CREATING PLAYFUL, FUN, AND EFFECTIVE USER EXPERIENCES

Seductive

INTERACTION
DESIGN

Stephen P. Anderson

Why Seductive Interactions?

THE SETTING IS the Odenplan Metro Station in Stockholm, but we could easily be in the subway in New York or the Tube in London. There's an escalator running next to a set of stairs, both leading up into the daylight. Only today something is different. An experiment is taking place. The stairs seemingly have been transformed into piano keys. Not only do they resemble the familiar black-and-white keys of a piano, but through the magic of technology—sensors attached to speakers—every step triggers the resonant sound of a piano key.

We watch as two people start to take the escalator, pause, then place a foot on the first step. A low, loud note rings through the station. They each take a few more steps, caution

transforming into more of a leap in their step. The air is filled with the sound of a musical scale. We see more people approach, just curious at first, but soon they are delighted. For some people the goal is no longer to even exit the station—they step back and forth across the keys to create a melody.

What's going on here?

This is an experiment in behavior change: "Can we get more people to choose the stairs by making it fun to do?"

As the video continues, we see that most people choose to abandon the escalator in favor of the makeshift piano key stairs. In fact, on the day this experiment was run, "66 percent more people than normal took the stairs."

This little experiment in human behavior is a good distillation of the kinds of things we'll discuss in this book: ways to influence behavior through fun, playful activities. More specifically, we'll learn how the same tactics we use to attract a mate can apply to interactions between humans and interactive devices.

As our focus is on human behavior, in all kinds of contexts, both physical and digital, let's couple the piano stairs with a second story, this one taking place on a Web site.

LINKEDIN AND PROFILE COMPLETENESS

If you were signed up with the professional networking site LinkedIn prior to 2004, then you may remember when they rolled out their Profile Completeness feature (see above). Like many other people, I had a rather anemic profile. I had set up basic account information and added my current employer, but that was about it. Then I got an e-mail about profile completeness. It turns out that my profile completeness was only about 25 percent. But, by adding another position I could be 40 percent complete. That was a simple enough request. So, I added my last employer. Then I got another prompt: "Adding a summary will bring you to 55 percent." So I did that. And so on. Through a series of prompts, eventually my profile was 100 percent complete. My profile now listed past and present employers, my education, a summary, recommendations, and other details you might find on a resume. LinkedIn had somehow managed to pull quite a bit of information out of me. And not just me, but thousands of other users. What's interesting to note is that this little feature wasn't difficult to develop. It didn't require a skilled information architect. It wasn't even a visual design challenge. It was psychology that made this feature so effective. Understanding just a little bit about what motivates people resulted in more people sharing their information and using the LinkedIn service.

The question I like to ask with examples like these is, "Why does this work?" What are the underlying psychological principles that made this an effective feature? In game design, this is an example of *progress dynamic*. And we see this not only in games, with points and levels, but also in other contexts, such as martial arts. Think of the different colored belts you earn while advancing toward a black belt. What would happen if, on the first day of training, the instructor told a young boy, "Stick with it,

work hard, and maybe in ten years you'll be a black belt like me." This goal would seem unattainable. By having different colored belts (and stripes on each belt), you get rewarded and recognized along the path to mastery. These belts are a tangible, achievable goal to work toward, and once acquired, they signal to you and others: look at how far I've progressed!

But this still doesn't answer the question, "Why does this work?" At least not in a way that we can use in another context. And that's what this book is concerned with: why we do the things we do. In the case of colored belts, points, levels, and progress meters, we could look at several ideas from psychology:

- *Sequencing*. We are more likely to take action when complex tasks are broken down into smaller tasks.
- *Appropriate challenges*. We delight in challenges, especially ones that strike a balance between being overwhelming and being boring.
- *Status*. We constantly assess how interactions enhance or diminish our standing relative to others and our personal best.
- *Achievements*. We are more likely to engage in activities in which meaningful achievements are recognized.

