persuade PM about code, QA should persuade PM about their testing processes, and PM should persuade other departments about their decisions too. Why is only UX under a companywide magnifier, and especially by those who don't know it?

Collaboration is necessary and good when done well, but as mentioned elsewhere in this book it does not mean compromise. There should be limited compromising with UX design, just as UX should not expect PM to compromise their business goals. If you allow non-UX-savvy people (regardless of rank) to think they have a say in the UX, you're opening the door forever to contamination, not collaboration. I should clarify I'm talking about the taxonomies and mechanics/grammar of the UI, not business decisions like content and content prioritization.

Much of the problem that companies suffer with concerning UX is simply the misunderstanding of what it is and what it isn't. It is clearly not just another cog in the production machine. The company higher-ups need to determine and support their ideas of who was hired to do what and who has what responsibility and authority. UX should not product manage and PM should not design UX. But both can help each other through discussion. PM should make decisions to support the business requirements and UX to support the customers. If that distinction and the department definitions/responsibilities are understood, there should be very few clashes, if any.

Of course this seems idealistic, and it is to a point, although not impossible to implement. The problem lies with people too intent on obeying the letter of the law to please their managers, not wanting to explain things, not wanting to “give in”, not thinking about how to resolve an issue to everyone’s acceptance, and not caring to take up time in a meeting “getting into it.” It may also be viewed as requiring compromise (again, going beyond collaboration), which may be required to some degree, and each side doesn’t want to grant it. Extended collaboration in this concern can be a real headache, but it is worthwhile in two ways: 1) the particular issue is more acceptable than the quick fix which accommodates only one side and leaves everyone unhappy, and 2) it promotes such collaboration in the future, meaning people will go into such discussions with a more open mind.

## Best Practices

Terms should convey what they mean without just social acceptance of a definition that is unclear or missing pieces for the sake of being short and digestible (it is curious that we shorten labels for easy reading but depend on contextual recall for definition). Once we get around the over-concision of this term, we can get a clearer view of what it means.

There are actually two different definitions for best practices.

1. Real best practices. These are the ideas that have stood the test of time, like understandable error messages with a way to fix the problem, avoiding distracting screen elements, avoiding typographical errors, excessive or inaccurate search results, dead links, and so on. In other words, best always.
2. Timely best practices (a better label is “the most popular current practices”). In other words, best for now. They are not determined by time or a panel of experts and they usually only last a short time. This type carries weight by who made them, not by true value or anything beyond popular acceptance. Current trends only become best practices with the test of time (and that means months not weeks).

Best practices, by firm word definition, would be practices that cannot be challenged and will last long. What we call best practices seems to change as often as every month, and seems to be in constant flux, with no group authorizing their definition. Mere social acceptance, popularity through first impression or visual coolness, maybe a tie to some name, and pride of knowing a term promote its existence – all reasons with no real substance.

True best practices are those that are supported by the UX experts based on research, experience, high usability, and prolonged existence via true acceptance in the user realm.

## CHAPTER 7



# Planning

Prior to, or perhaps the first part of, performing UX architecture and design is a planning step. Who is the customer? What can or should we provide (based on both user needs and speed of release)? What would we like this to do? What does the customer expect? Are the answers to these questions in compliance with our mission/charter?

## General Product Planning

This topic was touched on briefly in the discussions, in kind of a side door way. Many research, design, and production problems have their roots back in the planning stage. This often can't be helped, but such issues should be recorded and reviewed in initial planning sessions for the next product.

There are levels of planning, or plans for each level if you will. As a suggestion from experience, there should be an overall plan which would involve the reason for the product, who the audience is (and all its parts), the taxonomy of products if ecommerce, etc. The next level would involve the site mechanics, such as the navigations and flows, and rules for such. The next level would go into page design, and so forth into finer granularity.

Whatever the approach within the levels, it's important that companies spend time with these plans, and hopefully have some people with experience not only in planning but in execution in the planning structure. Where I've seen planning fail is where individuals try to influence thinking in favor of their department or group, often unknowingly, through some fear or other. Such activity is hard to avoid because it is often hard to recognize – each individual knows only his or her own area of expertise so there is no check beyond logic and simplicity. This causes another level of error because since everyone can discuss the simple stuff the group can relax in a comfortable discussion. Good decisions may be made from this, but the topic will overshadow other things that need discussion that have lesser discussion comfort. Anyway, the psychology of group dynamics requires that someone (or everyone) keep a vigil in the meetings and coffee talks of letting lesser topics snowball over more important ones.

## Responsive Design

Responsive design means that your site will adapt to the screen size and characteristics of the device used. You needn't build two separate sites, for example, to allow access from both a laptop/desktop machine and a smartphone. This means a single source for design as well as content - no duplication. The basic way this is accomplished is with specific mechanics (the @media rule/query) in your styles, and providing properties for your page components according to the screen sizes you wish to accommodate.

There are other methods, such as allowing your site to recognize the device that has opened it and writing custom code for each device. Content can still be single sourced (through a CMS - content management system), and careful creation and maintenance of styles would provide the design. This way may be used for special concerns, such as a different or limited app for smartphones and the main site being the superset.

CAUTION! Of course creating different code, even with shared styles and data, and even if maintenance is given high priority, can be a disaster if built too differently. Apps are always considered to be subsets of their corresponding sites, and as such may lose some features; but, *the code for all devices should be created against the same UI and the same flows.*

There has been much discussion on design for various screen (device) sizes. I'll use the words "large" for desktop, laptop, and notebook computers, "medium" for tablets (which could include small notebooks and large phones), and "small" for mobile devices, as a very general distinction for the purposes of this section.

Actually, there are so many things to consider in screen design per size that they would fill a book (and probably already have). But generally there are enough differences -- not only in screen size/interaction but in feature priority, ease-of-use, manual manipulation, experience differences (sitting, walking, running to a train, etc.) -- enough that the general recommendation is when designing a product to be used on all devices to start from scratch (after determining goals and tasks) with all devices in mind. Arguably the best approach is to start with the small screen first, after features are determined, assuming the small screen devices will have a subset of all features. In fact, I agree that user research, specifically for the phone experience, is necessary - the bad assumption would be that the list of all goals, tasks, functions, features, and so on, is the same list for all devices, depending of course on the product size. Generally speaking (and I mean *generally*), design of nav and flows for the small screen will probably be very adaptable if not usable as is for the larger screens.

The design of the app or mobile web site has at least slightly different requirements, even to accomplish the same main goals, and even above and beyond device size and manipulation. In fact, some features can be hidden in mobile where it's not necessary to do so (and shouldn't be done) in the large screen. More to the point, some features may not even need to exist, or at least not in their full detail, on a phone, due to low or nonexistent usage need in that device experience. There is also the geo-location ability in smartphones (especially for emergency and ecommerce), connection speed and other differences, which may alter feature priorities.

Further, it's not just interaction, but mindset and expectation. The mobile experience is different in enough ways that I would recommend thinking about design for it (at inception) as separate from design for other devices, even to feature and goal level, again depending on device properties that may affect usage differently from those of another device.

Here is an example chart showing a simple way to think about product design for devices. List the features and consider which devices need them, in part or in full.

Feature	Mobile	Tablet	PC
<feature A>	IN	IN	IN
<feature B>	OUT	IN	IN
<feature C>	LIMITED	LIMITED	IN
etc.			

Some rules to consider when designing for multiple devices:

- The product should have the same look and feel between devices, for those who use multiple devices with your site
- Navigations should be similar in form and content
- Pages should look and act similarly, except where device differences are understood by the entire public (although you may want to consider newbies)
- Understand manipulation differences – mouse versus touchpad versus thumb/finger, also finger use on tablets versus finger use on mobile
- Small screen usage has increased over the past years and it would be a good idea to think about accommodating that device above the others, of course based on your product type and target market

## Design for the Highest Percentage of Market

I have often worked in companies whose UX departments (or business leaders) felt that the 80/20 rule is ok – that is, to design for 80% of their users and hang the rest out to dry. This is a decision based on a ghost – the seemingly but seldom real high cost of time and effort to design for the max. I have always attempted to design for all or most, by first thinking about what the real cost might be (to go beyond the 80), including that design thought, in time and effort (and resource). A few times I was able to convince the leadership that with just a little more effort we could build to suit over 90%. But this doesn't happen without taking the first step of giving a few minutes more thought in design. We are too quick to judge based on nothing more than possible extra effort and possible schedule extension, rather than the *real* issues.

## CHAPTER 8



# Design Concerns

“Design” is one of the most dangerous and misunderstood words. Design can mean anything from natural order to planning your escape; from architecture to art; from creating a bridge to destroying one. You can design software, design a room, design someone’s career, have designs on a person. Unfortunately, this word has come to mean a variety of things in business – artistic work (such as graphic design), engineering work (such as building or software design), and experience design, among others. We need to be careful of the hidden tags on this word that we may or may not see in the context of the discussion.

Prior to a discussion on design tools (mental not manual) or design implementation, some things we may take for granted should be considered and understood. Some of these are basically definitions and awareness; some are older ideas that still apply; some are second nature, and some really need to be thought about more often and more deeply to create better products.

## Clever vs. Usable

Many people, including some UXers, consider “clever” to be “good”. While cleverness is often impressive, the impression most often doesn’t last. Clever is usually *not* good for this reason and others. The point of UX is not to make something flashy and eye catching, but to make something easy to use first. Can these ideas coexist? Can a site or app have cleverness added to ease of use without frustrating that ease? Sure, but 1) first make it easy to use, and then 2) make it visually appealing without cost to the usability. This is a difficult task and not one quick to construct.

The real question here is how much reflection does something “cool” require? Any at all means a failed UI but let’s examine this anyway. The answer generally depends on each instance. A good example is the hamburger icon in mobile apps (the three horizontal lines at the top). It started as three lines in an obvious button, which provided no clue to where it would go but at least looked like an affordance (something tappable/clickable). Then designers took the button background away. Then they got visually clever with it and made it look more like just a design component, assuming everyone would know what it is (since, so they think, everyone with a smartphone uses apps that require it, and there will be no new smartphone users ever again, and if there are, they can learn from



their smartphone-carrying friends). The anomaly here is simple: UXers with experience and software engineers automatically know what to do in such a UI, since they use such apps not only every day but often *within* every day, unlike many of their users.



At one company the project group I was in had a weekly design meeting, in which we were to show good or bad design ideas we found on the web. One day someone showed a site for a wiki which used very compelling animation when loading a page. It was smooth and beautiful, and quite clever and creative. However, it took a few seconds to load a page, and the links settled, only after chaotic movement all over the page, with no obvious visual structure to them (and actually there was none), causing the user to scan the entire page for the desired topic. If one was found that wasn't quite right, the user couldn't look nearby for a similar title because they were all scattered. The beautiful animation would generally be appreciated about twice. After the honeymoon, frustration sets in not only because the animation has gotten old, but because it is now realized how awfully long it takes to load the page. And the visual joy is destroyed by the lack of any structure. Worse, this was for a wiki site! Such a site is about finding information. And even worse, and I find this quite scary, the younger people in my UX group actually considered it a good UX design. It was only a good design in the visual sense; it had no concern for the users' needs or wants, just glamour and pleasing the business (payers/client) – nothing that lasted past a second glance and became tedious after that. And worse to an even deeper level: there was no way for the user to defeat the animation. We have to think of the future, not just the first two times we use the page.

Here are some ideas to guide the thinking in this area. It seems that, since cool has its place, and can actually help the competitive edge *when usable*, it is our job to find/promote the usable cool, and fix or avoid the unusable cool. Here are some suggestions for doing this:

- Educate both the designers *and the deciders*
- Offer similar but better options
- Refer to Don Norman's (and others') examples
- Train designers in usability (or start a standard practice of automatically checking designs with UX)
- Embed a seasoned UXA with authority into the process

There is no inherent correlation between cool/trendy and usable. A site can be cool and not usable, probably resulting in low revenue and low returns (and high head-scratching). It can be usable but not cool, which, depending on content, could also have less visits and returns. I remember an e-commerce site that was both low in usability and ugly, but high-revenue; this was mostly because of the content – price, quality, and availability. After all, a site is not high revenue, a company is, mostly if not only through what they sell. Yes, the *aim* should be for all components to satisfy both the users and the

business. Typically the aesthetics are the first front (and a quick one), usability the second and ongoing through the experience, and content usually realized third (and afterward first if high in satisfaction).

A good example of coolness over good sense was exemplified some time ago with carousels – a 3D rotary movement that expanded a few items “in front” and reduced the others which were situated in an ellipse going behind those in main view, at an angle up and away from the viewer, like a ring laying on a desk. The user swiped or clicked to move the entire chain and bring others into focus. These were all the rage – now they have fallen out of favor. Was it ok to use them for the time they were in vogue? If yes, then exactly when and why did they fall out? Maybe some analysis of this question can help us avoid making similar mistakes in the future. My (educated) guess is, given a standard look of such a device, that although they offered a view of the full range of items, their movement may have been annoying and they only showed about 3-5 items at a time without user intervention. Rental movies for example shown this way did not offer the quick visual scan that a grid of say 12 items per screen view did. So in my estimation the “coolness” of the carousel rotary view did not outweigh the convenience of the less cool grid view which offered more items at a glance but wasn’t as clever and eye catching. Is this really surprising? And did the carousel increase sales? Maybe for a short time – but the cost of redesign after the fact probably offset those sales quite a bit. This was a great example of eye candy overcoming good sense. However, I would say that if a company wants to constantly update their site(s) with the latest trends, fine, as long as they are comfortable with and aware of the cost.

Customers can adapt to UX changes over time, but my observation here is about the industry being too attuned to current trends, seemingly good interface ideas, and loss of face by superiors when confronted with advanced thinking by the UX team and the need to have UX practices explained to them. These things make the UXers interpreters of customer communication and in general the only company department required to explain its actions in detail. The other areas also force us to provide statistics, the only justification the business is comfortable with (nevermind that we are experienced in user psychology and might know even better than some little bit of research data). In any case, hold fast to your convictions about good UX practice regardless of trends, marketing likes/dislikes, and copying bad competitor ideas. You may be outvoted, but at least you’ll be exonerated. There are many beneficial paradigm shifts happening constantly in UX, but they don’t happen often and fast enough.

