UX101

A Primer on User Experience Design

Joseph Dickerson

UX 101: A Primer on User Experience Design

Joseph Dickerson

Second edition published May 2014

ISBN-13: 978-1492277248

ISBN-10: 149227724X

Dedication

For aspiring user experience designers everywhere, I hope this book helps “kickstart” you into your future career.

For my wife and sons, for not interrupting me (too much) while I wrote this.

And I’d like to express my gratitude to the enemies of crime and crusaders against crime throughout the world for their inspirational example. To them, and to lovers of adventure, lovers of pure escapism, lovers of unadulterated entertainment, lovers of the ridiculous and the bizarre... To fun lovers everywhere... This book is respectfully dedicated.

Table of Contents

Introduction

If you walk down the street of any city in the world and ask people what User Experience Design is, the majority of them would say, "I have no idea." Yet when you ask them what mobile phone they use, the store they love to shop at, or the place they love to travel to... you'll find they will most often name products, stores, and destinations that were the result of user experience design.

Apple. The Gap. Target. Disney. Procter and Gamble. Some of the biggest companies in the world are not in the product or service industry... they're in the Experience business, applying User Experience Design process and principles to what they sell to their customers. And they are very successful because of it.

So, what is user experience design? User Experience, or “UX” Design is a discipline focused on understanding users needs, wants and desires and applying that understanding in product and service design. It uses research, conceptual design, and user testing to identify what works, to reject what doesn't, and to help companies focus on producing usable and useful offerings to bring to market. Applying UX design processes can, and has, helped companies save hundreds of thousands of dollars, by making sure what they are producing services real needs, and not trying to solve “imaginary” problems.

As a user experience architect at a Fortune 500 company, I've worked hard to ensure my company’s offerings are designed with the user in mind, and because of my and my colleague’s efforts the many products we provide have been acclaimed by users and helped my company win contracts... and the hearts of customers.

User Experience Design is still a young profession, but as companies around the world has seen the benefits to applying user-centered design principles to their processes and products, demand for UX professionals has exploded. There are over 9000 job

openings for UX professionals on LinkedIn as of the time I write this, and that's during an economy still struggling to recover.

With such high demand, where are the UX design professionals in the future going to come from? There are formal courses of study out there, and lots of books on the topic... but not a lot of materials are aimed at the novice interested in the domain. I hope to help solve (a little bit) of that problem with UX 101.

My Background

I'm lucky, in that I've done a LOT of different things - and I think that variety of experiences has helped me become better at... well, user experience.

It all started in a 5,000-watt radio station in Birmingham, Alabama. With just a $50-a-week paycheck and a dream... No, seriously, Ted Baxter references aside, one of my first real jobs was in radio. I did college radio and, for a while, overnights for the local rock station. Doing a job like that taught me the importance of organization and multitasking, and, most importantly... communication. If you didn't fill that dead air, it was exactly that... dead air. A lot of my communications skills can be attributed to me being behind a "hot mike" and trying to both communicate and entertain.

I also had a passion for design, and at the time the notion of the domain that eventually became known as "user experience" didn't exist, so I learned graphic design and worked doing magazine layouts and advertisements for sporting periodicals such as Bowhunter and Deerhunter Magazine. It wasn't glamorous, but I learned how to flow text around images and present things in a creative way.

For a while I managed a record store for a Kentucky-based chain, and that helped me get a sense of the practical realities of a segment of the business world that allowed me to communicate to business people with a better understanding of where they were coming from.

I then learned about this crazy new thing called "The Internet" and taught myself how to code HTML and make web pages, a skill that came in handy when I started working for a web service company that was a huge opportunity for me to learn about EVERYTHING: database design, server configuration, uptime, DNS tables, web security, e-commerce... the works.

You may notice that I have not yet described what my formal education was... and there's a reason for that. I learned more by working and doing than I ever did in college, though I attribute what I learned when I was majoring in Journalism with a lot of my skills in usability testing, listening, writing, and being able to be objective.

So when UX finally became a domain of note, I was ready... it was the perfect fit for what I had learned and done - it was about something I was always good at - designing solutions for users, and solving problems.

What this book is about (and how to use it)

My previous book on user experience Experience Matters was a collection of essays on design, as well as some practical advice on how to do key activities in a user-centered design process. This book is more structured and formalized, detailing out specific processes and approaches as well as providing overviews of aspects of the user experience world and how the discipline is applied in organizations large and small.

It is, as the title indicates, UX 101... an introductory primer on user experience. It is not intended to be the “one source” to rule them all - one book can’t detail everything that aspiring UX professionals need to know to work in the field. This book is meant to be a helpful stepping-off point to get you acclimated to the domain and user-centered design process. I list a lot of great additional resources at the end of this book, and UX 101 is in no way intended to replace or supplant any of them.

If you are an experienced UX practitioner, you may not see a lot here that you don't already know.... but I hope that you find some nuggets of knowledge that can help you in your design process.

NOTE ON THE SECOND EDITION: I have taken the opportunity to correct a couple of mistakes in the second edition and add some more context and graphics to support the reader. Hope it helps.

So, with all that being said, let's get started.

Joseph Dickerson

April 2014

Getting Started

The beginning is always a good place to start, and so... we'll start there. This section details what UX is, a brief history of the domain, some common terms, and some "behind the curtain" secrets that will hopefully demystify things for you.

What is UX, anyway?

So, what is UX? Let's look at the Wikipedia entry, shall we?

User experience (UX or UE) involves a person's emotions about using a particular product, system or service. User experience highlights the experiential, affective, meaningful and valuable aspects of human- computer interaction and product ownership. Additionally, it includes a person's perceptions of the practical aspects such as utility, ease of use and efficiency of the system. User experience is subjective in nature because it is about individual perception and thought with respect to the system. User experience is dynamic as it is constantly modified over time due to changing circumstances and new innovations.

Well, that's a lot of really big words that really doesn't say much. Maybe the ISO definition is clearer...