This is the perspective I'd like to offer throughout this book—one that looks for the underlying reasons behind the things we do. If we look at examples like piano stairs or percentage completeness, what principles can we take away from them? And how can we apply these principles to our own projects? More specifically, how can we use these principles to help

people to fall in love with our Web sites, applications, and services? This leads us into the idea of seduction.

WHY SEDUCTION?

With a title like *Seductive Interaction Design*, we'd better start by clarifying what this book is (and is not!) about.

Seduction is defined as:

"the process of deliberately enticing a person to engage in some sort of behavior, ~~frequently sexual in nature.~~"

(For our purposes, we'll ignore that last bit.)

Even at that, "seduction" might seem a rather strong (and odd) choice of words, especially if we're talking about things like increasing conversion on an e-commerce site, or getting people to complete their profiles or write a review for your product. Can these kinds of online interactions really be considered seductive?

One negative meaning of the word seduce is "to lead astray." In that sense, you might think of tricking someone into engaging in behavior they might not otherwise pursue. Think of the Sirens from Greek mythology, who lured sailors to their death with their irresistible song, or the famous eighteenth-century womanizer, Casanova.

But seduction can be seen in another way, as a critical part of attraction. To be seductive is to be tempting or attractive.

In this sense, seduction means "to be led along." In nature, seduction is seen when the peacock shows off his plumage to attract a mate, or when humans flirt and play to get to know

each other more intimately. Courtroom lawyers are skilled at seducing jurors through carefully chosen words. Great speakers know how to seduce an audience. Even well designed experiences such as Disney World are essentially seductive interactions. And ask any parent: children have to be seduced into doing the right things. We may use other words, like motivate, persuade, or inspire. But in these cases, there is nothing insidious about seduction. It is a necessary and critical game we all play in most areas of our lives. (Whether we're any good at it or not, that's another matter.)

Why should the things we're building be seductive?

See if this sounds familiar:

You've got a great site (or Web application). People seem to like it. It does fine in usability testing. Maybe you have some outstanding feature requests, but nothing big. In fact, there's nothing particularly wrong with what you've built. But, maybe you've observed some of these problems:

- High bounce rate: people come, but never come back.
- Low adoption: people just aren't using the service. Maybe it's an enterprise app that only 10 percent of employees are using. Maybe it's a startup with few active users.
- Too few registered users: your measure of success is sign-ups or registered users, but people aren't even doing this.
- No differentiation from the competition: you're in a crowded space and the differences between your product and others are too subtle.

- Very few referrals: people just aren't interested in telling their friends about you.
- No clickthroughs on e-mail campaigns.

We could go on, but you get the idea. If your app could speak for itself, it might say: "I'm a great app, if people would just get to know me."

A few years ago, I began observing some of these same problems on two very different projects. One was a media focused software application for a very large electronics retailer. The other was a small technology start-up in the search engine space. In both cases we found that when we walked people through a demo, they generally liked what they saw. But, these applications did a terrible job of selling themselves. Here were two very good applications that people liked and might use—once they got to know them. But that's the problem—outside of a usability lab or the urging of a good friend—few people would take the 5, 10, or 15 minutes needed to get to know these applications. And this is where I began thinking about seductive interactions.
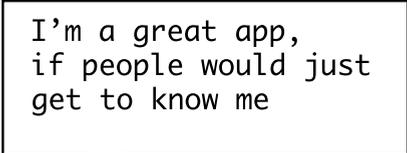
I'm a great app, if people would just get to know me
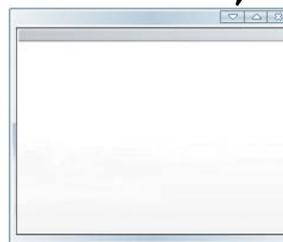
Think of our products as that geeky friend, you know the one—they are the nicest person in the world, or have really great ideas about things. But, they have poor social skills. Maybe they have a difficult time striking up a conversation. Or they need substantial conversational support from friends. Teaching them to be seductive may call for some simple confidence-building exercises. Or maybe a few conversation starters and icebreakers would help. They could also learn to listen and ask interesting questions.