## Trends

Trends are the latest cool UI components that become popular because they seem really cool, most often regardless of usability.

Be careful - trends are tricky. When using them, consider first how usable they are, how they affect other page components (disruptive aesthetically and to the layout), how they affect flow, and especially how long you think they’ll be in vogue. If you change your product UI every few weeks, then no problem, because that causes your users to be often surprised in any case.

It's good to know the current trends in UI design, not so they can be automatically included, but so they can help your first impression and hopefully up return count. But they should be considered for inclusion only after a good examination of their practicality and projected life. If you spend time and money in redesigning and redeploying your product on a monthly basis, you can probably use most any latest whistle out there. If not, it is a good practice to decide if using the trend is worthwhile, based on your product revision cycle time, design/development costs, and, as mentioned, it's practicality. It is often better to think about a design and create your own solution, assuming it hasn't already been done (although you might have a tweak not thought of) and circumstances permit.

Some trends may be ok, and some may metamorphose into devices/affordances in their own right. It's not always easy to tell, and even the experts can be wrong. You just have to ask yourself if it's worth the chance.

Many people believe that trends should be copied regardless of characteristics. This gives them a certain false power. We all seem to forget that they nearly always go away, which leaves a mark on our users' minds that we are fickle and just followers; and the last holdouts of the trend suffer an embarrassing closeout. Granted, trends may initially attract more customers, but over time this just makes the business look petty to some degree, even to those in younger demographics.

More specifically, UI trends should be analyzed carefully before inclusion, *especially if they require extensive work to undo*.

Good trends would be those that employ good usability, aren't just clever and will probably last. The best of course would be those that are both cool and highly usable. Everyone up the decision chain in a company needs to understand this (coolness usually takes precedence for those not truly UX-savvy – and unfortunately for some who are). I strongly recommend to all that those helping to make such decisions that they trust their UX people, as they will feel the pain of UI trend failure.

## First Impressions

In web site design, and in application design, users' first impressions are extremely important. They set a general feeling for the site/app that is difficult to change. Since most users' first interaction with your product involves the home/landing page, it is especially important to present a clean, attractive appearance, and a clear display of content elements and places to go.

For most people, the first second or two of viewing a home page sets certain expectations of the site and also sets the viewer's mood and feel for the company. This is only improved on if the site is easy to use and has quality content. How important this is could be argued, but it is important that expectations be met and moods are set positively (even changes in subconscious decisions are sometimes difficult for the ego). There is another issue here, however: the user generally is entering a site or front page with a task in mind, in other words, with expectations, so this tiny amount of time can carry a large amount of import.

In most cases and for most sites/apps, basic expectations on entering a page for the first time, especially the home or entry page, are the following (these are expected, not necessarily all the best ideas):

- a level of professionalism (in look and content)
- standard nav options (sites) and placement
- appropriate content
- logo in upper left
- easy to digest and meaningful
- ads are not obtrusive (off to the side; not animated)
- pleasing aesthetic
- no surprises
- no obtrusive subwindows
- no lengthy introductory animation without a way to stop it and never show it again

Note that some of these show simple consideration for the user. If any of these are missing or defeated, the user's expectations have possibly already suffered and a small negative mental mark has been made on the overall experience (and possibly on the company). Sometimes other things can overcome that mark, such as ease of finding content on a subsequent page, but this should never be relied on to fix the real problem. The most lasting impressions are the first ones, so it's well worth spending time getting the entrypoint right.

This applies also to the first screenload – what the user sees before scrolling. Newspapers call this “above the fold”, and for many years have realized the importance of what viewers see before even buying their paper off a stand. The headline is here, and usually a picture, which are the first eye contact the viewer can have with the paper. This screenload translates in web roughly to the first 600 pixels in height on a PC screen, less on the smaller devices.

## SEO and UX

SEO (Search Engine Optimization) acts to get users to some page on your site. UX (and Information Architecture or IA) act to optimize the user experience (which helps in retention and return) and goes to work when the user gets to the site. Once the user is there, SEO for your site is over.

SEO is basically a method for optimizing the ranking of a site/app in search engine results lists. Certain implanted components in the markup code and content accommodate search engine probes which determine priority in a search results list (the full mechanics of SEO is beyond the scope of this book). The mentioned components include META tag content and elements of the visible text content, so there is a science to it. Worse, what the search engines are looking for or analyzing changes – what could be good one day may be detrimental the next. Architects, designers, and copywriters need to keep a vigil on SEO trends.

Some SEO elements affect the user interface in various ways. For example, a balance of the usage of certain words affects the SEO positively, but these words are used in content text, so too many or too few appearances of a word can affect the reading of it negatively. Beyond this, if you want to use certain words many times it can have a bad affect on the SEO. It is a delicate dance.

SEO can affect the IA and vice versa. One example of this concern is building multiple “sister” sites which are for the same base company but belong to various subsidiaries, all of which sell the same products, more or less. If it is desired for all the subsidiary websites to appear high in search lists, it may be required to build different navigation styles to avoid a redundancy the search engines may spot and reject.

The bottom line is: no matter how well a web site ranks in the search engines, if it is not usable, no matter of SEO will save it.

Be aware that SEO engines are in constant flux - mostly to guard against those designers that discover a loophole and take advantage of it.

Designing for the combination of SEO, IA, and UI together is a topic all its own and well beyond the scope of this book. For more help search the web for SEO sites and books.

## Information Architecture

There are two types of IA – back end and front end. Back end IA has to do with database design and data component field definition, that is, the data at the most fundamental levels. Front end IA is about the structure of the UI to display that data (content). The Information Architecture Institute defines (front end) IA as “the organization and labeling of website content to support usability and findability.” The findability mentioned here is about finding information within a site, not about finding a site from a general search. I like another source’s (Human Factors International’s) thought on the definition: organizing the content and characterizing task flows/relationships within the organized content.

## Customers vs. Business

One of the most difficult obstructions to good usability is business desires. Businesses with something to sell want to “guide” the user into buying as much as they can convince them to buy. This is effectively being a “pushy” salesman. This pushiness manifests itself in many ways. In ecommerce two ways are called cross-sell and upsell. There is nothing wrong with such concepts in themselves, but like anything else they must be balanced against user wants, needs and expectations, and not hinder the experience by being overdone. But how can you tell when this is overdone? User testing will tell, but we want to have a guideline to avoid overdoing in the first place. A good point to remember here is that when users are inundated with ads they tend to ignore them, so such a technique at some point becomes self-defeating. Also this doesn’t just apply to your site, but all the others that have contributed to such general user desensitization.

In addition to this, there is the fact that the business is often quite ignorant of usability principles (or at least how and when they are applied) and worse, depends heavily on what “seems” right. It is often up to the UX people to educate them, every time it comes up, on why UX is suggesting something to be done or avoided. It might be a good idea to have product managers, copywriters, and other business factions either be instructed in UX and/or made willing to listen to and comply with UX. I have found it

is somewhat easier for business people to explain their reasoning to UX in each instance rather than have UX explain why that instance has good or bad usability. Business reasons usually involve the mechanics of complying with orders from higher up and product use history, which are generally simpler to see than the UX psychology involved in some circumstances.

In ecommerce, we need to realize that the entire phenomenon of any purchase system is about the customer, and not really about the business; or a better way to say it might be it is about the customer and therefore the business must think about what the customer wants. That said, business requirements can be applied, but should in most cases be kept unapparent.

Any requested customer information should be optional, never mandatory, and any cross-sell/upsell by the business should not interfere, especially visually, with the flow of the purchase. This is not rocket surgery. The business isn't doing the customer any favors - they can buy their items elsewhere. And any in-your-face (that is, in the way visually or flow-wise) marketing should have happened before the purchase decision, not afterward. I have often stopped checkout at the point where they want my email because I know what that means and worse because they destroyed the fun of the purchase. If I (customer) shop at a brick-and-mortar store with cash, nobody knows any info about me beyond what I look like, so there is *little question in my mind* whether I'm going to be harassed with marketing questions (unless of course I would want to). I don't care to be asked about my zip code either. Such questions are not something for me, but for the business, and I (again, typical customer) can't see how that will reduce prices now or even in the future. But notice I emphasized "question in my mind." That is the trigger point, not the info itself. The smoothness of the flow and therefore the happiness of the shopping memory requires that no questions AT ALL arise in the process.

Of course, getting this through to the business people is not an easy task. Generally, without some time-consuming statistic gathering to produce numbers, they don't care to discuss it (often the bane of good UX).

## Edge Cases

I have too often heard "well that's just an edge case" which means to the listeners (regardless of what the speaker meant) that those are throw-away. In other words, ignore that group of users that such cases might affect. How about instead just think a little further (often 5 minutes helps) to accommodate such cases without great effect to the rest? Why not strive to cover all cases (meaning all customers) if it only takes a little time? In other words, if accommodating the few means confusing the most, make sure there is no way to provide a good experience across the board before throwing any segment away. I recommend after identifying edge cases to address them and determine effort, time, and affect. Consideration of amount of users to please is not arbitrary - they should all be considered. If you want your product to be the best, you really can't afford to ignore any users unless the business deems it absolutely necessary and understands the consequences.

Consider the 80-20 rule and negative bias (plus keen competition). If you reach 80% of your possible community today, and later make changes, you can't assume you still have 80% - you may have picked more up this iteration or lost a few. The problem here is the few that are lost, due to our human negative bias, will remember the reasons and

most likely never return (speaking only about the UI here). This means that next time the 80% is really, say, 79%, even considering the few that were gained; the extended math is not encouraging. To clarify, those that were pleased with iteration 2 *may* remember that positive thought and return (and will not necessarily tell anyone about it); those that were not pleased will more likely remember that negative thought and never return (and may tell others).

So the real point is we need to be very aware of the trade-off between the real amount of saved time ignoring the 20 and the time and other costs into the future of addressing the 20. The probable reason, besides false forecasts, is that we can't see the numbers for both ways here to make a better educated guess as to which way to go. My advice is to start an attempt at reaching the 20 no matter what, at least until the cost of that effort becomes realistically more substantial.

## OTB (Not Off-Track But Still a Big Gamble)

OTB: “Outside-the-box” thinking. The “box” is generally understood as the usual, accepted methods, processes and cognitive efforts. As with anything else, lack of an explicit definition of what's in the box causes some ambiguity. Worse, at some point an item of OTB thinking has become well-known or well-used and therefore jumps into the box itself, possibly without everyone knowing.

Generally when something is considered outside the box it carries some element of risk (just like innovation), because one reason it is outside in the first place is because it was not seen wrapped in a snuggly blanket of popular acceptance. The downside of OTB ideas is that as soon as they are revealed they are judged by some ITB safety/feasibility/cost rules that determine immediately whether they should be pursued. It is better to let them ripen a bit before swatting at them, as quick rejection may be incorrect, and as other ideas may emerge from them.

## Policing

Some interface ideas “police” the users, a concept which, like most others, can be good or bad for usability depending on how they are handled. The usual example is performing checks on too many decisions in a flow, and asking the user questions about those checks, to the point of limiting the experience so much that the user gets slowed down, confused and/or annoyed. Often those checks are necessary; when they aren't, the flows should be reviewed. Ask yourself which checks are really necessary and expected, especially those that have to do with your site content (e.g., product pushing in ecommerce) and other periphery, being more protective of those that exist for a good user flow first and promoting the business afterward.

## Context

Probably the single most important idea concerning UX design and architecture (and information architecture as well) is context. This is important from multiple standpoints.

We understand what’s being said or displayed by cues in the environment or area we’re in at the time. For example, there is no label on a room light switch, but we know what it does. The fact that it turns a light on and off is understood through learned experience, but the (more initial) fact that it has something to do with the room is via context. You can see the individual, unconnected switch in a display in a home store but context (the fact that it isn’t wired to anything) tells you immediately that it doesn’t affect any circuit. UPS is understood to mean United Parcel Service in general, but if discussing it concerning computers we know it to mean uninterrupted power supply (battery backup).

Unfortunately we take this to extremes at times and expect others to derive meaning from context without realizing drawing on context isn’t necessary or the contextual clues aren’t strong enough. In the above example, if a new UPS unit was being shipped to the company via UPS, some minor confusion could occur. This is a very simple example, but you can see how it might get worse. In UI design, lack of attention to how buttons or menu picks are grouped can get very confusing – an anomaly of the designer knowing too much and not being the user (and far from it). The best UX people are those that keep themselves naïve to the content (not an easy thing to do); but the design can also be tested by users unfamiliar with the visual to determine if the grouping makes sense. Going deeper, groupings themselves can be interpreted different ways based on, you guessed it, context.

So the suggestion here is twofold. First, be aware of context in the design. It’s easy to get sidetracked into flows and/or personal contextual interpretations. The designers must separate themselves from their own contextual realm and consider the contexts that the user may be experiencing in a page or flow – not an easy thing to do, but the attempt must be made. Second, this is a major area of consideration in testing – before dismissing any strange comments from users consider what context the user may be experiencing.

## The (Quick) Science of Options

### Options in General

Considering options in general, and in the simplest form, when you provide or recognize two options there are really four and all must be considered, although the two that exist by consequence don’t always apply. These are the two taken as both and neither:

1. A
2. B
3. Both A and B
4. Neither A nor B

When you provide two wireframe options, you actually provide three: the third is the combination of elements of the other two. This offers a great advantage in wireframing for clients because most people like to feel they are helping to build the product, and choosing from the “pot” of components provided promotes this feeling. Of course, UX must keep vigil on the combination in case it incurs usability issues.

A fourth option may become apparent, being neither of the two but something new. This could provide a fifth, which would be a combination of the parts of the original two and parts of the new idea.



Taking this further, when offering two wireframe sets and allowing for new requirements to be discussed as well, eight options occur for consideration:

1. A
2. B
3. Neither
4. Both A and B
5. A combination of parts of A and B
6. A plus new requirements
7. B plus new requirements
8. A combination of parts of A and B plus new requirements

I added this here only to show that 1) there are many options possibilities and none should be automatically dismissed, and 2) offering the client(s) some feeling of helping to design the product is not only good psychology but may actually unearth requirements they didn't think of originally.