ISO 9241-210[1] defines user experience as "a person's perceptions and responses that result from the use or anticipated use of a product, system or service."

MUCH better. To rephrase, user experience is shorthand for how people think and feel when they use something... so a product or service could have a bad user experience or a good user experience. Or something in-between.

But, is that all there is? Can it be that simple? In a nutshell, yes... but there are a lot of UX professionals who have defined, reframed and described UX in different ways.

Two quick examples:

Pete Morville, the great designer and head of Semantic Studios, defined a "User Experience Honeycomb" that is more detailed than the above ISO description.

In it, he defines seven different facets of UX:

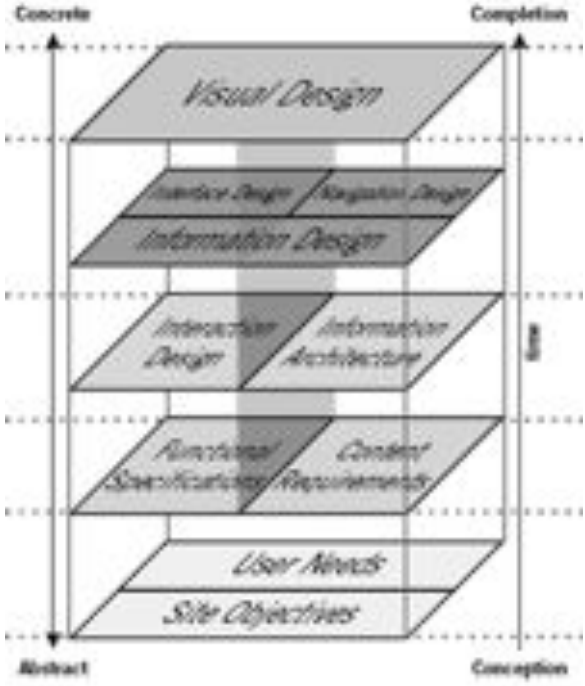
- Useful
- Usable
- Desirable
- Findable
- Accessible
- Credible
- Valuable

He goes into more detail in his article here, and here's a key excerpt:

"It's a great tool for advancing the conversation beyond usability and for helping people understand the need to define priorities. Is it more important for your web site to be desirable or accessible? How about usable or credible? The truth is, it depends on your unique balance of context, content and users, and the required tradeoffs are better made explicitly than unconsciously."

All areas Corville lists in his "Honeycomb" will be covered in more detail later.

Another well-known and popular detailed description of UX came from Jesse James Garrett, in his book The Elements of User Experience. He defined "layers" that go from the Abstract to the Concrete, and among those layers are visual design, information design, interaction design, functional specifications, and user needs.



This visualization is focused on deliverables, process and context... What activities and artifacts are required to produce a quality user experience? This “picture” of UX is useful for those people who need a more- process driven description of UX.

Which one is “right”? Well, they both are, the same way that the many other visualizations people have created to define what UX are as well. There’s no wrong steps and provide examples of the various tactics and processes of UX design. UX crosses many different domains and disciplines, and so it’s hard to “pigeonhole” it into one easy classification.

The various domains of UX

(But what about that other term you heard about, “Service Design, you ask? More about that later...)

Seek out these and different sources to get a deeper sense of the domain, but remember this: There is one key aspect that you should never forget, one that many "process-focused" UX books and articles often do.

People.

The most important thing about user experience design isn't technology, or screen layout, or content... It's people, the users who will engage with the final product you and your team will produce.

People are complex creatures. What motivates person A may not be important to person B. The knowledge that person C brings to a device may be markedly different than person D. And so on.

The title I most recently held in a user-centered design team, "UX Architect", was actually incorrect. I can't "architect" an experience, because that is an individual thing that varies by person. All I can do as a design lead is get a better sense of the users who will be taking advantage of the solution my team is designing, and make sure that solution aligns with their needs as much as possible. That's the hard part.

UX 101 is about user experience, and a typical UX design process. These processes involve design and documentation activities, but throughout all the activities users are a vital part of what takes "answer" as long as these descriptions don't stray from the key ISO description. Heck, this and the rest of the UX 101 book is my own "take" on UX, so I can't criticize anyone else's definition of the disciple (since the book is my own statement on the subject).

And so, now that we know about what user experience is… what is user experience design? The various definitions propagated throughout the Internet and the world tends to agree: UX design is the process of understanding user behavior in order to create the best possible experience for the user. There are various ways to do this, but there are some key steps any UX design process should have. UX 101 will detail those vital places. The correct UX design

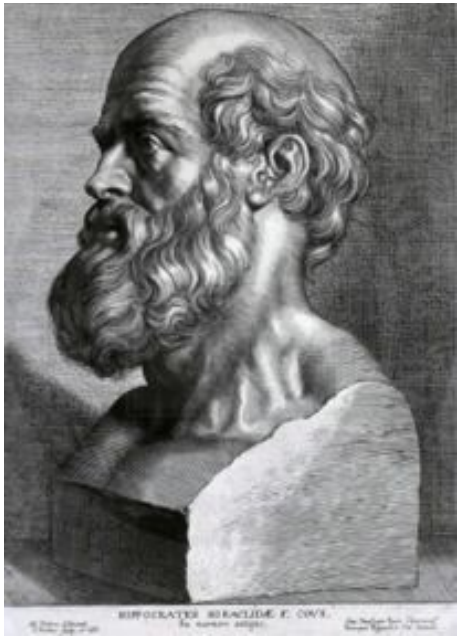
process is “user-centered”, hence the term “user-centered design” (used to describe both teams and the tasks the teams do).

Designing something that works for user involves involving them in the process... it’s a critically important way to ensure what you are creating aligns with how people think and act.

A brief history of the user experience domain

What we now call “user experience design” began a lot longer ago than many people realize. In fact, you can trace UX design WAY, WAY back... all the way to when humans created the first tools. A tool in its most basic form is an object that supports the accomplishment of a task, and tools that are well suited to the job at hand makes our lives better.