This geeky friend is most of the software and sites we have to interact with. They may be interesting, but we're not going to stick around to find out. Even in corporate environments, industries will spend millions of dollars to roll out some new platform, only to be shocked that no one is using it! Now imagine if that software is an online service. There are no sunk costs to compel you to use the application. Instead, there are billions of Web pages. Free trial accounts. Low monthly subscriptions with no commitments. And there's a new technology start-up launched every day. That's a lot of competition. How can you possibly stand out? We need to learn a bit about seduction—*why* were people seduced by stairs disguised as piano stairs and profile completeness progress bars? We need to learn how to "deliberately entice a person to engage in some sort of behavior."

THE iLIKE STORY

I mentioned that one of the projects I worked on was a media-focused software application. As part of the design process, I looked at dozens of music and media sites. Many of the sites I looked at were start-ups. And consistent with the problem I described earlier, my first experience with most of these services was so brief or unremarkable that I never came back. With some of these sites I created an account, but they did such a poor job of introducing themselves and letting me know why I should care, that I doubt I'll ever go back. I may even tell others not to bother, possibly for completely wrong reasons.
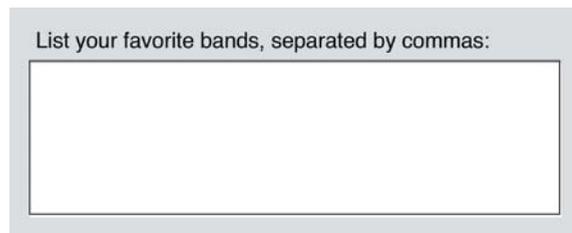
In contrast, the experience at iLike stood out as positive (as did BLIP.fm). Let's look specifically the iLike *registration* process.

The first few pages of the iLike sign-up process were nice, but unremarkable. They didn't ask for too much information. And they were very clear about why they were asking for the things they did. It's obvious why my e-mail address and a password would be required. But my zip code? They offered very clear (and brief) help text explaining that giving them my zip code would let them notify me whenever my favorite bands were in town. They emphasized that providing this information might benefit *me*. I'm sure they have plenty of business reasons for wanting my personal data—demographics, targeted marketing promotions, and so on—none of which I care about particularly. But their reasons were stated in terms of how it would benefit *me* to provide this information. Think about our dating analogy. At least as far back as Dale Carnegie's 1930s book *How to Win Friends and Influence People*, we've known that we are more interested in people who are interested in us. No one wants to sit and hear someone talk about themselves all night. The same is true in many online interactions.

There were a few other conveniences during the registration process. iLike made it easy to invite more friends by sharing my e-mail information. They also offered the iLike sidebar, an iTunes plug-in that monitors what I'm listening to in order to make better recommendations.

While convenient, none of this was particularly remarkable. It's what came next that got my attention.

In almost every other sign-up process for a music site—or any site that wants to find out your personal favorites—there's a page where you're asked to list your favorite bands. I call it the big, empty text box. It looks something like this:

List your favorite bands, separated by commas:

And we dutifully list a handful of favorite bands (or movies, or hobbies, or what have you).

iLike, however, never gave me the chance to *list* my favorite bands. Instead, they showed me a page with 35 artists and simply asked me to "click on your favorite artists." The Beatles. Radiohead. Coldplay. This was fun. I got to the bottom of the page and was presented with an option: I could be done with the registration process or click for more artists. Of course I clicked for more artists! In fact, I continued clicking on more of my favorite bands for nine pages, until "Click for more artists" wasn't an option!

At this point I like to pause and pull out "the money slide." Successful businesses figure out how to join business goals with user goals. We call this *value-centered design*.

In the case of the iLike registration process, I had a great time clicking on bands I like. In turn, iLike gained lots of data about my musical tastes and preferences.