To summarize, in any set of options, the true number of combinations is either  $n!$ ,  $n!+1$ , or  $n!+2$ , the 1 or 2 being all or none. The all or none options (either or both of them) may not apply in every case.

## Client Options

It is usually beneficial to provide wireframe options to clients. But how many? Generally, you need to cover the written and/or agreed-upon requirements, but often there are the unwritten ones which are usually things forgotten or trendy items, in either case not foreseen until something visual appears.

Concerning client wireframe options: if you provide two complete wireframe sets, say one that is usability-focused and one that is requirements-focused, and they are not the same, you can display them along with a usability gauge, such as a number from 1 to 10. Most clients in my experience (especially engineering types) love numbers as they can relate to them intrinsically and can make a judgment with more faith. The problem here is you must be ethical – you'll be biased to your design, but the numbers must be honest values based on usability issues you can point to (someone may ask).

## Average Person

An interesting human anomaly is the fact that we as a race have become overly dependent on statistics (what are the odds of that?), which have their own problems. But the resultant problem that we tend to ignore is that statistical results cater to the “average” person – a person that *does not exist*. Simply explained, in a group of three people that have test scores of 50, 52, and 60, the average score is 54, which no one had. And yet we will grade people's knowledge (skill, memory, whatever) accordingly. So if someone gets a score of 55, unless the scoring system is modified adding this new data, the person will

be considered to be above average, when 55 is more of a true average than 54. Two points here: 1) I could just as easily have said the grades were 53, 54, and 55, in which case there would have been a person that achieved exactly the average, but there would be only one, and 2) although the above example doesn't happen in every situation, *we tend to think this way*. We are so number oriented (whether engineer or artist) that the fact that a job applicant holds a 3.2 grade point average is generally more important to many bosses than what that applicant has done in his decades-long career.

The greatest part of this problem is in the definition of the word “average.” In our empirical experience, it can mean common, of the largest group segment, without extremes, usual, a “c student” or a grade of c on a project, the top area of a bell curve, the most seen, and so on. It seems to rarely mean its exact definition, inviting misunderstanding and possible other problems.

In UX, we tend to think, maybe subconsciously, of most of our users as average, which is most likely not true. “Well, the average user will...” is often heard. But what we're really saying is “we think most users are naïve enough to...” or “we think most users are smart enough to...” We then go on to test the screen or flow in question hoping in the back of our minds that the small user group we use will validate our assumption. Or we may feel that there is an average person within each persona group, which again is not true, but is usually good enough for pre-release testing.

In any case, my point here is to not let ourselves be influenced by convenience, and realize that the “average person” is a simple tool we use to view the user community (or target market) in a way that allows us to both gear our testing and design and to discuss such matters without long explanatory labels. In other words, don't make it more than it is.

## Delighting Users

My first reaction to this label was and still is “let's get UX done right first, THEN we can worry about delighting them.” Such a term assumes we all know all about providing usability, the first step (well, maybe we know, but is it getting applied?). We actually have some distance to go before considering delight, such as exposing dark patterns for what they are, determining (believe it or not this still happens) what functions to provide based on the user tasks, determining trend value, and generally understanding which things to value and which to avoid.

But, in any case, delight must have some value, even if subjective, comparative, and transitive. Delighting youngsters, for example, is very different from delighting shoppers. Younger engineer types may be delighted in discovery and the challenge of circumventing poor UI issues. Experienced users and those in a hurry are delighted in just having things work, work easily and work fast. So, as with other ideas and tools we can use in our designs, we need to get the definition (or definitions) right first.

There are different types of delight, partly to the point of what is mentioned above. The most basic and probably the most assumed is the feeling experienced in the first impression (arguably the most important impression). Interestingly, the first impression also has at least two parts: the immediate view to the eye, and a second later the first view to the brain (ease of the navigations and flows).

The initial visual does set the user's mental scene and overall feeling of the site and the company, but what designers often forget is that this scene will be seen again and again, unless the site/app is not one where there is concern about return or retention.

This means the designer is not free to do whatever he/she wishes. Worse, it means that the design should be pleasant, or catch the eye, without being overdone, time consuming and seen as trite the next time in. This puts a burden on the designer to make it interesting and memorable, but not visually overboard. And visually means colors, shapes, images and especially animations – everything grabbing the users’ eyes and/or costing time. Award-winning visual designs do not mean good usability; in fact, often the opposite is true (Norman 1990). So there is a difficult balance to maintain.

## Designing the Flow

### Flows in General

User flows are as important as site content. Users should not be surprised when they select something – that affordance should take them where they expected to go, based on the label (see the sections on labels). What shouldn’t be done is designing flows the way the business wants without user accommodation – that can easily frustrate users, unless they are designed in such a way that user expectations are met. This is usually not easy to do.

Number of clicks/taps to complete a goal is not nearly as important as providing the same within a flow the users understand. Of course, optional paths can make ease of flow difficult to design. That shouldn’t, however, keep a UX designer from thinking about providing simple flows and returns in a set of optional paths. It’s often worth the extra effort to try.

Optional paths often cause subordinate flows – a side track from the users’ main intent which normally leads back to the main flow in some way. If they don’t, you risk providing unnecessary confusion and possible abandonment, especially when it is simple enough to provide a clear way back to any major flow start on every page. Designs that depend too heavily on user knowledge of the UI components are usually just being too clever. Good designers allow for easy flow return for the UI-naïve without clogging the page.

Optional paths that take you away from the main flow and not back are subordinate dead-ends, which may be okay depending on the intent or nature of the flow. However in most cases the probability that a user may want to return to the main goal (top node) or the start of the subordinate task (leaf node) is high enough that letting them get lost is poor design.

### Stepwise Flows

Probably the most well-known stepped flow is that of the checkout process in an ecommerce site (known as “bottom of funnel”) – for example shipping, options, review, payment, verification of order, and receipt. In such flows, there are a number of important features:

1. how many steps there are total
2. what each step is about (complete labeling needed here – not brevity) displayed as header on step
3. visually obvious where you are in the flow at any time

4. what can and can't be edited clearly understood in each step
5. if the user can go back a step, or multiple steps, and whether lose that page's information (or that of pages in between) is retained or lost if you do
6. optional as applicable, but often helpful if accurate enough: rough idea of time it will take to complete
7. what will happen to the users' input if they close the browser (program, etc.)
8. how they can escape from the flow and/or start over, and whether anything is saved or not

## Autoflow

There is also some discussion happening about buttons versus automatic response, such as typing in a search box and seeing a list of possible matches to your text appear after every keystroke. However, automatic responses, like anything else, can be overdone easily. I'm sure you have encountered some sites that offer too much automation on keystrokes and/or cursor moves and became annoyed quickly. In the effort of designers to make the experience smoother and anticipatory the concept can backfire and make it less helpful because each automatic thing must be viewed and reflected (there's that bad word again) on, throwing stones in the path.

Also, if automation is considered because there appears to be too many buttons, this may well be a design flaw that should be addressed at the problem and not patched over with cleverness. Too many options may reveal a need to reconsider the overall design.

## Ad hoc Navigation

Using in-text links to offer another way to get to a page other than a menu or navigation list is a two-edged sword. Offering links in context is great for the user because it speeds their flow. However it also means that they may not be able to retrace their steps later, especially if the normal or global navigation selection routes them to a page that is slightly different (out of context) than the page the user remembers. Such in-page links should not be context specific but rather call the same exact page or page part that any navigation selection does.

## Flow Buttons

The labels for flow buttons or identifications for checkout flows have been done many ways. The ways that work are those where the designer/copywriter paid more attention to what the area/page meant than keeping the label short. In some cases I've seen such flows void of any step labels, keeping the users in the dark as to where they are and what steps remain. In the case of these flows, that is, process steps, both ideas should be used, and again brevity should make way to understanding in labeling.

## Overconveniencing

The business, product managers, etc., can offer too many features. Upon their first release of the iWatch, Apple offered thousands of possible customizations for the clock. That in itself is not wrong (other than the idea only a few dozen will ever be used), but 1) it could make it too complicated to use if the user desires to know what can be done, 2) it could be construed as making the device very expensive for invalid reasons, 3) it could require help files, or maybe all of these. Part of the marketing was probably that a main persona type is a geeky teen that takes pride in the discovery of some otherwise unfound feature whose discovery they can show off to their friends (a very real marketing scheme – if it’s easy to include thousands of options and “hide” most of them). But hiding options means careful selection of which should be hidden and which should not be. This could take weeks of testing to determine because of so many different user types and preferences.

If you are not concerned about such a specific persona, do you design this way with the rule that it is ok if users will figure it out eventually? That they will not be confused enough to send it back and buy the competitor’s? But of course, if there is a market for it, sense notwithstanding, someone will build it.

## Above the Fold

Some believe that concern for what appears above the fold is a myth. This is fueled by mobile device UI manipulation – it is very easy to slide your thumb or finger to scroll. However, the easiest interaction is not manual - it’s moving your eyeballs. Regardless of how easy scrolling is in mobile, you don’t scroll first - your first interaction is the visual scan of what you see where you landed. If you don’t see what you want there, then you scroll. Granted there isn’t much space to pack all the important stuff into the first mobile screenload, so scrolling is nearly always necessary - but my point is your eyes scan “above the fold” first (sometimes even if you’ve been there before), making it the most important area of the page. It is also important because for the home page it is your first visual impression of a site or app, with the first content impression (of the content on that page) a second or so later (assuming there is content in that area as well).

## Hamburger Menu Icon... Top Right or Left (Mobile)?

Here’s a discussion that has been going on for months – where do you put the menu icon (“hamburger” – 3 horizontal lines) on your mobile page?

I believe it is agreed (and somewhat obvious) that such an affordance would reside at the page top. But which side? This obviates research in side preference. Sometimes researchers get caught up in the analysis and in this case want to get findings on single hand versus dual hand use of the device, and hand dominance, assuming that it can be told from that data where to situate the button. But wouldn’t it make more sense to just ask people where they want it or feel it should be, and leave the more interesting part of how such results relate to single/dual hand and dominance later? More than that, hand

dominance won't really tell you much since it is hand *preference* that counts. Many left-handed people wear a watch on their left wrist, and vice versa. I am right handed, but when I use my phone with one hand it is usually the left (probably because when using both hands I hold it in the left and tap with my right forefinger).

A suggestion is to possibly invest in customization - default to, say, the left, and allow the user to change it (assuming the dropdown items span the width of the screen). A key to automagnify this of course is remembering the choice over number of accesses and asking if they want to set it that way permanently (assuming server or cookie settings are an option).

If you don't care to offer customization, make a decision on which side to use based on research, gut feel, and/or magic 8 ball; the more important thing is *not to change it once it's out there*. Don't confuse your users just because you can't make up your mind.

As to the icon recognition, newbies should be considered (when I first saw it I had no clue it was a dropdown). I suggest a small downward arrow on or under the bottom line to help suggest it does something, and/or designing it to appear as an affordance (like other "buttons" used), as mentioned, rather than getting too artsy with it (which I've seen happen) and making the user think. (See also the section "Clever vs. usable.")

## Simplicity

As Einstein said: "an explanation should be as simple as possible, but no simpler."

Designing for simplicity isn't simple. Simple has levels, so we have to exercise caution here. How simple can something get? In designing something to be simple to use (other factors aside), we may lose other users (customers) through boredom or a feeling of condescension or education/age level mismatch. If we design for simplicity but include questions, flows or terms that are not really easy to understand or outside our users' general knowledge realm, we can lose the more naïve users. There is always a balance. This is gauged by 1) how you perceive your potential users' knowledge/education levels within your target market, and 2) research and testing of your product prototype and/or similar past products.

Simplicity of the UX is about many things:

- Visuals - colors; order and position; grouping; busyness
- Flows - order; navigation; processes; forms
- Errors - meaning; resolution; consequences
- Content (copy/information) - grouping; completeness; little dependence on context; proper grammar; concise but complete; meaningful labels
- Overall - consistency within and without; following business purpose/charter

Also what is simple to one is not necessarily simple to all even when all involved are at similar experience levels. It is recommended to review product creations/changes with many people before release to determine if audience aim was on target.

## Proximity

Edward Tufte (among others) has discussed in his writing the idea of  $1+1=3$  in visual design, meaning two elements in close proximity cause a visible interaction. To explain further, two visual elements in close proximity, with no others in close proximity to them, and regardless of element types, cause an incidental (unless planned) visual grouping. This can also apply to the problem of two elements that should be close together but aren't. It's not enough to plan visual groupings on a page, it is necessary to see if there are any unplanned ones. Fuel for the fire to let others see your design – have them look for coincidental unintended groupings.

Visual groupings can form from shape, color, size, text, or just about any attribute of page components, as well as proximity and white space. The brain wants to draw similarities, especially in a busy page, because it wants to simplify the view.

I have two thoughts on this. An interesting anomaly here is that even alignment can cause grouping – misalignment, usually frowned upon by neatniks and those taught to keep even unrelated elements on a grid, can be used to defeat visual grouping as another or a last resort.

A second thought is that for at least a decade and a half “clean” has been a byword, and, as usual, has been used to justify all manner of page whiteness, with the culmination being pages of all white (except images) with very light gray dividing lines. They're clean all right, and often hard to visually organize. Images usually define their own areas, but text blocks can be boxed as well, using for example light gray boxes around the paragraphs. One of the most frustrating and dark things on a download page is to allow an ad with a big Download button for something other than what you came to the page for – very hard to distinguish from the text you want because it is filled with white space like your page and has no linework around it nor is it enclosed by a box. Don't frustrate your users – don't make them be on the lookout for such marketing tricks – box the ad. This often isn't caught in testing because you don't have the ads at that time – just box them all.

## Forms Design

Forms are a special part of UX because no one likes to fill them out – they're no fun. Therefore it pays to take time in their design, always considering the user at every step.

Group questions that make sense to be grouped – not sense of the data, but sense to the user. Sometimes these two coincide, but very seldom, so don't depend on it. Sometimes the questions involve data the user gets from a paper or a card etc., that they had to find and extract for example from a poorly made wallet (been there?), or a card with sensitive information that they couldn't leave out, and so on. If it is possible to anticipate this, you will be relieving some user frustration.

Some forms have questions dissimilar enough that each question could be on its own page. However if the total is low enough (like 4 or so), they may as well occupy one page together. The basis of good form design is just simple consideration for the users.