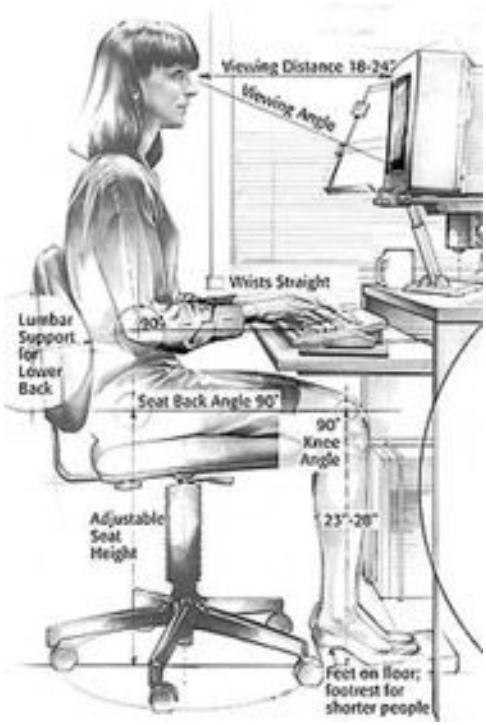
In a way, UX design is about creating that “right tool for the right job”, so in that respect the human race has been “doing” UX design for thousands of years... they just weren’t documenting the design process and principles (most of the time).



I say most of the time because, while the Greeks documented many different design principles in the 5th century, the idea of an active UX process and practice didn't occur until fairly recently.

That "fairly recent" moment took place over sixty years ago, when engineers started looking at "human factors" when designing equipment to fight World War 2. The term "ergonomics" was popularized after the war, when British psychologist Hywel Murrel founded the Ergonomics Society in 1949 to evaluate the lessons learned during the war effort.

Soon, the field of human factors was formalized, and engineers started identifying best practices on how humans interact with physical objects (to inform the design or redesign efforts). As you may have noticed, I went from using the term "human factors" to "ergonomics" and back again – Why? Because the two terms are effectively synonymous (though many practitioners prefer one over the other). As the domain grew, another domain sprung out of it: Industrial design, with practitioners focused on the practical application of ergonomic principles in the creation of hardware and consumer products.



With the computer age many human factors professionals turned their focus towards software design, at the same time many technical and software engineers were setting their own standards around how and creating computer “UIs”... and the user experience discipline, as we now know it, truly began.

Today, User Experience is still a fairly “young” discipline, but over the past decade numerous companies around the world have seen first-hand the benefits of UX design. As a result, demand for UX professionals has exploded. As of the time of this writing, there are over 15000 job openings for UX professionals on LinkedIn alone and that’s during a struggling economy.

If you want to be a UX professional (and I assume you are since you are reading this), the future looks bright.

The “founding fathers” of UX

There’s an old saying that “success has many parents, and failure is an orphan” and that is certainly true when it comes to the user experience field. Many different people helped shape the discipline over the past sixty (plus) years, and here are some of them (in chronological order). Most are published authors and speakers, and I urge you to see out their work to learn more.

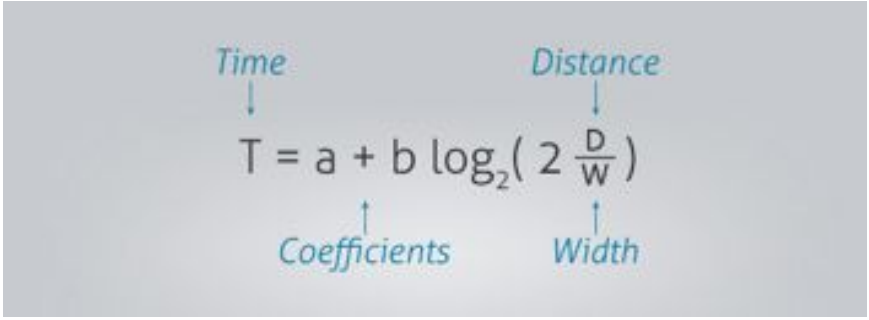
Some names you may know, and others may surprise you...

Dieter Rams

One of the most influential industrial designers of the 20th century, Rams created multiple products for the Braun Company in the 1950s and 1960s. His 10 principles of good design, summarized by the phrase “Less, but better”, are still relevant today (and are good guidelines when designing hardware or software).

Paul Fitts

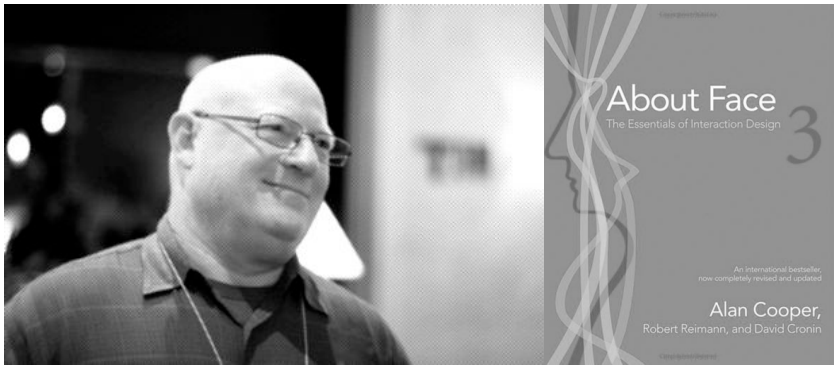
One of the founding fathers of Ergonomics, his famous “Fitts’ Law” (which predicts the time required to rapidly move to a target area, such as a button or control) is still in use today. Fitts was a psychologist who later served in the Air Force, where his work redesigning cockpits did a lot to improve aviation safety.



Fitts' Law. Don't worry, there's no math test later.