In fact, clicking on bands and artists was a lot more fun than filling out an empty text box. And look what iLike got out of the experience: I shared 35 of my favorite bands with them. Compared to other music sites where, at best, I'd list maybe four or five bands, iLike learned a whole lot more about what I like to listen to. This design of the experience was mutually beneficial.

So why did this work?

- **Feedback loops.** We're engaged by situations in which we see our actions modify subsequent results. iLike made a very small suggestion: "The more artists you rate, the better." They were careful not to spell out when I'd see results; it could have been on the very next page that reloaded or three months down the road after I was an active user. That merely suggested that my actions would tailor my iLike experience.
- **Curiosity.** When teased with a small bit of interesting information, people want to know more. I was curious: how would clicking on the artists I like affect results on the next page? Would I see less of one genre, and more of another? Would the suggestions become more personalized with each new batch of 35 artists?

- **Pattern recognition.** Our brains seek ways to organize and simplify complex information, even where there is no pattern. I was looking to see if there was more or less of one type of artist. Also, why these 35 artists? Was there a pattern to the artists I was rating?
- **Visual imagery.** Vision trumps all other senses and is the most direct way to perception. If you think about this page from a technical perspective, it's nothing more than a list of 35 checkbox items. However, by using photographs of the artists, there was a more immediate, visceral reaction. And I had a larger click target.
- **Recognition over recall.** It's easier to recognize things we have previously experienced than it is to recall them from memory. Instead of having to recall bands I liked from memory, this was an easier, passive experience. All I had to do was click on artists I recognized and like. No mental strength was required. More importantly, my hand never left the mouse—it's much easier to click from available options than to type things out by hand.

All in all, this was a remarkable sign-up process. But, my initial iLike experience wasn't over yet.

The iLike Challenge

About 15 minutes later, I was checking my e-mail. There was a confirmation e-mail to let me know that I had successfully created a profile on iLike. The e-mail suggested what I might do next. I could add a photo to my profile. Not now, thanks. There was another plug for their iTunes plug-in. Meh. Or—and this is the one that piqued my interest—I could "Play the iLike Challenge." I was curious. What was the iLike Challenge?

It turns out that the iLike Challenge is this really addictive game that should be avoided at all costs.

The iLike Challenge presents you with a 30-second sample from a song. The challenge is to identify either the artist who wrote the song or the title of a song. You get points for answering accurately and quickly. For example, the challenge may be "Name this Gorillaz song," and you have four options to choose from. If you answer in the first few seconds, you get ten points. If it takes five or six seconds to answer, you get nine points. The longer you take to answer, the fewer points you get. If it takes you 28 seconds to figure out what the song is, you're only going to get one point! And if you answer incorrectly, you get no points.

This game is fairly addictive in and of itself. It tests your knowledge of popular songs. And, with multiple-choice answers to choose from, you have some chance of guessing correctly.

Here's the part that's particularly evil: visible on the side of the screen is a scoreboard that keeps track of how you're doing.

- Your current rank
- Total points accumulated
- Points to next rank (Remember progress dynamics? Here I only need 48 more points to reach the next level. Hmm. I wonder what's after Music Intern?)
- Questions answered

- Percentage of correct answers
- Average answer time

These are all fun metrics to monitor. But, here's the one that really worked on me: *Best streak*. Best streak is the total number of points based on consecutively correct answers. Even if I did start to tire of the music game, there was another game introduced here, one in which I compete against my own personal best. If my best streak was 47, I had to beat that. And when I did, I'd set goals: can I get above 60? Above 100? All said, I spent over an hour playing this game. And then I shared it with friends and family. That's a *seductive* interaction.

Here's the other interesting thing about the iLike Challenge. As you're playing it, there's an area below the current challenge that gives you all sorts of information about the last song that played. It's very easy, while playing the game, to move your cursor and click "iLike" that song. Think of how nice it'd be if our car radios had a Like button for the songs we enjoy hearing. It'd be an easy way to identify and recall songs we

like, especially in cases where we don't know the artist or song title. iLike was providing yet another way for us to share what music we like.