Some questions may build on others. If so, they should be in a logical order if possible.

For large forms, especially multi-part forms, take time with the design on paper or whiteboard first. And again, keep the user in mind, not the data or the analyst. Group similar questions and then group the groups. One method of designing such a form on screen is to open the first part on entry, and when that is done, open the next part and close the first. This keeps the screen less confusing. But always keep in mind that closing a section can cause the users to wonder if they can go back to it and change it. Make it clear whether they can or can't, both in the section and at the start of the form, and whether they can skip around between sections, without losing input and knowing that the code will let them know if they missed something.

If no questions can be grouped naturally to the point of a comfortable number per screen, just put a few per page (same amount on each), and definitely with a status bar somewhere for users to see where they are (also as mentioned elsewhere) and how far they have to go. They also need an escape, and be told what will happen to their previously input data if they do exit.

## Ecommerce Specific Issues

### Checkout Process

The checkout flow in ecommerce sites is fraught with issues, most of them due to the (here's that problem again) desires of the business that may cause disregard of good UX practices.

Too often the business wants to promote sales to the bitter end of the user experience – every page, even in the checkout flow, has ads for sales of similar products, other warranties, what others bought that bought the product you're buying, and so on. While its argued that the company made revenue due to some of these ads, UX can argue that for some (maybe most – this would need to be tested) users are experiencing a little less joy in their shopping experience by being tripped and hit over the head with store promotions they don't care to see. My argument has always been that once customers have decided to check out, they are in the “purchase state of mind,” meaning they are done shopping, and only want to be bothered with truly pertinent information, like a maintenance contract or needed parts (both of which should have been obvious beforehand). Isn't it more valuable to be concerned with the customer's welfare rather than trying to take advantage of them by selling them something they don't need or had no intention to shop for? Isn't it better to offer a pleasant experience than risk their abandonment or even their switch or conversion to another store? Some online retail stores need to have a check on their business decisions. And what better department to do that in than UX?

Going a bit further with this, maybe the real issue is research. Gathering data about sales made via intrusive ads must be weighed against 1) loyalty of customers based only on the lack of such ads, and 2) similar sales made by ads that are non-intrusive. Often these checks are difficult to achieve, but are absolutely necessary to give any credence to the original intrusive idea at all.

With all this said, it must be added that there are certain technicalities and/or legalities that must be observed which may interfere with good UX practice, such as discount pricing that cannot be shown until the customer is in the checkout flow (can't see the discount in the product page). It is very helpful to the user if such issues can be



circumvented somehow in the original business requirements. If that is impossible, *then UX should make every effort to explain such anomalies in notes on the page* (on hover/swipe/text link, etc.). The reason here is not just to have good UX, but to show the company's concern for their customers.

## Guest Checkout

A retail business would like to get as many members as possible (member meaning known personal info for email ads, special marketing, etc.). Customers however include a segment that don't care to be members, at least at the time, and just want to purchase something today without hassle and without relinquishing their personal information (and not visiting the brick and mortar store). The better stores realize that such people could later become members, especially if they are treated well as customers first. This could include incentives and so on, but as far as their shopping experience goes, they should be allowed to check out as a "guest" without jumping through hoops or having ads pushed in their faces.

A suggested rule to bear in mind in ecommerce design:

The idea of any purchase system is NOT about selling, ***it is about the customer.***

This statement makes it easy to see the common problem of UX versus the business.

A more general suggestion along the same line, not only for ecommerce but any type of flow process:

The smoothness of any flow requires that no non-topic intrusions arise in the process.

A keyword here is intrusions. Side bar ads, small enough and without visual distractions (animation, large blocks of color, great difference from the main area) are acceptable but of course should be pertinent to the flow.

These concepts have one important basic premise behind them: it's all about the user, and no one else. In other words the business should be educated in the thought of catering to its customers, not just focused on making money (that is, without the ability to make money through good UX). This promotes not only a positive feeling about the site/store, but also a feeling of trust between the user and the company. It seems the issue is simply that the business tends toward instant return rather than the benefits of loyalty and long-term relationships via providing a pleasant, non-frustrating and non-obtrusive experience.

## CHAPTER 9



# Design Tools, Methods

This chapter is about the mental toolkit you need to know what to do and what not to do, rather than the manual implementation of your designs. It is not meant to cover everything, but these would be most of the important things.

There are many affordance controls – most are straightforward in their design and use; some can be complex and problematical. Here are some thoughts on these.

## General Labels

An entire book could be written about the science of labeling alone, and a few of the larger chapters in it would be about labeling UI affordances. This is not surprising, because there are multiple purposes of a UI CTA label, and because they are so easy to get wrong. UI labels that act on flow can be the most dangerous to good usability and the most difficult to write; those that don't, like simple navigation selections, also need some thinking beyond the “obvious.” There have been commonly accepted ideas of how to form labels in UIs, ideas which have for the most part been wrong from the start.

Label creation has been a problem since the first computer interfaces. It has been exacerbated by the myths of precious screen real estate in the GUI world, especially early on. With the evolving expansion of the web screen this has gained some help, but now there is the mobile screen to contend with and the problem is there again (some mobile apps however tend to have only major functionality so label concision is easier to understand).

A bottom line could be: brevity without complete interpretation doesn't work.

Another bottom line (for all of UX as well) is: always consider the welfare of your users first. One way to help you remember this is the link of the logo, in standard position at the top or top left of the page, which takes you to the home page, should have a hover help (web) or some affordance that offers an explanation. Why? Two excellent reasons. It helps newbies know what's going to happen, and it promotes the idea, consciously and subconsciously, that you care about your users.

Every affordance/CTA that is unclear even minimally should have some help method immediately available to explicitly say everything that is happening and in detail (and hopefully what's not happening), even if the label itself says it all. Of course, there should be no lack of clarity in affordances in the first place.

Our inherent desire to save time via making assumptions and following popular practice is the enemy here – our egos tell us to shorten what we know will happen, usually by common practice, and we skip telling users what actually will and won't happen. This is reinforced by the thought that brevity is expected and is somehow better, and this is reinforced by the designer's laziness ("it's common practice, so if it's wrong, it's not my fault").

The best rule of thumb about affordance labels that I've used for years is quite simple. For any given affordance, write down all the things that will happen, AND the things that won't happen (this is the hard part), considering what the users may assume by the wording or the nature of the flow. That is the true affordance label, however long it may be. Then identify all the actions by simple verb-noun phrases. If at this point you end up with too many, you may want to reconsider the affordance itself, add resident text to the UI itself, or think about a breakdown of the flow(s) (you may be trying to do too many things with one button). A great example of this is the note "shipping options appear on the next screen" in certain ecommerce checkout flows.

Labels should be made consistent in word choice throughout. Users notice, though maybe not immediately, that some functions that seem clear from page to page may have different labels, and this causes reflection, which means some small amount of frustration has been encountered. If it is important to point out some difference in one spot even though the functionality is the same, by all means do so, but make sure it is important enough.

One of the most difficult choices I've encountered in UI design is labeling on controls that may be difficult to put in words but will be understood by the users eventually (the "acquired taste" idea). This proposes a double problem: not understanding the label the first few times it is seen, and not getting bored with it afterward. Generally, the boredom issue is outweighed by the possible initial misreading. The same problem is addressed and neatly resolved with certain warning dialogs that include a checkbox which allows you to decide if you ever want to see that dialog again. Unfortunately, we can't easily use the same principle with control/flow labels.

As for jargon, as in the cases of small customer groups in a domain or a small target market, users may expect to see it, and will want to especially when the alternative is too wordy. As long as they consider new users, outside users (if any), and triteness, it could and often should be used.

## Column Header Labels

Column header labels are a special case. The header label, like any other, should be clear and concise, but often the words cause the column to be wider than needed because the column data beneath it is so short. It is tempting, especially when users want to see large amounts of data on the screen at once, to abbreviate or otherwise shorten the column label and make room for other data (or avoid horizontal scrolling). There is no hard and fast rule here – all the requirements need to be considered and then a decision should be made. An option might be to abbreviate, but also use the title property to describe the column label in full. Besides the requirements mentioned, don't forget consistency. A change in one column label can affect all column headers throughout. For example, if you've abbreviated a word in one header it is probably best to abbreviate it in all of them. Also, column headers can be multi-line in most cases, which may also alleviate column width problems.

Field labels (such as “Name” on the left of or above a text input field) should generally be left justified, but if labels left of the fields appearing together vary greatly in length this could look awkward and be hard to read. One solution is to make the text smaller and use a multi-line label; another is to right justify the labels against the input boxes and move them to the right (although the boxes should line up horizontally so this may look just as bad). A better solution is to move the labels above the boxes, all left justified – this requires making sure the user can easily associate label to box. Use your own judgment here, and remember that the justification should also be consistent throughout the site/app or at least throughout the form or page in question. It may be worth the time and effort to shorten lengthy labels for this reason.

## Page Titles

Page titles are also labels that require great care in forming and using. Every page should have a title, if for no other reason, to be able to reference it or discuss it, but mainly to tell the users what page they are currently on (remember bathroom breaks).

The title should reflect where you are and/or what can be done on the page. Page titles should be concise but complete. Remember that the title is what should be used word for word in the breadcrumbs list, which is usually horizontal, so titles should be short. However, main words can be used in the breadcrumbs, as long as there is enough differentiation in the words (users will draw comparisons of similar words and may get confused). Page titles should also be prominent and located consistently.

## Button Labels

Since the beginning of GUI, people followed a general rule that labels (especially on buttons) should be kept short, probably for different reasons but especially for the sake of screen real estate, as mentioned previously. Since screens have become larger physically and allowed for tighter resolutions (at least for the web), this assumed restriction has been relaxed somewhat. However, regardless of screen size and visual challenges, the label on a button must be first clear, then concise. One example I encountered at a financial client was the use of the word “Cancel” to mean “Cancel the financial transaction” (strict context), not the generally accepted “Forget what I did here and go back” (assumed context). I suggested using “Cancel Transaction”, which in turn required all the sister buttons to use the word “Transaction” as well (like “Reject Transaction, Approve Transaction”, etc., which didn’t require the extra word by themselves). Granted this required more real estate, but that should be minor against user confusion caused otherwise. So take note: here is an example of how to label a button that has nothing to do with the function of the button, but rather the consistency of the button set and the similarity of labels on “standard” buttons.

Labels on buttons should never be limited by old beliefs, no matter how applicable they may have been in the past. A button label should always be clear. Clarity doesn’t mean how well you can write or how cleverly you can use words, but rather how clearly the user understands what the affordance will do, and what it won’t do. A rule I’ve used with great success here is to try to imagine all the things the label could mean within the task mindset, and even without, and address these in the label if at all possible. This

should be done for all labels but especially for those operations the user has any question about. An example might be “Save”. If the save action also closes the window or brings the user to a different screen, it may be better to use “Save & Close”. The list below shows the various things to be considered when labeling buttons.

As mentioned above, one of the accepted ideas of button labeling is to make the label very short. Notice that doesn’t say concise. There is a big difference.

Consider the simple Save button. Save means save, only save, and nothing but save; not save and close, save and continue, etc. Any other actions are assumptions only. A CTA that is labeled “save” should do nothing but attempt to save the input made. This has become better over time, but the *idea* here is still often misunderstood. The idea is that UI designers need to say what they mean, not what they think, and moreover, *not necessarily even what the users that tested it say*. Just because a few users think its ok, and “were able to work around it,” doesn’t mean the optimal design has been reached. In fact, if that’s what the users actually said, that’s the proof it hasn’t.

- Clarity of action (regardless of subsequent boredom in rereading)
- Clarity that something that may be expected *won’t* happen
- Clarity of verbiage (use the right words)
- Clarity of reading (large and clear font)
- Clarity of meaning (obviously different from other button actions)
- Concision (short but complete description)
- Jargon (can or should be used at times, limit triteness, depends on audience)
- Consistency (same actions have same labels or at least some have same words)

Considering the logic of the legal phrase “the truth, the whole truth, and nothing but the truth,” it is easy to see button/link labels that fall short of one or more parts of this triad. Another way to put it is what they do, everything they do, and what they don’t do. This is a good test for your buttons and links – try to imagine what a user will think about the text, not just for what is there but for what is not. And if a button cannot be labeled short enough to cover all the cases, either it must be rethought or changed to a text link which is not expected to be short.

The best way I’ve found to approach button and field labeling (and titling, etc.) is to start with a prosaic description of what the button does or where it leads. This description should take into account whatever the user may think – what their expectation is, what they don’t expect, and what they might be surprised about. We all tend to assign labels based on prior knowledge, domain knowledge, context, and assumption, rather than spending time creating a clean, comprehensive identity for the action, aiming for no misinterpretation by the user; in other words, approaching zero doubt of understanding the label. Notice I said aiming and approaching – there is no guarantee that you can achieve zero doubt in every user’s mind in every circumstance. The point is that it is worth the time to try. Here is a simple example.

---

<i>Full description:</i>	“Close this window (page) but don’t save it to final storage until they pressed ‘Save.’”
<i>Narrowed, keeping the important ideas:</i>	“Close & Return (Don’t Save).”
<i>Final labels (including all functions needed for the set):</i>	“Close (Don’t Save)” / “Save and Close” / “Save”

---

The final options leave very little doubt in the user’s mind. Notice that when all the options are realized that there is a context understanding in the button group. Without the other labels, the “Save” label may not be understood as not closing the window as well.

## Copywriting for Labels

Another prevalent concept in the industry is to have copywriters create/review button labels. Unless your copywriters are trained in UX psychology and label text creation, this could be a mistake. There is much more to determining label text than writing copy, as illustrated above in this section. Plus, most copywriting tends to follow business ideas rather than UX. I have argued with them many times about wording, and it was clear that they were more concerned with the business liking the labels, while UX was more concerned with users understanding what the buttons do. The best of my workplaces have been those where the copywriters understand that although there are certainly some common ideas, there is a vast difference between marketing copy and UX labeling.