Don Norman

His seminal work The Design of Everyday Things, released in 1982, was one of the first popular books on "user experience", and Norman was the man who coined the term. With a background in cognitive psychology, Norman championed the idea that objects and software should be designed and aligned to user's needs... hence, UX or "user-centered design." Norman is still an outspoken supporter of UX at the age of 77.



Alan Cooper

Not only did Alan Cooper create Visual Basic, he was one of the earliest innovators in user experience design. Cooper founded one of the first interactive design agencies (named, appropriately enough, Cooper) and was an early advocate of the use of personas in design. His book About Face: The Essentials of Interaction Design is one of the "seminal" UX books.

Walt Disney

Yes, THE Walt Disney. When he envisioned and built the world's first "immersive experience" – Disneyland – it involved a

tremendous amount of “user- centered design” long before the term was even defined. His pioneering integration of technology and robotics to craft interactive encounters and rides laid the groundwork for much of what we consider “virtual reality” today, and I consider at his “Imagineers” as one of the first UX teams.

Edward Tufte

While he wasn’t the first professional to focus on “information design” (the presentation of complex data in an easy to understand way), he is the most famous figure in the domain. Tufte has written numerous books on the topic (his first was The Visual Display of Quantitative Information in 1982). He has consulted with numerous countries and governments and his books are key sources if you want to know more about creating effective visualizations and infographics.

Jakob Nielsen

The “Father of Usability,” Jakob Nielsen was an engineer at Sun Microsystems when he defined some of the first formal design standards around web usability. He is a pioneer in the user testing and research discipline. He has some critics, who disagree with some of his opinions (most famously, he has stated that visual design in a UI is not that important). He holds a Ph.D. in human–computer interaction from the Technical University of Denmark in Copenhagen.

If you ever see him at a design conference, you will probably see him debating about eye-tracking software with...

Jared Spool

A peer of Nielsen, Spool founded User Interface Engineering and started looking at web design and usability long before the term was popularized. Spool consults with dozens of companies, has authored several books and is an extremely entertaining speaker if you ever get a chance to see him.

Steve Jobs

The co-founder and former CEO of Apple. The late Steve Jobs was not a UX professional per se, but under his direction Apple produced many "user-centered" products the past twenty years, to great success. Apple is case study #1 for UX professionals who need to demonstrate to skeptical stakeholders why user-centered design "just works."

Other notable figures

The following are all "thought leaders" in UX, and most of them are published authors or frequent bloggers (or both). I recommend that you check out their work:

Bill Moggridge, Jesse James Garrett, Peter Morville, Whitney Hess, Kim Goodwin, Steve Krug, Indi Young, Bill Buxton

What makes someone good at UX Design?

I've identified some key skills that I've always found in effective UX designers, and here they are, in no particular order:

Problem solving is one of the key aspects of the user experience discipline. The ability to identify a problem and ideate multiple solutions to that problem is a must-have skill.

A logical mind is another important skill, in that applying logic to a particular problem space or design challenge can help take the emotional bias off the table. Not saying that UX designers should be like Mr. Spock, but a rational focus is something that is pretty crucial.

An appreciation for good design and art is a key input to the design process. If you can't appreciate good design and art, it hampers your ability to such design. And that's in all mediums - film, music, print, hardware, and television.

Empathy is a huge thing, and knowing that you are designing solutions to solve the problems of real people makes the designer focus on making the work more approachable, understandable and usable.

The ability to communicate is also key. If you create a good design and can't explain the details of how it works to the people who have to approve or implement it then that design may never be executed, or it is may become muddled and diluted.

Listening is also key, and with it the ability to take criticism well. Great designers are their own worse critics, but it is not themselves they are criticizing, it is the work... great designers know that they are NOT the work, and so they are open to having the work openly critiqued. By collaborating with peers, listening to their feedback

and responding to it, by testing designs with customers, the good becomes great.

Finally, passion is the "secret sauce" that all great designers have. If you are looking at a UX design job as just something to pay bills and "get by," then you probably will never become a great designer. It's the different between someone who writes classified ad copy and someone who pours their soul into a novel. The ad copy will never be great, but the novel has a much better a chance at being so.

Case Study: The $300 million button

Many years ago, Jared Spool was consulting for a very large e-commerce company. It was early days in the company's history, and (like all companies who sell things) they wanted to increase sales and reduce abandoned shopping carts. They had some very good analytical system and so they knew there were a significant number of customers who left the site without buying anything.

Even with all their data, they had no ideas why so many people were leaving without checking out.

So Jared looked for potential problems. The form that looked to be the source of the problem was very simple: the fields were Email Address and Password, and the buttons were Login and Register. After talking to users and testing the form, he uncovered the problem. The problem wasn't the form - it was where and how it was in the experience.

It was an "interrupt screen" that appeared after they clicked the checkout button, and the attitudes of the users he interviewed were consistent.

The registration form didn't make things easier - users perceived it to be a "blocker." They didn't want to register, they wanted to BUY STUFF. The interview subjects (potentially correctly) thought that "Register = "Sign up for marketing e-mails."

Spool suggested a change, one which was quickly implemented. He had the Register button relabeled to "Continue" and added a message: "You do not need to create an account to make purchases on our site. Simply click Continue to proceed to checkout. To make your future purchases even faster, you can create an account after checkout."

The number of customers who checking out and completed their purchases increased 45%, and the first month sales increased by $400,000. In the first year after the change, sales went up by over $300 million. CEO response (besides, hopefully, a nice bonus for the work): "Spool! You're the man!"

The lesson is simple: Know your customers. Make the verbiage and functionality align with what the customer want to do and explain things better. Sometimes, the little things can matter. A lot.

Like, potentially, 300 million dollars.

Common Terms (that you should know)

We will cover many of these terms in more detail later on.

Application: A compiled (or not) program that allows users to accomplish one or more tasks. You will often see this term aligned with the medium of delivery: “Desktop Application”, ”Web Application”, “Mobile Application”, etc.