Let's put our business hats on again.

I had a great time playing this game. But what do you think iLike was doing? What are they doing behind the scenes, perhaps? Here's a chance to capture not only more artists and songs I like, but with the right systems in place, the iLike Challenge could be a pretty intelligent system for collecting preferences. If I only get 5 percent of the R&B questions correct, but I answer accurately on just about every indie song they throw at me, that information reveals a lot more about what I listen to. By monitoring my knowledge of different songs, they could learn even more about my musical tastes and preferences. That's clever.

Why did this work? It would be easy to focus on the gaming mechanics: points, levels, a countdown timer. But why do game mechanics work? I devote several chapters of this book to answering that question. Recall four of the ideas you've already seen: *status*, *feedback loops*, *achievements*, and *appropriate challenges*. There is also the *sensory appeal* factor. We're engaged by and more likely to recall things that appeal to multiple senses. Given the constraints of a digital context, it's hard to engage multiple senses, but with the iLike Challenge, you have the artist's visuals, the motion of a timer counting down, the audio input, and the interaction required by a game.

BEYOND USABILITY

At this point, I'd like to point out something:

It wasn't a focus on usability that made this a great experience. It was psychology.

Usability clears the way for a good experience by eliminating troublesome interface distractions, but a great experience stems from something more—an awareness of why people could or do care. The danger is in confusing "ease of use" with actually *desiring* to use something. These are two entirely different things. Both are essential, but simply making something more usable won't guarantee any more clicks or conversions. In this case, it was psychology that made this so engaging. To be clear, if a business approached me tomorrow about making their time-tracking tool fun to use, I'd start by making sure it was first easy to use. Adding "playful" elements on top of a frustrating experience will only complicate things. Fix the basic problems before moving on to the types of things described in this book. However, I've seen cases where the motivation to do something outweighed the usability challenges—the

(Joshua Porter)
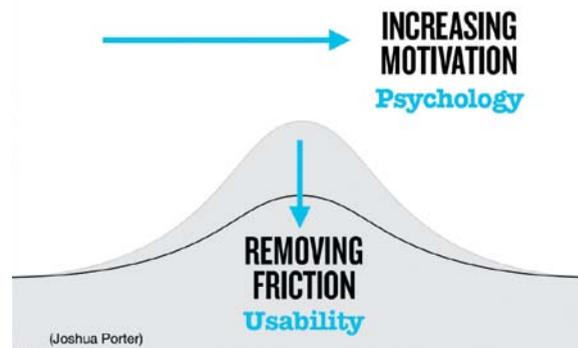
pull outweighed the friction. In one such case, there was a very long form that asked for lots of unnecessary information; but, if you managed to get through the application, your group got listed on a heavily trafficked site. Understanding motivation can be a powerful design tool.

Usability is about removing the roadblocks and obstacles that get in the way of a great experience—there's nearly two decades of excellent literature on this subject; everything I discuss in this book is about increasing motivation.

With this in mind, I'd like to introduce you to a model I developed to help discuss different user experience needs.

IT'S ALL ABOUT EXPERIENCES

Author and professor Donald Norman once stated, "When technology delivers basic needs, user experience dominates." The mobile phone industry is a great example of this evolution. No one would now consider entering that market with the original Motorola DynaTAC. What made it successful at the time—a phone that doesn't need wires—is now an expectation. Mobile phones have become more reliable and usable and are packed with some useful (and not so useful) features. The introduction of the iPhone and its imitators raised the bar once again—not in terms of more features, but in the way in which people experience information. Now our mobile phones aren't just tools—they're also fun to use! This process of product maturity forms the basis for my *User Experience Hierarchy of Needs* model, in which I propose that most technology product and service experiences go through six levels of maturity,

moving from the bottom to the top, from "Hey, this thing actually works!" to "This is meaningful in my life." However, this model was also my way of resolving a lot of different ideas around what is important to an experience, and the relative priority of those things. This latter idea is seen with a top to bottom focus.