## Internal and External Consistency

There is a sensitive and complex balance between consistency (both internal consistency - consistent pages, look, flows, etc. - and external consistency - similarity to other comparable sites) and maintaining business requirements. This happens especially when new ideas are introduced by the business. When such ideas are suggested, they may be at odds with consistency practice. When this problem occurs, UX people need to draw on their overall experience with the company (and its vision), with UX and users in general, with projecting on how much the new idea will be used in the future, and balancing the existing methods with the new one. Accommodating a new idea without concern for established standards will probably surface later as a considerable problem - the sooner decisions are made concerning blending the new idea with the old or dismissing it entirely the less trouble in the future. Consider also that other new ideas may surface, and then it gets incredibly complicated (been there) because the tendrils of each idea and the existing ones create too many possible confusing paths, which requires a great amount of effort to fix, often resulting in redesign.

## Wall UI Displays

I am and have been all for using walls for designs and flows - I've been a proponent and evangelist of wall visuals since the 80s as they have many advantages (if the viewers are at the wall location):

1. they show at a glance the organization of the site or site segment, and can be viewed by a simple turn of the head
2. no exporting is necessary, sharing is immediate and requires no device
3. the breadth and depth of the site or segment can be seen instantly
4. if kept up to date daily changes are immediately displayed.

A side advantage, which may or may not be desired, is that product management and other areas can view it.

There are three keys to using wall displays (Figure 9-1):

1. keeping it organized, with the addenda dated and initialed
2. having daily discussions on changes/additions/progress with the entire team and adjusting the wall accordingly
3. that the UX team understands each other and that while rank is respected, egos are discouraged.



**Figure 9-1.** Storyboard of entire project on a wall. Flow is down and to the right. This could also be considered a high-fidelity paper prototype.

## First Idea Visualization (Initial Sketching)

Whether you start with an idea or just start by doodling, I highly recommend the most basic method of visualization and communicating those visuals – hand drawing and writing. Pencil and paper (and other such hand tools) aren't just another way to begin a design, they are really the best way. This includes using marker and whiteboard, Post-its, crayon on lunch bag, and drawing in the dirt. The reason is simple - nothing has surpassed these methods for quick editability.

If you think about drawing to express ideas, the real benefit of speed is editability. The faster you can change text and linework, the quicker you can show the change and the less chance there is of losing the thought, losing your audience, and/or falling into the tool trap (where you are forced to spend time with or get more interested in how to use the tool than in getting your point across).

## De Facto UI Standards

We're all familiar with web page logos – their position and that they link to home. This familiarity didn't happen overnight. While still becoming an understood grammar component, smart companies, rather than depending on trends (in this case for no very good reason), still provide a Home button or link as well. There were many cases I know of personally where not providing an obvious affordance to get home caused the user great frustration. Some users just kept hitting Back, some reloaded the site, and worst, some just went elsewhere. Remember that most users will give up after three tries (and many after two). The sad part is it didn't have to happen. If it's really so awful and cluttered to have a Home button, then there is something else wrong anyway. What other reason is there? And even if testing shows that users are asking why it's there doesn't mean you should remove it. Be careful of 1) the definition of clutter, 2) the trade-off between good usability and clutter, and 3) doing things only because others do them.

Assuming standard UI manipulations in mobile app design has been a great source of frustration to users because the societal knowledge of such standards was assumed to happen too soon by many companies. A good example is the “swipe-to-delete” construct. Because there is generally no instruction, or precedent inferred by context, components like this are learned basically by word of mouth. Many companies feel that the cyberworld is so social that anything can be learned by asking your friends. One problem with that is some people don't want to appear unknowledgeable, and therefore don't ask. No, this is not their fault, it is clearly the UI designers' (or the business's). *There is no justification for unclear user interfaces.* In fact, they are worse than reflective ones.

## Futzers

Futzers are people who “mess with” (or “fiddle with”) a UI. That is, they keep trying things as just trial and error, usually with no structure to the method, and often go beyond what is necessary as well. This is not necessarily wrong; however, if *users* are futzing, it means a problem with the UI. If designers are futzing, it is probably (hopefully) just experimental probing. In either case the down side is time cost; the up side is creative flow (and fun), and sometimes discovery of something new. But in any case, design futzes must be tempered with good usability practices.



# Specifics

## Scrolling and Columnization

The assumed (and in some cases real) need for horizontal scrolling has been a bane for designers almost since GUI was born. This issue is a great example of getting users to understand trade-offs and also of user-designer confrontation.

The basic problem is that users want it all – they want all the data to appear on the screen, but they want to be able to read it too. Data-intensive grids of information in medical interfaces for example require many columns. The columns cannot all exist on the screen at the same time, even if using a small condensed font (which would become the next complaint). There are a few solutions to this, such as hovers that display subwindows with the row in question shown as a floating column, automatic horizontal shifting based on cursor position, “piling” data into multiple lines within the cells, or user-determined column location/hiding. The first question I put to users in this dilemma is “which columns must you see on the initial screen and next to each other?” If this is easily determined, the only problem left is the need to scroll to see the data to the right. If this is not acceptable (and the designer should argue that it indeed isn’t because the users generally won’t like it later), I suggest the hovering row-into-column popup idea. Sometimes piling data elements together in a cell (as mentioned above) is a good solution, like address information (city, state, zip on top of each other in one cell). At some point users must see that there is no real magic – even if all the columns could appear at once, users may require to have certain ones next to each other for visual scanning. The most important thing is to know what all the requirements are and get them first.

The user-designer confrontation issue is this: often the users will not accept that you know all the possible “tricks”. Armed with the information above, you should be able to win them over and get them to accept your suggestions. If you offer enough choices, they won’t have time to try to “solve the riddle” themselves. Otherwise be prepared to explain why their strange ideas won’t work or won’t be as accepted after some use.

- What are the columns needed and their header text?
- Is horizontal scrolling acceptable?
- What columns must appear on the initial screen?
- To what extent can abbreviation be used?
- What columns must appear next to each other, if any?
- Can some elements be “piled” in the same cell, and how does this affect visual scanning?
- Is viewing a row vertically (temporarily blocking out other data) acceptable?
- Is user-controlled column placement acceptable?
- What is the priority of the columns – i.e., which ones are most used?



Depending on the nature of the data and application there may be other ideas as well, such as an automatic scroll when the mouse is moved to the right. These are the important basics to keep in mind: columns required initially; columns required next to each other; row height restrictions; and whether other data can be blocked out while viewing a row's information in a popup.

Vertical scrolling is not as much of an issue. People have come to accept it. Maybe this is due to centuries of viewing tall format newspapers and portrait oriented books and documents. However, this doesn't give license to creating lengthy pages (even when dealing with data lists). The issue here is that users don't want to try to remember where in the list they saw that last row they were concerned with. If they want a long list, offer them some kind of marker to highlight a row so they can find it again quickly. A scratchpad may be another idea – they can drag and drop rows into it, and visually compare elements. Shortening a list's viewing area just to keep it short is not always the best idea either. Making users bounce between pages or use some special scrolling technique is worse than making them simply scroll.

In both horizontal and vertical scrolling there is no single-bottle snake oil cure. Each situation must be carefully examined and questions must be asked up front to save time and effort. This is a great example of the fact that UI design is not just calling up pertinent rules to follow.

One important idea to remember is that users may want to compare the cells of a column (for example when comparing prices). This may require placing other pertinent columns close by requiring no need for a horizontal scroll (either by device or even by eye). Customization of column placement is what is required here if the columns the user wants to see aren't viewable in the same screen. This of course is an extensive undertaking in UX, design and code, but may be worthwhile if used in various pages and if determined to be needed for the life of the software/site.

## Date/Time

You would think a “universal” date/time set can be used anywhere, but the full set may not be necessary per the requirements. For example, how far back in years should be allowed for input? When dates are about currently living individuals, there is no need to scroll back to the 1800's – but if it concerns a range in history it may need to go back further. There is also the concern of what date components are necessary – full month/day/year or just some subset? Is time necessary? To the second? And so on.

Handling the interface must also be considered. Are pulldowns (or mobile scrolls) the best way for the specific need? And what are the defaults per component, if necessary/sensible? The bottom line to interface components is to make them easy to understand and use, without getting too clever and with consideration of consistency with other forms/pages and even other sites. There is no one size fits all for all date input.

## Progress Visibility

Progress bars and other such techniques have been one of the most maldesigned affordances in UI history. Some are done correctly of course, but it is difficult. Progress bars that keep refreshing, even with accompanying text, are useless without showing an overall progress bar with them.

Most progress components show progression of loading from a server or from offline, and as such are subject to the limitations of the foreign base, transfer speeds, and so on. This makes it hard to really show accurate progress. In other words, it reflects what was done, but not that what was done was really the visual fraction of time shown; half the data may be loaded but the bar may only show that a quarter of the entire process has finished. Often the bar seems to freeze for a while only to jump later, not really helping the user (did it stop? is it going to go fast again later I hope? and so on).

Remember that progress components should reflect three things:

1. how far into the process you are (what has been done)
2. how much there is to do in total (data or time or both, and specify)
3. whether the function behind it is working or not (movement)

There is an inherent 4<sup>th</sup> concept: that each point on the bar shows how much time has elapsed, not just how much of the function is completed.

A good rule for this is to avoid using progress bars that don't accurately reflect time or amount, and to simply show something animated to reflect that the function is working. It may help to state that the progress is about data transfer, not time.

## Progressive Disclosure

A proven UI method to keep the screen clearly understandable (as when searching for example) is to reveal subordinate selections from the user selection of a higher level item. This works well with large menus with selections that fall easily into subordinate groups, as exemplified by some pulldown/flyout menus. But some of these menus also show how such groupings can get out of hand, such as with large ecommerce sites that have a huge amount of products. While we all try to limit the pulldown choices across the top (without piggybacking them as rows which yes is too confusing), the result, most often due to the desire to have no more than two selection areas (the top menu selections and the resultant box from each), may still be confusing. While keeping the number of areas to a minimum, the result could be inundating the user with a hundred choices in the pulldown box. In other words, there is a need for another interim level. Some ecommerce sites do this, and it works much better.

But product plethora is not the only place this problem can occur. Any site that has a huge number of items, whether for purchase, download, reading, renting, whatever, is going to need a clear, uncomplicated way to dig down through categories to those items. It can get even more complex when users want various categories – for example, in movie

rental some may want genre first then director, some may want actor/actress first then availability, and so on. I've seen this handled a number of ways, the worst being frozen categories (no choice – genre first, then actor, and so on, whether you like it or not). This is a bad business decision, but understandable if the alternatives cost a lot of product code and design work to offer some level of customization. However, I have seen that biting the bullet and doing it the better way the first time usually pays off, especially if the main charter of the company is to rent (or even sell in the future) movies. That's not going to change much in the near future. And supplying just any old way to get to market may be a marketing detriment that won't be easily overcome later.

Progressive disclosure may be used in searching also. (See the section on search tools.)

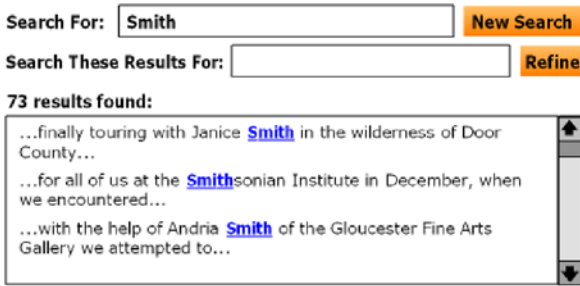
## Search Tools

Users may not be able to navigate larger sites easily. Even if they can, they may not think they can, and therefore without help their frustration level is increased through a bit of anger brought on by fear.

A site map or site index can be a handy help, but only if it is easy to find and not too confusing. Another idea is to break down all your site content by obvious categories, or else by task groups, and visually separating these chunks on the page by color or shading or white space. Don't be afraid to duplicate links either – certain topics may be represented by different user words. This duplication could cause a maintenance problem, but you can use a link check tool to make sure links aren't orphaned. I suggest asking the users, maybe after preparing some options they can consider.

A search box is handy if correctly implemented. Be sure the user doesn't mistake it for a search tool that applies to the entire internet. In fact, in most cases it doesn't make sense to put an internet search tool inside a site. It is better to provide a button or link to open a new window to some preferred third-party search site if you think it applies, a search box that has a label or affordance that mentions the third-party tool, or a clear labeling that the search box is not just for the site. Also, consider what the user may be searching for. From your research and/or your educated guesses you need to determine if your users' search queries will be single words or longer. This can make a difference in how you create the search tool, or in what type you want to purchase. Usually, if single word search is good enough, you may consider taking care of that with an alphabetized site index alone. Remember too that there is no law against having both a site map (organized by topic – mimicking your nav) AND a site index (like a book index). This of course is especially applicable to larger sites/apps. If you think a more complex search tool is required, be sure to consider user help for it and consistency with other search tools. In any case, make it consistent with the style of your site.

For large applications where there may be a large quantity of search results, consider the user's ability to refine the search. Most importantly, make it very clear to the user that the refinement will be a subset of the group already found. Remember they may want to simply start over as well. Each of these ideas should have their own clearly labeled affordance, as in Figure 9-2.



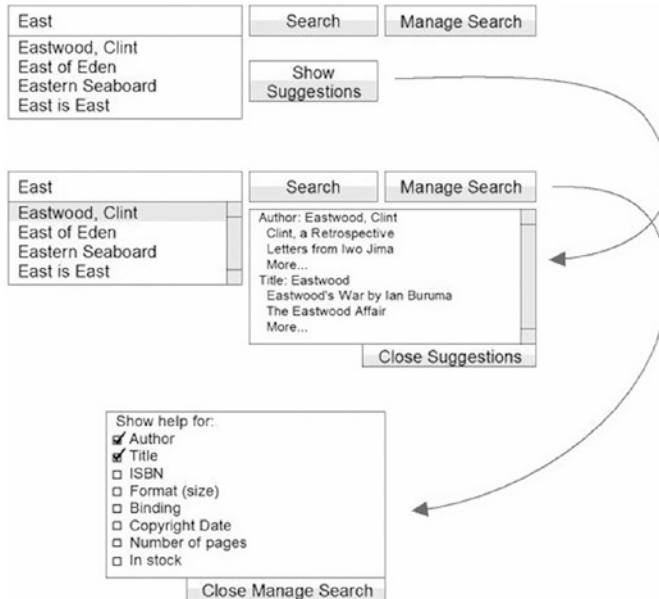
**Figure 9-2.** Search results screen for large number of results

For public sites that have a good number of possible search criteria, it is important to remember that most users don't care to be inundated with options. However they should somehow be offered. One way around this problem is to allow for basics first and offer ways to see more filters or more advanced features. For example, for finding books, the first level (for everyone) is to offer typing in a title or author. The next level, achieved by a button, could also provide input for copyright date, publisher and ISBN number. This would appear directly next to (or underneath) the first level, looking like an extension of the first one, which it is. The next level could offer size, type size, or bindery. What options appear in which levels would be determined in design by statistics and/or experience, and could of course be tested with users.