Fitts’ law: “A theory on human mechanics as it pertains to aimed movement” About the size and distance to the target, so you can model the distance and accuracy of use.

Interaction Design: Interaction Design involves how users... interact with a system. What does it do? How does the system “react” to input?

Information Architecture: Information Architecture is the art and science or organizing and labeling data to make it usable and findable

Personas: Personas are “representative users” that you can use as “targets” for your designs. “Will Clara be able to use this?” Personas provide team members an “empathy point” that can be used to ensure user needs are being met.

Usability: The general ease in which a user can accomplish a task in a system.

Usability testing: Testing of designs in order to make sure they are usable, understandable and align with users perceptions and needs.

User Research: Interviewing and investigating people in order to understand their needs, drives and motivations so the designs that are created align with those factors.

UX's Greatest Secrets, Revealed!

I'm a big fan of Penn and Teller, and have been for many years. I saw them live for the first time last month and was blown away by their performance.

What I really love about Penn and Teller is that they often "pull back the curtain" and reveal how they do their magic. Other magicians produce an air of mysticism and pretense around their craft, but Penn and Teller will have none of that. They know they are playing tricks, fooling the audience, and by letting everyone in on what they are doing they debunk mysticism while also (hopefully) teaching you something.

Their attitude towards their work inspired me to write an article that hopefully "pulls back the curtain" on some of user experience

design's "greatest mysteries." Much like Penn and Teller's mocking of "artists" like Criss Angel, I have met quite a few pretentious design types in UX who think way too much of themselves and what they do.

This type of attitude frustrates me, because much of what UX professionals do is actually easy to teach and apply. That's why I've written one book of essays, lessons, and tactics around user experience, and have begun work on a second one: to demystify the domain and make it accessible to all.

To that end, here are some of the "secrets" of UX.

We aren't all-knowing prophets

I know many UX designers present themselves as unquestionable experts on human beings; as seers whose edicts should be followed to the letter. Come on. First off, no one can be that good. If you think you are, such arrogance will in all likelihood prevent you from seeing some basic truths about human behavior, that is to say truths that don't align with your worldview. The key to success in UX is to start from a place of humble confidence, not arrogance.

User experience designers don't design user experiences

UX designers don't design experiences, experiences happen when users encounter a situation and respond to it. They can respond well or badly. All UX designers can do is understand users well enough to design a series of objects, interactions, and/or screens that make sense and work for users—hopefully provoking a positive "experience" in users.

The most important UX skills are "soft skills"

There's an old saying often attributed to Woody Allen, "90% of life is just showing up." I'd say if you are a UX professional, a good chunk of the remaining 10 percent consist of the soft skills of listening, empathy, and communication. Listen in order to understand the problem and listen to the feedback from users to formulate solutions. Have empathy towards users, so that you can

care about what you are doing to help them. Be a good communicator so that you can message your solution and discuss it with the people who have to execute it.

Failing is awesome

In our society failure has a stigma: if you fail then somehow you are a "loser." A huge part of UX design involves conceptual design and user testing, and when we fail (if we are paying attention) we can learn from it. At the very least, when a design we pilot doesn't work we know that particular approach doesn't work. Some of the best insights into people and their behaviors have come from testing designs that users did not understand and/or could not use.

Simple is hard

In my experience, the best designs are simple designs. Creating simple designs is really hard. Not only do you have to fight your own instinct to make things more complicated than they need to be, you also have to fight that same instinct in everyone you are working with, including stakeholders and product managers. Which is why you need...

Debate skills

Not only do good UX people have good communication skills, they can also sell their points in the face of critics and cynics who don't like the solutions they've produced. At the same time, an open "balanced" designer can debate a point but also accept when they are wrong. Because...

It's not rocket science

Yes, when you do formal usability testing or user research, there is a certain amount of analysis involved, but it's not that hard. If there were a lot of math involved, frankly, I wouldn't be doing it... because I suck at math.

Common sense is the best tool a designer has

If a design solution that you or a peer creates doesn't "make sense" then it probably won't make sense to the end users. A big dose of common sense helps designers filter out good ideas from the bad ones. Don't try and "sell" a design that requires leaps in logic and over-thinking things.

UI Design is not that important

I've designed screens and interfaces for over a decade, and what I'm suggesting might make some of my colleagues mad, but I'm going to put it out there: UI design is not that important, and it's not even that hard. It may have been hard ten years ago, but we now have a plethora of design patterns and best practices available for our review and use. We have design guidelines from every major software platform.

Focus (and sweat) the details, yes, but don't try and rethink things that people who are smarter than you have already figured out.

The best designers know users, not UIs

UI design is not hard. Understanding users and figuring out how to create designs that make their lives better — THAT'S what's hard. The best designers spend their time trying to understand who they are designing for by doing research and interviews. "Know your audience" is a common statement I have heard in multiple domains, and it's absolutely true in UX design as well.

Usability testing can be done (almost) anywhere

I have built three different usability labs in my career, and have done the majority of my usability testing with one laptop in a quiet area (a coffee shop, an empty office). Building out a huge technological terror to do something as simple as testing an initial concept with some users is unnecessary overkill.

Developers can do UX Design, too (and many are really good at it)

I've never been a big fan of "silos" in projects, where a dedicated group of designers do a design and then they hand it over to developers to build it. That's why I like working in an agile team, where developers pitch design ideas just like us UXers do. I've met many developers over the years that could create great useable screens, often better than those the "UX professional" made.

If you are on a team but ignore the talents of many on the team because they aren't "accredited" designers or don't have the right title, well, I pity you. You're missing out on some talent that may be right under your nose.

Usability is not enough

We have gone beyond usability. Now, UX designers have to think about desirability, about content, and about how to frame the offering to increase usage. Usability is "table stakes."