Moving from bottom to top, you have a basic product maturity continuum:
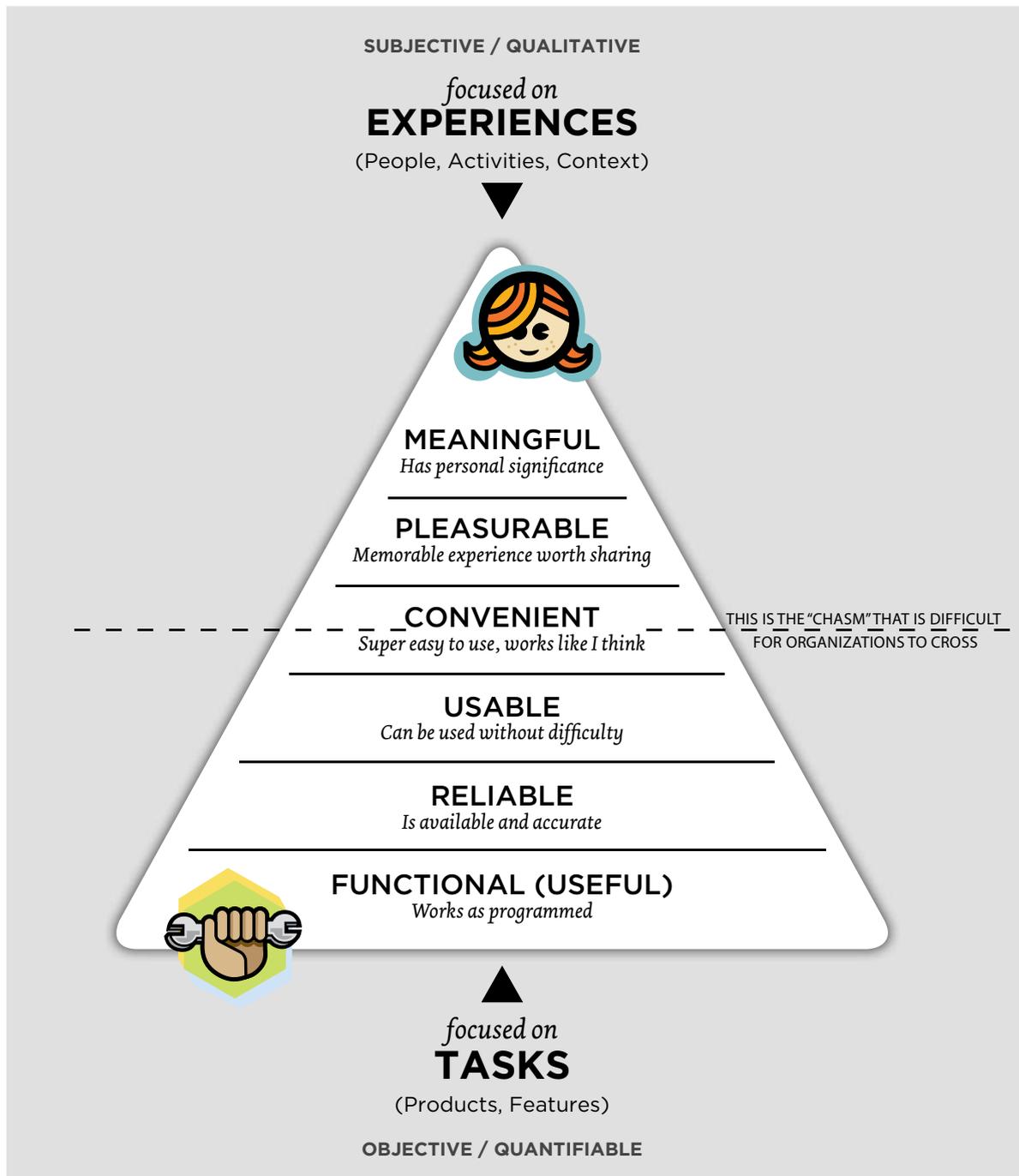
Functional

Ideas typically start off as *functional* solutions to a problem—something *useful*. Think of the first Motorola cell phone. Sure, it was a brick, but it allowed you to make calls untethered to a fixed spot! Every new technological innovation starts at a functional level.