Automatic search or type-ahead search, where the system starts offering a list of a fully typed search lines based on the first few characters typed, can be helpful especially for the typing-challenged or on typing-difficult devices such as mobile devices.



A further enhanced experience could include suggestions for books based on one criterion such as author. This is called progressive disclosure (Figure 9-3). Simple buttons allow for deeper drilling without showing all possibilities at once, so the user chooses the depth of complexity. In a list say to the side of the search dropdown such suggestions would be shown with their corresponding reason. In this example one would see other books by the same author with the author's name above them, and possibly other books with similar titles with the words "Similar Titles" above those. If this should get too involved, there could be a button (maybe saying "Manage Searches") that would display a dialog of all such search suggestions and a checkbox for each to turn them on or off.



**Figure 9-3.** *Progressive disclosure*

The key here is to let the user choose what level of complexity s/he wants and show those levels only as desired.

## Discovery and Hidden Affordances

Discoverability is finding some hidden affordance, found only by trying clicks, taps, and swipes on areas with no (or no obvious) visual clue. This goes completely against the good UI practice of making the UI disappear (Norman, 1990). There may be some very specific instances where its use would make sense, but not in normal UI design. Generally it is only good for things not necessary to the flow of the user goals.

That said, exploration can be encouraged, if it doesn't hinder the users' flow to goal achievement. But exploration is not about hiding things, it's about finding all the things that are right there but maybe not very obvious to the flow or goal in question (by the same token too many things at some point just become clutter and an inadvertent admission by the whole product team that they are more concerned about quantity than quality). This multitude of features of course allows for loopholes, so designers need to police themselves with *constant* awareness of good usability.

One of the worst justifications for poor design is the idea of discovery. That is, letting the user discover things that are hidden, usually due to poor design, in turn due to limited space on mobile screens. Consider for example swipe to delete, where you swipe left on a row of data to reveal a delete button that can be tapped to remove that row. I found this out myself the hard way – by watching someone else. Not good.

It seems designers are going to assume everyone on the planet knows a great library of unobvious and hidden affordances (in mobile especially) which is growing daily – used in general to avoid difficult designing to do correctly that would provide all users (including new users) enough information to accomplish their goals.

Do we just not care about new users? Do we assume that people with phones are all addicted to social media enough that no one will miss out on new hidden things due to constant competitive showing off of discovery of those things to their friends? When you are older and you work with younger people, you notice such attitudes. But the problem here is that the some groups are not caring enough about other demographic groups that make up a good percentage of their market. We can't allow that to happen if it infects good interface design.

## Parallax Scrolling

Parallax scrolling is a visual effect that gives an illusion of depth by, for example, making the background elements on a page scroll slightly slower than the foreground elements. This is a great example of how we must consider carefully the use of trendy or cool UI concepts before jumping in. Not that they are all bad – the concern should be ignoring good practice over displaying eye candy (which usually doesn't last). For example, consider load time, visual time, distraction from content, development time, target audience eyes, fit, and site and/or external consistency. For further help search the web for parallax or see <http://www.uxmatters.com/mt/archives/2014/11/parallax-scrolling-attention-getter-or-headache.php> and/or search on parallax ux.

## Double-Duty Affordances

Some affordances are designed to do more than one thing, acting as a toggle to another state not labeled (meaning users must know intrinsically what that other state is). For example, a button labeled “show” is expected to change to “hide” once clicked/tapped, with no other label possible in anyone's mind. This is only a good idea when there are two states and they are more than obvious. Testing that will give results like “80% were able to figure it out” – elsewhere in this book I mention that “able to figure” means the UI failed. Consider “grid view / list view.” It is not obvious to users that the other state of “grid view” is “list view.” Icons and labels would help here and one would be necessary, but display them both with some visual method that makes it obvious which state is on. For another example, consider a set of switches that can be turned on or off individually or all on collectively by one switch labeled “all” (the switch is disabled if all switches are on). If all the subordinate switches are on, the “all” switch is disabled. When the user selects any of the other switches off, the all switch is enabled. And so on. The key here is to not be clever or artsy – make sure the user knows what is going on at a glance without trial and error.

## Skeuomorphs

A skeuomorph is a visual affordance that looks and/or acts like its 3D counterpart. One example of a poorly designed skeuomorph is some of the mobile app slide switches: you can't slide it, although it copies the look of a switch that slides; rather you must tap it, so reflection is required after initial failure, which means it has failed as a skeuomorph as well as a UI component: eventually (warning word!) it's action becomes universally understood, although it will still fail for new users. The point is it should never require an "eventual" understanding.

Skeuomorphs must both look and act like their real counterparts to be effective (see also the definitions section).



## CHAPTER 10



# Research and Testing

Many feel that the process from research to design to product is cut and dried, and that the steps employ simply defined and obvious tools. Not so.

## Problems with Research

Trusting too deeply in your (and others') research technique(s), questions, tests, and so on may not reveal what is really needed. Most companies need to put more resource effort and time into proper research methods rather than assuming it is a small step and easily done because we are all born research experts. Beyond the research methods, there is the analysis, which can also involve error if done too quickly or haphazardly. Don't underestimate your research step. I'll add here that I'm not just talking about acceptable margin of error, but rather the margin of error that is a concern prior to that – in other words, beyond justification based on sample size.

Often we come upon the need for affordances that we assume are standardized, but aren't, and probably should be. A good example is the date input set. I have seen this handled in many ways; most bad, some very good. But the important point here is that the initial assumption can be quite wrong, especially in view of the specific need in the interface.

The same is true for research. Be careful of initial assumptions from the numbers. Just because 54% of users like it one way and 46% the other may not be cause enough to select the first. In fact, such a close call really means you need something else to help you decide. Maybe the two choices aren't enough. But many UX departments grab the higher number because they know the left-brained business people are number-comfortable, and will never question the higher one (and feel safe that those higher in rank won't either).

As with testing, we all seem to think that we know how to research, how to organize the results, and how to interpret the results. Anyone can discuss topics with people, very few can make sense of the findings.

## Time

In Copernicus's and Galileo's times, if you did research to see who believed the earth was the center of the universe, you would have achieved near a 100% "yes" response (mostly honest opinions, but partly said in fear of social reproach). If you said that today of course you would suddenly feel more room around you. Research is therefore a function of time and society, and the results, even when properly analyzed, usually have a value only in the short term. There are ways however to prepare your research for short- and long-term understanding, but this comes with years of experience, and, sadly, it seems businesses are not interested in the long term.

## Concision

We tend to think fast and talk fast in business, as though we haven't time to really understand and evaluate subjects, combined with the egotistical slant of understanding everything in context as well as feeling forced to. I can't count the number of times someone (or I myself) got lost in a conversation because we didn't want to admit not knowing some phrase, word, or acronym. This has four resulting routes: secretly looking it up online if possible, stopping the conversation and asking, waiting until after the meeting to ask someone you know, or never asking and costing time and money (and face) later by using a guess that may be wrong. Concision also involves mistakes due to the speed of research performance. We are usually happy with some limited understanding because we are too busy, too bored, too hungry, or too ready to go home. Often this is a result of either poor research planning or an erroneous projection of the time it would take. The latter can only get better through experience. The former can be avoided by writing out a plan and having another person check it. Yes, it's worth the time.

## Research Plan

Often research is done without any planning. That only works sometimes.

Creating a plan for your research can be simple. We can borrow from plans made for grant application and other types without getting too bogged down:

### ***Do's:***

- Focus on the theme without sidetracking
- List out what you're really trying to find (root analysis)
- Be realistic in terms of schedule
- Provide some possible ways to extend the schedule if it should become necessary
- Have some idea of results limits (when you're really done, not when you want to be done)
- How many people or organizations, etc., will be reviewed/contacted?

***Don'ts:***

- Don't include any former results
- Don't be blind to, ignore, or justify your own biases
- Don't feel swayed by general results before completion (may cause bias)

## Audience Spectrum

Even with a focused audience, it behooves you to spread your coverage across demographics (age, gender, race, education, etc.). Often some results from outside your intended audience can shed light on things you may want to consider in your product design. Consider for example that a product you are purposely designing for one group is working well for them, and later deciding that, with a few tweaks, it may work for a broader audience. Remember however that your research, no matter how broad, does not cover everyone, and is not the end-all know-all.

## Eye Tracking Results Analysis

See the section on eye-tracking in psychology tools.

## Problems with Testing

Following are some of the problems in general:

1. First is the assumption that user feedback can always be interpreted to mean what was said, includes everything the user wanted to communicate, and is assumed to be comprehensive.
2. UX testing trusts users to know exactly what the buttons mean even though they can be misinterpreted (see sections about labels).
3. Tests assume environment and other conditions mean too little, such as the room, how many others are there, the noise level, how close to real usage conditions the overall situation is, and so on.
4. There is also the simple thought that the user performing the test knows it is a test, which is a condition applicable only to the test and not to reality.

5. There is also the idea that *the user has no stake in the matter* (no money is being spent, no work is being done, they are told there is no right/wrong answer, they are being paid to test, etc.).
6. Tests are also brief and singular. They are usually taken only once, with no repeating. We all know that a second visit to a site involves learning from the first, remembering things not called to mind in the first, and possibly a different view due to visits to other sites in between. I have often changed my mind about something in the first site on the second visit. Singular testing is also a breeding ground for initial false positives.
7. Testing methods aren't tested like products are. We all assume that if it's in a book or an article it must be good and therefore can be adopted. Many UX departments and individuals use testing heavily, or maybe "intricately" is a better word. We all feel we need to rely on our site/app, and we use testing to increase that reliance. But we don't test our testing methods for the same reliance - we just assume the method is reliable.
8. We also believe that we all know how to create tests. We rely on our education, articles, books, and trends to gain the false assumption that all our methods of testing and test creation need no review. In one of the larger companies I worked for, which had over a hundred UX people, I was always asked for my research findings before my comments were regarded (even though I had 30 years of experience). In that same company, the testing methods were not questioned (see #7).

In-person or online testing can carry the same issues.

I have also seen test results change from day to day even though the same users were involved. This is no surprise if you think about it, as interfaces are usually complex enough to offer various routes for the same tasks.

There is another inherent problem to testing. This is an example regarding ecommerce flow. Let's say you want to test a flow button within any of the cart screens (overlay or full page). There is generally an overlay for users to view their cart 1) maybe after you've added an item and 2) if you want to view your cart without beginning the checkout sequence, to just be reminded, to edit the cart, or to see it prior to intent to checkout. The button label "Go to cart" from #1 is intended normally to cover "View" and "Edit". All okay, but if you added to your cart with an "Add to Cart" button, the only reasons you have to go to the cart is to edit it or to start the checkout sequence. Some sites offer "check out" on every page once the cart has something in it, and an "edit cart" button as well, identifying the tasks explicitly to reduce reflective cognition. I wouldn't say this is better in all circumstances, but should be considered.

The testing problem here is that seasoned users, already used to the shortened way ("go to cart"), won't see any great advantage to splitting the tasks to two buttons. But there is, and not only to new users. The key is attempting zero reflective cognition, without any added congestion to the UI.

Some of these issues can be overcome, and of course some can't. But all of them should be considered rather than assuming they will always take too much extra time. Your customers/users are always worth a few extra minutes of such design consideration.

## Test Creation

It seems taken for granted that we all know how to build a test. We need to realize that 1) if we weren't educated in test creation we may not know it, and 2) we are all biased and usually don't know to what extent. I have seen tests that assembled such apparent predisposition that it almost looked like a setup. Apparently the test builder(s) either were too inexperienced or didn't realize their own biases. The best way I know to avoid such issues is to have an impartial viewer check the test out before performing it, looking only for bias in the test design.

Of course there is always a cost in time (and money) to building and adjusting tests. In my experience this has nearly always been justifiable, especially since a bad test propagates its problems into the released product which just resurfaces later, after some users have had to suffer unnecessary frustration.

## A/B Testing

Much A/B testing is quite valid, but not all. Very simply, if given a choice of two bad things, users will still pick one, without considering that they might both be bad, or even equally bad. This has to do with one of the many cognitive biases. We tend to accept that the group presented is all-inclusive, especially when presented by experts in a field in which we are not experts.

That said, there is the condition of "option science" (see that section in this book) that suggests there are four options when presented with two – either, none, or both. There is also the possibility of added ideas often resulting from conscious thought brought about simply by seeing the other options. However, when this happens, there is an impetus caused by realizing the late stage of the process that suggests just opting for the combinations and/or new ideas and plugging them in. This is a point of possible error or failure, since the combinations and more so the new ideas must be blended into the prototype which in turn must be retested.

## Test Results Analysis

Another assumed understanding (which doesn't always happen but it's been noticeable to me) concerning the testing process is how to analyze test results. This falls into two areas: interpreting the analysis and deciding on what to do based on the results.

As an example of interpretation issues, in a test that compares photographs, content has an inherent bias (or at least invokes a bias). Two pictures of models modeling clothing can rarely have an objective comparison of just the clothing. Assuming the models have comparative figures and heights (as relevant), there are other visual factors at play, such as face, facial hair, expression, hair color, hair style, skin tone, even eyebrows. Beyond that, just the pose can have an effect. And yes, these things can often enough influence a user's decision to purchase.

Concerning what to do, I have been in companies whose decisions were based on percentages with a margin of 1%. For example, one test I witnessed revealed that 34% of users liked A, 32% of users liked B, and 33% liked C. The decision was made that A should be used, regardless of how close it was in comparison to the others. With a margin of even 5%, these findings would all be equivalent, and either further testing must be done, or, assuming the test was valid and complete, resident UX experts should make the call.