We are all storytellers, so tell a good story

Explaining is a huge part of what we do: explaining what we have found out about our users, explaining how we can help users with what we have designed, explaining why a certain feature should be accentuated or de-scoped. The best way to explain is through telling a story, because humans are all storytellers. Some of us are good at it and some of us are not.

The best of us weave a compelling tale, and the best UX professionals know that storytelling is the key to it all.

Typical UX Roles

While UX is still a fairly "young" discipline compared to other occupations, there are some fairly consistent standards as to roles and responsibilities. Here are some typical (and not so typical) jobs that you will find out there, with some descriptive detail and my opinions of each position.

Designer

This is the "starter" position, the one that you'll probably get as your first job. This job is about designing screens and interactions, and involves a lot of sketching, documentation, and refining.

The title is both precise and vague at the same time, because a "Designer" can be a Creative Designer, an Interactive Designer, or a Graphic (aka Visual) Designer. I've seen many companies try and combine one or more of the above into the global "Designer" title, and I've also seen companies be very specific and use titles like the above. I like specificity, myself, so I'd rather have a descriptive title.

Researcher

This is, as the title indicates, a person that researches people – what they like, dislike, need, and do. As most companies are more design and delivery focused than research focus, you'll find a lot less UX researcher positions than Designer jobs. This is changing, though, as more and more companies discover what user research brings to the table.

Information Architect

An Information Architect is someone who creates the information hierarchy that is used in a system or a site. This job is fairly straightforward, but because many people aren't knowledgeable about what IA is this job title is often assigned to jobs that are

really UX Architects (see below). I can see this job being "deprecated" over time, and UX Architect taking its place.

UX Architect

A User Experience Architect works collaboratively with the User Experience team to discover user requirements, and to conceptualize, design and prototype these ideas. He/she will define the "high-level design", which is the blueprint of how a system should work, including the interaction patterns and controls used and the content strategy/tone of voice.

You'll often seen UX Architects working with junior designers, as they define the standards and the designers flesh out the details.

Usability Analyst/Tester

This is a person who is responsible for testing designs with users. The lines between a Usability Analyst and User Researcher is very blurred, but because there are many organizations who consider the roles as separate, I will list this separately as well.

Manager

This role is a manager of a UX team, and usually is more focused on mentoring and project management than actually "hands-on." A UX manager recruits, hires, and mentors user researchers and designer and oversees their performance.

Director of UX

The Director of User Experience is responsible for creating and maintaining a team of design and research professionals, ensuring that product management and development is receiving the desired level of detail to execute effective design. A UX Director is someone who helps define process and sometimes manages multiple teams (A design team, research team, usability test team, etc.)

VP of UX (AKA "Chief Experience Officer")

This is more of a strategic role, in that a VP of UX works on executing a user-centered vision and process throughout the organization. There's not a lot of VPs of UX out there, but as the domain matures I can see many companies building this position out in the next few years.

What role should you focus on? Do what you love. Identify any particular areas you are particularly passionate about, and focus on doing that. What do you like to do? If you like meeting and talking to people, you may want to pursue user research. If you are a detail-oriented individual, you may be better suited to be a designer specializing in detailed design and documentation. Choose the path that works for you.

A few words on Service Design and Content Strategy

User experience design is a pretty broad discipline, with many different areas of focus you can align your career to. There's information architecture, user research, usability testing, requirements modeling, wire framing, prototyping... each with a different focus, approach and outcome.

I personally don't like to specialize on one particular area. I like variety, and I also think that specialization can lead to lost opportunities and less chances for learning and advancement. However there are two UX specialties that are increasingly in demand, with lots of opportunities for professional growth.

These two disciplines, service design and content strategy, are considered by some as an extension or separate from UX altogether. I disagree, but I can also see the demand for these services resulting in them becoming as "big" as UX is now.

Because of this potential, focusing your attention on either one is a good idea, so it's worth spending some time discussing them here.

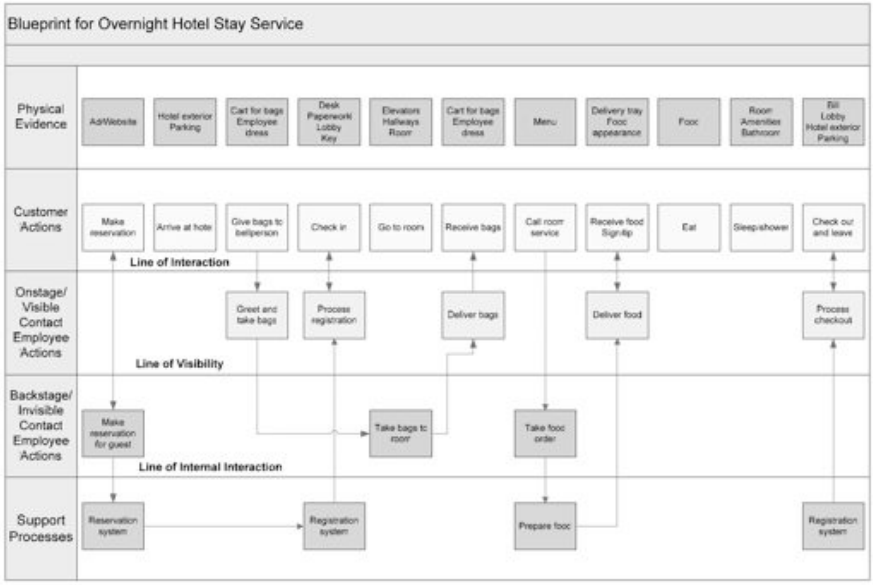
Service Design

When we engage with a company, we want the experience to be a positive one. We don't want to have to jump through hoops to get what we want/need/desire, and (most of us) want's it to be equitable for both parties. How does an organization make sure this happens every time they interact with their customers? They spend time, effort and energy planning out the best way to service their customers... a process called service design.

The service design discipline is about organizing people, processes, and technology to make sure the interaction between a company and its customers are as efficient and "positive" as

possible. It is more than process design, in that it takes a user-centric look at everyone involved in the equation – the customers and the employees who are engaging them.