Reliable

From there, things have to be *reliable*. This can be reliability of the service (five nines uptime?) as well as integrity of the data. If I purchase tickets on a travel site, the ticket prices need to be current and reliable. If I host with a site, I need to know my data is backed up and accessible at all times. This is reliability. When sites fail at reliability, especially where personal data is involved, little else matters.

Usable and Convenient

It's not enough to allow me to simply do something—it has to eventually be less awkward to use. This is where the next two levels, *usable* and *convenient*, come into play. I make a distinction between usability and convenience. Both make something easier to use, but in my experience most usability groups focus

focused on
EXPERIENCES
(People, Activities, Context)

▼

MEANINGFUL
Has personal significance

PLEASURABLE
Memorable experience worth sharing

CONVENIENT
Super easy to use, works like I think

THIS IS THE "CHASM" THAT IS DIFFICULT
FOR ORGANIZATIONS TO CROSS

USABLE
Can be used without difficulty

RELIABLE
Is available and accurate

FUNCTIONAL (USEFUL)
Works as programmed

▲

focused on
TASKS
(Products, Features)

OBJECTIVE / QUANTIFIABLE

User Experience Hierarchy of Needs model. From bottom to top is a basic product maturity continuum: a top to bottom focus starts with the experience you want people to have.

on fixing known problems—removing the hurdles. A focus on convenience asks, "Is there a more natural way to make this work?" MapQuest and Google Maps are great examples of this contrast. MapQuest was perfectly usable. But Google Maps, with its draggable interface, physics, and other more natural behaviors was a much more convenient way to interact with maps data. Touchscreen interactions, such as those offered by the iPad, are a perfect example of a more convenient interaction. Things work more like they might in the real world.

Pleasurable

Whereas convenience focuses on cognition, the next level—*pleasurable*—focuses on affect and emotions. How can we make something emotionally engaging (and memorable)? This is typically accomplished using things like friendly language, aesthetics, and humor, and doing things like arousing curiosity, creating flow, leveraging game mechanics, and other similar tactics. Everything described in this book is about creating more pleasurable experiences.

Meaningful

The highest level is, of course, "meaningful." To be clear, you can't make something meaningful for someone—*meaning* is personal and subjective. But you can design for meaning by focusing on the preceding levels as well as shepherding beliefs and the communities surrounding the product or service experience. Great companies know how to develop a story that people can believe. Disney World is about more than rides. Apple is about more than electronics. Whole Foods is more than a grocer. These brand stories transcend any particular product or service.

Also, whereas the other levels build on each other, a product can be meaningful without any of these levels. For example, I have a 1966 Karmann Ghia that doesn't even run—it doesn't operate at even a functional level; however, the ownership connects me with a group of people in a way that is personally meaningful.

The challenge of this model is this: if you want to truly create a revolutionary product, you have to shift your thinking from a bottom-up *task* focus (which will only get you so far) to a top-down focus that starts with the *experience* you want people to have. By approaching things from this perspective, we see a host of new ideas, not to mention better ways to implement ideas that have been around for a while.

But there's another takeaway: in mature markets, where you have stable, usable products, taking it to the next level means focusing on more experiential things like emotions, clever language, and aesthetics. I present this model as a context for everything in the chapters that follow. The tools and products we build—Web apps, software, mobile apps—have reached a point where we can engage people in meaningful and emotional ways. But let's not forget the basics!