Another anomaly is speed. Why spend time analyzing and retesting something that will probably become moot very quickly, due to reasons of near-future changes such as the knowledge of other requirements coming down the project pipe. This can breed laziness for subsequent occurrences, as it gets pushed further up the list of things to consider the next time. That said, maybe this is just another item to educate the business on concerning UX impact.

## What Do We Test for Concerning UX?

UX testing is very worthwhile, actually vital, in its various forms and at various process points. But differences in products, audience and UX team strength must be considered for any testing regimen, and these things should be considered up front. Ask: what is the UX testing for? My experience tells me: 1) how users deal with the content, 2) how they deal with flows, 3) how they deal with the visuals, and 4) how generally usable/valid the designs are. The content (nature) may be complex or simple; the flows may be obvious or not; the visuals may be clean or obtrusive; the overall usability design may need help on its own. If most of these ideas have simple answers, maybe limited testing is ok - if the product is large and/or complex by nature then more or deeper testing (or testing more often in the process) is needed.

## Usability Testing Lab Environment

I was involved in a discussion on how to decorate a usability lab. Many suggested a wall (and appointments) color based on color theory, such as some red to help users think or beige to make them feel less surrounded, etc. One issue with this discussion was that sites, apps and dedicated applications could be used in varying places. A site and app could be used at home, at work, on the street, or in the store. Some dedicated programs, such as used for public kiosks, may have a real environment that is economically unfeasible to reproduce. Such things as noise, prolonged standing, uncomfortable seating, too much or too little light, and so on, can affect a user's decisions. Also consider that when the user is out in public, or in the weather, there may be a sense of urgency.

It would be impossible to provide all these things, but they would give you the truest picture of usage possible in a test. The bottom line is you can't reproduce a true environment because 1) users' real environments vary greatly, 2) the user knows it is a test, even if it was being conducted at home or work, 3) the user has no stake in the outcome, as he/she does when actually using the site/app, and 4) in most cases the user performing the test is getting paid.

I would suggest just making the lab room pleasant. Add a little noise (perhaps only at times) and a little color. But don't worry about it because it won't be perfect. The best testing lab is that of a familiar environment to the user, alone or with friends, with a stake in what happens.

## CHAPTER 11



# Hiring UX

There has been much discussion, and it continues, concerning hiring – definitions of titles and the sad lack of UX understanding “out there.” Some business heads don’t want to spend time researching to understand what UX is and who does what and who doesn’t do what, so apparently depending on their own ideas and their desire to look good to the boss (by saving money on associate billing), they put together a job description that has no regard for common sense or logic and trust in their own thinking that anything that is front end is UX, and therefore all of that can be done by one person. More left-brained hiring managers also feel more comfortable by including some kind of development in the position (sometimes even back end development like Java). But all that said, to be fair, where are the titles? Where are the definitions? Who does what? Why is this so complicated?

Listing definitions of titles could probably be a book by itself. However, since I’ve been in the work place almost half a century, I’ve seen the definitions change and more or less solidify in the past few years. Table 11-1 lists what I see out there. Note that there is some overlap, or can be, and this adds to the confusion of hiring managers; so my recommendation is to determine the functions needed first, then match them to the appropriate position. This would also help write the job description.

**Table 11-1.** *Defining titles*

Function/title	Responsibilities
UX Researcher	Know how to research (not simple) and understand research bias; research what is happening; ask what should happen; organize findings and present options and suggestions
UX Strategist	Develops a strategy (and/or sub-strategies) for the user experience based on business requirements; acts as liaison between the business and UX
UX Architect	Analyzes and designs the flows of the site/app, the navigations, the page arrangements and overall experience (goals, tasks)
UX Designer (UI Designer, interactive designer)	Builds the page with UI components; builds the interconnections; maybe more graphics creation/editing
Front End Developer (UI Engineer)	Builds the functionality of the site/app in front end code (html, css, javascript, etc.). See note below.

Note that front end coding is not the same as back end. The best explanation I've found is this: programming languages, such as java, manipulate the information; languages like html (and style sheets or css) are for markup – they manipulate the *display* of information. Javascript is the oddball of the languages, as it can manipulate the display code (for example it can loop to display different data as table rows and call the server to access a database for that row data). Other programming code, like asp and jsp, are really back end manipulators that reside in the html page, or front end. The best way to remember this is there are languages for markup and languages for data manipulation, and that some data manipulation languages can reside in the front end (html page) as well as the back end (server).

UX isn't a simple thing. Not only are there so many parts, like strategy, research, testing, design, and architecture, but it is a psychological pursuit, not just some mix of engineering and art, as many business people seem to think of it. It is not a segment of the development process like coding is – it is part of every segment (user interface design, including wireframing, can be considered process segments, but UX is more than the HCI or UI). Even when some are told it is more than that, dollar signs appear in their eyes and it is more comfortable for them to group it all together into one position (see last paragraph) and/or couple it to engineering or marketing in some way. Granted business people are limited to budgets and pleasing the boss/investors, but they wouldn't give such treatment to a position such as, say, lead developer. And this is not difficult to see, since programming has been around longer as a discipline than UI design has (in the workplace) and is easier to identify, as its baseline is coding instructions into a computer. What instructions? Those made by the business, not the developers. Extending this into all process segments, it's easy to see how the newer concept of user experience design was thought to be done by designers but was really created, to some degree, by the business. Because UX is often misunderstood in business, it is taking a long time for everyone everywhere to understand just how UX fits.



The stigma of the front end being less important than the back end has existed for decades and its falseness is being realized very slowly. The sooner everyone realizes that both front end and back end are complicated and important, and the symbiosis they share is as well, the sooner UX will be recognized for what it is – the overseer of product development and customer view into the company, as well as the greatest competitive edge.

So my recommendation to a company is this: do your homework and figure out what UX is and what its parts are, and understand what is *not* part of it. Then figure out what processes are involved, and what positions can have combined responsibilities, if any can. As far as which should be in house and what can be farmed out: in my experience I've found that it is better to have all UX positions in house and at the same location if possible (and it is better to have all production areas and UX together at one location as well).

## CHAPTER 12



# The Future of UX

The following are just some comments of where I see UX heading, based on 35 years in the field and having kept my eyes and ears open in the duration.

Over the decades I have noticed an increasing trend toward speed, coolness/ cleverness, and social acceptance. Rank power enforces the dubious value of these ideas. Fear of losing your job often causes poor judgment. The result of these is that we all as a race lose time while the gears of confusion generally grind away in futility, and we must depend on the smarter companies to confuse the rest with attempting to understand and think before jumping on the bandwagon.



Good examples of the problems of just complying with trends would be:

- The cute, tricky UI components which seemed like a great idea and were clever and visually interesting, but fell out of favor in a short time. This despite the fact that their initial use (or at least initial influence) was by a very well-known company often followed even in UI concepts due to nothing more than their high market share. And just a note here – sadly, this UI fallout did not cause other companies to stop looking to their UI influence.
- Discoverability, which allows UX designers to avoid thinking about where to put affordances and explanations on the page, especially in small mobile screens, and force the users to discover how to use the components they were comfortable with. And then incorrectly justifying this with “less is more.”
- Assumed knowledge, similar to discoverability, where it seemed justifiable to assume the users knew UI concepts and components because they “have been out there so long” – where “so long” means over 3 weeks. This is the “false consensus bias,” which basically means our unconscious assumption that everyone thinks alike, and there are never any new users.

- Increased valuing of initial first impression through eye candy and cleverness, to sell the client or the users on the site/app at the very start, regardless of how trite or cumbersome that candy becomes to your online teeth after only a few revisits.
- In general, jumping in to taking advantage of naïve or gullible users to trap them. Of course, this is always accompanied by some false justification, like such-and-such company does it and it will be ok in the long run. The very good news here is I have heard a number of users say how they don't return to that site even though they thought it was clever or pretty.

# CHAPTER 13



## UX Tips

Here are some UX gems I've unearthed along the way.

A UI that causes someone to think, even a little bit, has failed.

-----

Examine trends for usability and make educated guesses as to their lifespan before deciding to incorporate them.

-----

Similarly, don't let design take the place of usability. Learn how to combine the two. Design lasts for the short run, but can last for the long run if usable. Usability is always for both the short and long runs.

-----

Discovery is a crutch; it allows for poor UX design and its only fun for people with time and something to prove; and even then, only for a while. Donald Norman said that the best UI is the one that disappears, and in my decades of experience I've seen nothing to counter that statement.

-----

Above the fold is not a myth. It's what the user sees first. Not only does it provide the first impression, it is expected to house the ways to start paths to any goal users may have.

-----

The business isn't always right. If you cater to your customers the business will benefit in the long run. Research in the short term may suggest otherwise, but research in the long term will prove to have the advantage.

-----

The easiest physical interaction mechanism is not moving your finger, it's moving your eyeballs.

-----

Touch screens require eye contact, tactile interface components do not (unless poorly designed). Vehicles for public travel should not have touch screens, as you must use your eyes for feedback. Most such ridiculous ideas spawn from user input – proof that just because users want it, and will pay for it, doesn't mean it's the best thing to do. I'm surprised and sad that I actually have to point this out. Hands-free is not the best restriction for safe driving – eyes-free is (don't take my word for it, just think about it). Mind-free, coupled with eyes-free and hands-free, would be better, but has some obvious issues.

-----

Make sure everyone understands the definitions, in detail, of words/ideas being discussed.

-----

Statistics are usually fine, but they're just info; knowing how to interpret them and how they fit into production is not as simple as most believe. Also most statistics carry some type and some degree of bias – intentional or not.

-----

Differentiate between the experience, the tools for designing the experience, and the tools for executing the experience. In other words, the journey, UX design/architecture, and coding.

-----

UX should be a department on its own, not force-fit into marketing, product management, or (gasp) engineering. There should have been CXOs years ago.

-----

Experienced UX people know the nuances of UX rules, history, process, and how it fits or should fit into the business. Take advantage of their experience – learn so you don't make the same mistakes. This is another point I'm surprised I have to point out. Experience is the most important thing concerning knowledge, processes, and production - it supersedes formal education, social contacts, salary, and attitude. An interesting thing I noted during my last years in the work arena was that those with more experience did things a bit more wisely, as opposed to faster, more socially acceptable, more peer group oriented, or with more strict adherence to the status quo. They had years of research behind them, and because of that they could second guess much of the testing that was done by the less mature in the discipline. They automatically designed good usability into their work as they just couldn't help but do that. They also forecasted decisions made by the business or the UX group with a great deal of accuracy. These workers know the real value of trends as they watched so-called benchmark sites make changes for years. I highly recommend employing some amount of highly experienced UX people and allowing them to hover so to speak over all the production work, acting as managerial number ones if not in management positions themselves.

-----

Copying what other sites have is a two-edged sword. One edge is maintaining universal consistency if the copied concept is prevalent. The other edge is promoting a wrong method and adding to the confusion of acceptance (and even elevation) through commonality. All that said, pure creativity can inadvertently result in copying if some research on the concept isn't done first – which isn't wrong, just probably not desired.

-----

Companies should take the time to learn what UX really is instead of assuming they know, and do their homework before creating positions that accomplish much less than what they wanted.

-----

The more tools and trends available, the easier it is to make a UI, and *the easier (and faster) it is to make a bad UI.*

-----

No complaints doesn't mean all is well.

-----

Consider your users – aim for zero doubt in understanding the UI.

-----

A company's website does not belong to the company. It belongs to the company's users.

-----

“Less is more” is an often misunderstood and abused axiom. The reason is quite simple: how much is “less”?

## CHAPTER 14



# Notes on Online Discussions

Often some of the online UX discussions go on for months. There is a point where we need to sit back and look at the whole thing, and ask what the problem is. Online discussions seldom end by resolution, but when they last too long without abandonment it usually means there is some complication as well as an extensive interest in the topic. This is one reason for this book – to suggest a solution, or a method to find the solution.

I have also noticed that of the (thankfully) infrequent disagreements about causes and methods and more, the differences are usually between age groups. The younger people generally think they know but lack the experience, the older people do know but generally lack the ability to convince the others. As my career progressed, I noticed this get continually worse in the workplace. Rank, attitude, and yes education are no replacement for experience.

Online discussions bring up interesting questions and some members offer their solutions to things. But it should be understood that online boards are not a way to get things resolved. They are for sharing ideas, not just promoting your own. You have to listen as well as speak. Also, they should not be just whine rooms. While venting your frustrations to those with similar backgrounds is often relieving, others are looking for help, or at least for some ideas for resolution to their problems. I suggest we leave the emotional stuff for the more purely social media rooms.



# Summing Up: What Does All This Point To?

I've noticed over the years that UX rules and procedures concentrate on the mechanics of architecture and design and making sure the business is happy and the designers are free. The UX area must still justify any attempt to satisfy users without research numbers, especially if presenting something different from all the common, overused UI out there already. While not a pure cause and effect relationship, it seems that the rules and procedures gradually lost focus on the users' needs/wants/expectations and their welfare. There also has been a loss in honesty, I suggest mostly due to competition.

Instead of using dark patterns and other types of manipulation, is it really a stretch to say that honesty (not just ethics and legality) is something users would recognize and appreciate? And so much so that it would be retained and associated to a company with a positive note? They would have to see it, simply because of the lack of it out there. Remember that honesty is not just refraining from lies – it's telling the truth and the whole truth as well. It's being up front with your users with the assumption they are smart, and showing that you assume they are smart; not clever, smart.

One way to employ honesty is to be considerate – thinking of your users like your family - children and parents rolled into one. What the users want is very important – what they *should* want is even more important, although quite tricky to provide properly. As I mentioned elsewhere, it is not user desires we should be so concerned with – it's user welfare.



# Glossary

## Affordance

Every component of a UI is an affordance in the strictest definition of the word (they afford the user something, even if just text to read or pictures to view), but we're concerned more with purposely interactive (choice-making) affordances – some representation of a clickable or tappable area, such as a button or link. An image can also be interactive. A good affordance is one that all users will without reflection recognize as clickable at first sight.

## Breadcrumbs

A set of affordances (usually text links) separated by right-facing sharp braces or arrows that represent pages previously visited in order, usually situated near the top of each page.