Service designers plot out interactions in "service blueprints" – maps that track all the moving parts in a way that gives you a "big picture" view of how complex even simple interactions are. What infrastructure needs to exist to support the employee? What information does he/she need to access? What information does the customer bring or need? How many "touchpoints" does the employee have with his colleagues? What training or skills does the employee need to help the customer? Etc.



A service design blueprint

Service design has been around for years, in various forms – McDonalds and many other fast food restaurants looked at service design long before the term or discipline was formalized in the early 1980s. It has become an area of focus for many companies as well as many governments, who are doing service design to improve the way citizens engage with government works.

Content Strategy

What content do you need to create for your website, application or marketing materials? How do you write this content to make sure it makes sense to users, supports their needs, and gives them the information you want them to have? All this and more is covered in the domain of content strategy.

Content strategy is a key focus when it comes to providing users the best experience possible. If you have a great product but lousy "copy" that defines and supports it, you impact the potential success of that product.

The elements of content strategy

It’s not just about writing, it’s about setting standards and planning out when and what type of content is produced. It aligns with information architecture, interaction design, and more. Good content means a good experience, and a focus on content strategy would be a good career for aspiring writers out there.

In a way, service design is user experience design writ large... it’s about defining and optimizing all experiences a company provides, not just one app or website. It also requires a deep skillset and a lot of experience in multiple disciplines. But if you’re like me, and have a lot of interest and/or experience doing different things, it may be a good fit.

Closing

The great thing about the user experience domain is that it contains multitudes, different areas that you can focus on. Because of this variety, you can engage and add value based on your specific interests and talents. And whether it’s service design, content strategy or another sub (or related) discipline, they all have one thing in common: creating good experiences for people that makes their lives better.

On process and people

A good process is important. It allows you to produce repeatable results and hone your skills with practice. But no matter what the process is, a process that does not keep people in mind is doomed to failure. In this chapter, we cover the user-centered design process and how user experience professionals work with people who help make their designs a reality.

The User-Centered Design Process

One of the things I get asked a lot by people interested in user experience design is “How do you do what you do?” The sometimes want specifics, but often the question is broad... what is your process? So, let’s talk about design process.

A typical UX design process

NOTE: The process that I am detailing here is high-level, and intentionally so – we will dive into the details in later sections of UX 101. Also this process is “generic” – many design teams have their own approach and process different than below, and that’s fine. The process I’m covering here is a good start to understanding what designer should do and when, and we should all be ready to accept different ways of doing things.

Step 1: People

It all starts with people. It’s called USER experience design, after all. Whatever functionality or tool needs to be designed, it needs to be designed with the end-user in mind. Producing functionality that

doesn't align and solve actual user needs results in the creation of offerings that have no audience... and no success.

So, how do you (to paraphrase) "People First" design? You do something that is actually quite difficult for some people to do... you start talking, and listening, to other people.

Later on, I'll cover what it takes to be a good UX designer, and one of the main characteristics I'll flesh out is curiosity and empathy. You have to be curious about other people's problems, and empathetic to their problems and challenges. If you care about people, and want to make the world a better place... you have the makings of a solid UX professional.

The tactical ways you can get a sense of what users want/need/ desire are many: reading some books on cognitive psychology can give you some academic grounding, you can have informal focus groups with friends and family, you can do formal research and ethnography, you can look at analytical data, you can look at message boards and forums on the Internet... lot's of weapons in your quiver. The key weapon, as noted above... start talking, and more important, be listening.

(Note: some large design projects require the creation of an "Information Architecture", which is a visualization of the information that is being presented in the final experience that is being designed. As this should be a "user-centered" activity (you want the structure to align with the way users think of the information, for ease of use and navigation) I consider this part of the "People" phase. You're welcome to disagree...)

Step 2: Sketching and creating candidate solutions

"But, I can't draw!" Some of you might think. Nonsense. When I say sketching, it could be on paper, using a design tool, on a whiteboard, or even a narrative outline in word. The key to this step in the design process is to create candidate solutions to whatever you are designing, "getting down" your thoughts in a way that is understandable and sharable. If you can't draw or build

out a design, then partner with someone who can. A good design team has people who partners with each other, and as they say, “a rising tide lifts all boats.”

One thing to remember in this step is that perfect is the enemy of good. This is not the time to tweak and hone your sketches... it’s about creating and ideating with no restraints. Try and think of the ideal experience for the user, and yes, many of your ideas may not be feasible or practical... but that’s fine. You’ll have to face the hard fist of technical feasibility later on.

The key to this step is to start out by really brainstorming and coming up with a lot of different designs and approaches. You want to create options because if you settle too quickly on one approach, you may quickly find that approach fails when you get to the next step, and you are (quite literally) “back to the drawing board”.

Step 3: Testing

This is where the lessons will be learned, the moment where you present your candidate solutions to test participants and you find out how right or wrong you are. You will find out, in all likelihood, that many of the Great Ideas you have don’t work or make a lick of sense to the end user. Taking your design concepts into usability tests is often humbling... but it’s also a learning experience as well.

There’s many ways you can capture user feedback, both formal and informal. Many UX professionals call initial user testing “formative” and the “second pass” I detail below “summative...” no matter what you call it, the key is to get feedback and identify what works and what doesn’t.

Depending on the scope of what you are designing, you may have to do two or more rounds of initial testing to zero in on an effective design solution.

Step 4: Revising and extending your design

By this point in the process, you should have "landed" at one or two candidate solutions that have tested well... though tweaks are needed. That's what you'll be doing here, fleshing out the solutions and making the design more substantial and "real." You can do this in many different ways, and many people start formal design documentation at this point. I prefer to create functional prototypes of the design so that when we validate it in the next step, it has a higher degree of "verisimilitude" to the test participants.

(I also like using the word "verisimilitude.")