## Cognition

The way we think and learn. There are two types:

- Experiential: requires no thinking; understood without reflection
- Reflective: requires conscious thought effort

## CTA or Call to Action

An affordance often referring to a user option to make a selection or initiate or continue a flow.

## Disabled

A state of an element. A disabled element (e.g., a control) is often “grayed out,” but still appears and is readable, and is obviously not interactive. Opposite of enabled.

## Discovery, Discoverability

The ability to find components and task flows that aren't displayed in the UI.

## IA

Information Architect or Information Architecture

## ISD (ID)

Instructional Systems Design; Instructional Design

## Metaphor

Created graphics, flows, and other elements to mimic well-known objects or processes because of close association. See skeuomorph.

## Modal

A subordinate window requiring some action before closing and allowing continuance. This term has been incorrectly used to mean any subordinate window. The opposite of modal is **modeless**. A modeless subwindow can coexist with the parent, that is, there is no need to complete something in a modeless window in order to continue in the calling page/window – both can be open at the same time.

See also subordinate window, overlay.

## Model

The perception of a system or set of tasks to anyone involved in the system or its creation/change. Generally there are the user's model, the developer's model, the designer's model, and the management's model.

## Overlay

Any window appearing subordinate to another, often called a modal or popup (however, see Modal). See also modeless, subordinate window.

## Perceived Authority

The importance of a position as people see it regardless of its true place in the organization or hierarchy.

## PLA

Principle of Least Astonishment. This simply means that the IAs, UXAs and designers must endeavor to make any concept, visual or flow step have the least surprising effect on the users.

## Progressive Disclosure

A design concept that presents complexity of choices (as in searching) in visual steps or sections rather than displaying all such sections at once.

## Prototype, Paper Prototype

A prototype in UX terms can be anything from a visual-only mockup to a partially interactive site/application. It could even be made by voice. Digital, interactive prototypes can be shared and tested by others and look and act like the final product but without data connection so no content changes. They can even be built so data appears to change if that is crucial (done with client-side javascript).

But it is the front end only, that is, it is usually not connected to any back end code or database as it doesn't need to be. A prototype is mainly used to show what the product may look like or even how it may act without actually performing except for the screen flow (therefore, since the flow can be represented by the order of the screens, a prototype can be made on paper). The other just as important purpose of a prototype is to allow for testing to some degree prior to (or in tandem with) any front or back end development work.

## Site/App

The term "site" usually applies to a web site built purposely for the size of screens of desktop and laptop computers. An "app", short for application, can be a site or a way into a site, but is usually either a possibly reduced functionality site for mobile or a tool (or game, etc.) purposely created for mobile screen sizes. Data speed and transfer size are also concerns due to limitations of the devices used.

## SME (Subject Matter Expert)

An expert in the subject(s) of the product content. There may be a number of subjects and therefore a number of SMEs.

## Storyboard

A term borrowed from the animation, movie and writing industries to mean pictures that show the “story” or flow of the site/app. Storyboards can be made on any media such as paper or white board or created using specialized software. See also prototype.

## Subordinate Window

A window that is an overlay, that is, that doesn’t completely cover the calling page or window. Subordinate windows typically appear subordinate by the simple fact that the user can see part of the calling page/window behind it. Subwindows are often called overlays or dialogs.

## UCD – User Centered Design

A design methodology that considers the user in every step of product creation.

## UE, UX

User Experience

## UI, UID

User Interface/Interaction, Design/Designer

## UXA

User Experience Architect. The UXA designs the experience mostly through navigation and flows; also creates wireframes.

# Wireframe

A representation of a page, subwindow or page part, to any degree of fidelity, active (clickable/tappable) or inactive (static, paper, whiteboard, etc.). Wireframes have multiple uses:

1. As a tool to build a product prototype for testing and review
2. As a sketching tool to try ideas for pages or sites/apps
3. As quick visual ideas of pages to model or verify interactions
4. As a representation of the clickthroughs of a site/app
5. As a representation of the look of a page or site/app
6. Any combination of the above

See also prototype, storyboard.

# Bibliography

Many of these references were used for this book and many used indirectly for research and personal education.

Branaghan, Russell J. (editor). "Essays on Usability". Usability Professionals Association (UPA), Chicago, IL, 2001. ISBN 0-9702272-0-5. [www.UPAssoc.org](http://www.UPAssoc.org)

Cobley, Paul and Jansz, Litza. "Introducing Semiotics". Totem Books, 2005 (through Icon Books Ltd., Thriplow, Royston SG8 7RG, UK, distributed in USA by National Book Network Inc., Maryland). ISBN 1-84046-584-0.

Conway, Susan and Sligar, Char. "Unlocking Knowledge Assets". Microsoft Press, Redmond, WA, 2002. ISBN 0-7356-1463-6.

Cooper, Alan, Reimann, Robert and Cronin, David. "About Face 3: The Essentials of Interaction Design". Wiley, New York, 2007. ISBN: 978-0-470-08411-3.

Dumas & Redish. "A Practical Guide to Usability Testing". Intellect Books, Portland, OR, 1999. ISBN 1-84150-020-8

Garrett, Jesse James. "The Elements of User Experience". American Institute of Graphic Arts, New York, 2003. ISBN 0-7357-1202-6.

Gordon, Ian E. "Theories of Visual Perception". Psychology Press, East Sussex, BN3 2FA, and New York, 2004. ISBN 1-84169-383-9.

Hackos, JoAnn T. & Redish, Janice C. "User and Task Analysis for Interface Design". Wiley, New York, 1998. ISBN 0-471-17831-4.

Harvard Business School Press. "Hiring Smart", HBSP, Boston, 2006. ISBN 1-59139-972-6. [www.HBSPress.org](http://www.HBSPress.org)

<http://www.acm.org/interactions>.

Interactions, a magazine included with membership in the ACM SIGCHI (Association for Computing Machinery, Special Interest Group of Computer-Human Interaction); ACM, New York. ISSN 1072-5220.

Krause, Jim. "Color Index". HOW Design Books, an imprint of F&W Publications, Inc., Cincinnati, OH, 2002. ISBN 1-58180-236-6.

Krug, Steve. "Don't Make Me Think". New Riders, Indianapolis, IN, 2000. ISBN 0-7897-2310-7

Marcus, Aaron. "Graphic Design for Electronic Documents and User Interfaces". ACM Press, New York, 1991

Nielsen, Jakob. "Designing Web Usability: The Practice of Simplicity". New Riders, Indianapolis, IN, 2000. ISBN: 1-56205-810-X. Also his web site: <http://www.useit.com>.

Norman, Donald A. "The Design of Everyday Things". Doubleday, New York, 1990. ISBN 0-385-26774-6.

## ■ BIBLIOGRAPHY

Papanek, Victor. "Design for the Real World". Chicago Review Press, Chicago, 1999.  
Schneiderman, B. "Designing the User Interface: Strategies for effective human-computer interaction". Addison-Wesley, Boston, 1992.

Sion, Avi. "Phenomenology: Basing Knowledge on Appearance". Lulu, Raleigh, NC, 2009. ISBN 978-2-9700091-5-3.

Tufte, Edward. "Envisioning Information". First edition 3rd printing with revisions, 126 pages. Graphics Press, Cheshire, CT, 1992.

Winograd, Terry. "Bringing Design to Software". ACM Press/Addison-Wesley, Boston, 1996. ISBN 0-201-85491-0.

Worldwide Web Consortium. See <http://www.w3.org/UI> for information on UI concepts, dealing mostly with tools such as XHTML, XSL, and CSS.

### **Some helpful links...**

[www.useit.com/](http://www.useit.com/)

Jakob Nielsen's site on usability and web design

[www.upassoc.org/](http://www.upassoc.org/)

The Usability Professionals' Association (UPA) web site

[www.usability.gov](http://www.usability.gov)

The "...primary government source for information on usability and user-centered design"

[www.aber.ac.uk/media/Documents/S4B/sem01.html](http://www.aber.ac.uk/media/Documents/S4B/sem01.html)

Introduction to semiotics

[www.usabilityfirst.com/glossary/term\\_1197.txt](http://www.usabilityfirst.com/glossary/term_1197.txt)

Usability Glossary from Usability First - some ISO documents on usability

[www.catalyze.org/](http://www.catalyze.org/)

Site of Catalyze, a "community for all professionals involved in defining business systems, designing software applications and creating websites," including forums, resources, blogs, and more.

Linkedin.com (UX groups)

There are various UX groups that discuss some good issues, from what UX is to what its various jobs may pay.

# Index

## ■ A

Affordance, 67, 93  
Agencies, 1  
Audience spectrum, 75

## ■ B

Best practices, 5  
Breadcrumbs, 93

## ■ C

Carousels, 41  
Chromostereopsis, 19  
Cognition, 93  
Collaboration, 32  
Color psychology, 8  
Content, 43  
Copernicus, 26  
Cross-sell/upsell, 45  
CSS, 100  
Customer experience  
(CX), 29–30  
CXO, 2

## ■ D

Dark patterns, 8  
Design concerns, UX  
  above the fold, 52  
  Ad hoc navigation, 51  
  autoflow, 51  
  average person, 48  
  clever *vs.* usable, 39–41  
  client options, 48  
  context, 46  
  customers *vs.* business, 44–45

  delighting users, 49  
  design flow, 50  
  edge cases, 45  
  first impressions, 42–43  
  flow buttons, 51  
  forms design, 54  
  hamburger menu, 52  
  information architecture, 44  
  OTB, 46  
  overconveniencing, 52  
  policing, 46  
  proximity, 54  
  science of options, 47–48  
  SEO, 43–44  
  simplicity, 53  
  stepwise flows, 50–51  
  trends, 41–42  
Dunbar's number, 20  
Dunning-Kruger effect, 20–21

## ■ E

Ecommerce, 55–56  
Emails, 15  
Empathy, 18  
Experiential cognition, 13  
Eye tracking, 75

## ■ F

Field labels, 59  
Frustration, 67

## ■ G

Gallileo, 26  
Gestalt, 22  
Graphics, 7



## ■ H

Hamburger, [14](#), [39](#), [52](#)  
Horizontal scrolling, [64](#)  
Human-anything interface (HAI), [2](#)  
Human-computer interface (HCI), [2](#), [29](#)

## ■ I

Industry influence, [14](#)  
Information Architect (IA), [94](#)  
Instructional Design (ID), [94](#)  
Instructional Systems Design (ISD), [94](#)  
IT departments, [2](#)

## ■ J, K

Jargon, [58](#)  
Javascript, [80](#)

## ■ L

Labels, [57–58](#)  
    button labels, [59–60](#)  
    column headers, [58–59](#)  
    copywriting, [61](#)  
        De facto UI standards, [63](#)  
        futzers, [63](#)  
        initial sketching, [63](#)  
        internal and external consistency, [61](#)  
        wall UI displays, [62](#)  
    date/time, [65](#)  
    discovery, [69–70](#)  
    double-duty affordances, [70](#)  
    frustration, [58](#)  
    hidden affordances, [69–70](#)  
    page titles, [59](#)  
    parallax, [70](#)  
    progressive disclosure, [66–67](#)  
    progress visibility, [66](#)  
    scrolling and columnization, [64–65](#)  
    search tools, [67–69](#)  
    skeuomorph, [71](#)  
Luminance contrast, [22](#)

## ■ M, N

Metaphor, [94](#)  
Minimum Viable Product (MVP), [31](#)

## ■ O

Outside-the-box (OTB), [46](#)

## ■ P, Q

Pencil and paper, [63](#)  
Perceived authority, [95](#)  
Phenomenology, [22–23](#)  
Popup, [64–65](#)  
Principle of Least Astonishment (PLA), [95](#)  
Progressive complexity, [95](#)  
Prototype, [95](#)

## ■ R

Reflective cognition, [13](#)  
Research technique(s)  
    audience spectrum, [75](#)  
    concision, [74](#)  
    eye tracking, [75](#)  
    planning, [74–75](#)  
    timing, [74](#)

## ■ S

Scotoma, [23](#)  
Search Engine Optimization (SEO), [43–44](#)  
Semiotics, [12](#), [99](#)  
Shapes, [8](#)  
Society, [5](#)  
Software development life cycle (SDLC), [11](#)  
Statistics, [86](#)  
Storyboard, [62](#), [96](#)  
Subject matter expert (SME), [96](#)

## ■ T

Testing problems  
    A/B testing, [77](#)  
    “Add to Cart” button, [76](#)  
    test creation, [77](#)  
    test results analysis, [77](#)  
    tests assume environment, [75](#)  
    usability testing lab environment, [78](#)  
    UX testing, [75](#), [78](#)  
Type-ahead search, [68](#)

■ U

- UI design, 80
- User centered design (UCD), 96
- User communication, 14
- User experience (UX) design, 5, 96
  - dark patterns, 91
  - diplomacy, 25
  - discovery, 83, 85
  - false consensus bias, 83
  - groups, 86
  - hands-free, 86
  - honesty, 91
  - languages, 80
  - naïve/gullible users, 84
  - online discussions, 89
  - physical interaction
    - mechanism, 85
  - planning
    - market, 37
    - product, 35
    - responsive design, 36–37
  - statistics, 86
  - tact, 25
  - types of manipulation, 91
  - usability, 85
- User welfare, 14
- Users' emotions, 9
- UX psychology
  - basic drives, 12
  - best practices, 33
  - chromostereopsis, 19
  - cognition, 12, 13
  - collaboration, 28
  - communication, 15–17

- curse of knowledge, 19
- vs.* CX, 29–30
- dark patterns, 11
- decoy effect, 19
- definitions and thresholds, 13–14
- depressive realism, 20
- Dunbar's number, 20
- Dunning-Kruger effect, 20–21
- eye tracking, 21
- full stack unicorn, 27–28
- gestalt, 22
- industry influence, 14
- levels, 13, 26
- luminance contrast, 22
- MVP, 31
- negative bias, 15
- phenomenology, 22–23
- product management, 32
- requirements, 11
- scotoma, 23
- specialization, 28
- statistics, 30
- subservient, 31
- unlearning, 23
- users, 14

■ V

- Vertical scrolling, 65
- Visual groupings, 54

■ W, X, Y, Z

- Welfare (user), 14
- Wireframe, 97