Step 5: Design validation

Again, get people involved! Have them try and use the design you have created to accomplish the task the tool is intended to support. These tests will be more formal than the early, more exploratory testing, and you will still find issues and "bugs" in the design. And that's fine, you still have time to tweak things before you do the last step...

Step 6: Documentation and delivery

This is where the experience you have designed is detailed to the nth degree, and delivered to the people who are responsible for executing the solution, which is usually developers. The key is to not "over" document, but to detail the design as much as needed but no more or less. When it doubt, ASK. The team receiving your documentation will be able to tell you the level of detail that is required... and it may be more than you expect.

Remember, design documentation is NOT the final deliverable... the product that is being created is, so don't look at design documentation as an end in and of itself, but as a means to an end.

Final thoughts on process

The process detailed above is a holistic take on UX design, and it does not cover how it works in different types of processes such as Agile or the traditional "waterfall" method. We'll cover those

processes, as well as “Lean” UX, later. Also not discussed is stakeholder communication and management throughout the process... that is a big enough topic that it deserves a separate discussion.

On working with developers and project managers

No man is an island – to be an effective UX professional you need to work and partner with people with different backgrounds, skills, and priorities. If you don’t build solid relationships with the many people responsible for executing the design ideas you have created, there’s the very real possibility that none of your ideas will ever reach the end-user.

This article covers some of the people you will work with, and some hints on how to build the working relationships you will need to produce results.

Project managers

Project managers are the “schedule-keepers”... the person who is responsible and accountable for the on-time delivery of the project. They are usually very organized and structured people, and they tend to be “Type A” personalities. Finally, many of them don’t like uncertainty... which, when it comes to an iterative design process, can sometimes be a challenge.

A good project manager will challenge you and want to understand your design process in great detail... not only so that he/she can make sure all the steps that are in said process is in the project plan, but also so that they can communicate the process to stakeholders and in project updates.

It's important to "overcommunicate" with project managers, especially early on, during project planning. Let them know what you are doing when, and more importantly WHY these activities are important. A project manager who is misinformed and/or has never done a UX project before may push to compress design activities, or jump to detailed design with no user testing. Push back – hard – against this if it happens: skipping steps usually results in a less-than-optimal final design.

Developers

There's a cliché about developers that you frequently see on TV and movies: developers are anti-social nerds who don't like human contact, obese men with neckbeards who slams down Mountain Dew while they spend 12 hours a day coding with Cheetos-dust covering their fingers. Like most clichés, it's simply not true.

Yes, you will find some anti-social eccentric developers out there, the same way you will find eccentrics in every profession. But the idea that developers are these second-class odd code monkeys is insulting and untrue. I've worked with a lot of developers in my time, and in most instances I have found them to be incredibly smart friendly and creative people.

Yes, I used the word CREATIVE, because it's true... software development is a creative exercise in the same way that user experience design is, just in a different way and with different tools. If you think it's not... well, just try and design a model-vie-controller based UI architecture that scales to millions of users.

Go ahead. I'll wait here...

When you are working with developers, understand that they have ideas, too... sometimes very good ones. And while you will be tempted to be very dogmatic with your designs when you hand them over to be coded, try and be open to their feedback. Explain WHY certain design decisions were made, so that they know the "back-story" about what drove the design direction.

Optimally, engage with developers throughout your design process - collaborate with them, listen to their ideas, and most importantly... discuss the viability of your design ideas. The last thing you want to have happen is you and/or your design team comes up with a design approach that isn't technically possible based on the constraints that exist.

Finally, make sure you properly document your design, so that there is no ambiguity. Developers (and after them, QA) don't like ambiguity, because that usually means bugs and rework.

Closing

I started this section with one cliché, and I'll spout another one to wrap up: There is no I in team... successful execution of a UX design means active conversations and collaboration with the rest of the team. Don't work in a vacuum, partner with your team leads to increase the likelihood of success.

On Software Development Life Cycles

Why a section on software development process in an introduction to user experience? Because you will need to know and integrate with such processes as part of your job as a UX professional, that's why. You will need to know the basic software development lifecycle (or SDLC) that your company or team uses in order to understand what you will do and when.

While there are many "variations on a theme", there are only two different types of SDLCs: "Waterfall" and Agile. We will cover both at a fairly high level here, and recommend you do additional research on your own (especially before you join a software development projects).

"Waterfall"

This is the "tried and true" method of software development, which is a stepped approach. This process is incremental, in that each phase must be completed before the next phase begins. The "timekeeper" is the project manager, who often creates incredibly complicated project plans to track and report where the team are to stakeholders.

There are typically five phases in the Waterfall process:

Requirements: The gathering and documentation of requirements, usually in the form of narrative documents called use cases.

Design: The creation of the screens and interactions that is used by the users.

Construction (aka Implementation): The coding and development of the designed solution/offering.

Testing: The formal testing of the solution/offering by dedicated quality assurance team members, who use test cases (often extrapolated and mapped to use cases).

Production: The “shipping” of the offering to the end users

The Waterfall process has come under a great amount of criticism, for several reasons: it’s front-loaded and heavy in documentation that needs to be interpreted by developers and testers... and this can result in unnecessary complexity and “defects.” Additionally, when schedules are compressed this takes away from test or development team.

UX activities in a waterfall process can start during the requirements stage, but most of the UX process will take place – appropriately enough – in the design phase.

Agile

Also known as XP or Scrum, this is a more accelerated method of development that is more evolutionary than waterfall. Instead of “figuring everything out” in a Requirements phase, this process allows teams to start coding with a very thin set of requirements... often high level ones. The project team works closely together, with designers and developers frequently collaborating on whiteboards and in “code” to create the initial design.

The work takes place in iterations or “sprints” that usually last two weeks, with clear goals in mind for each sprint. Unlike “Waterfall,” activities such as requirements definition, design and development take place simultaneously, with the goal of having something tangible to show stakeholders at the end of every sprint.

Agile is very much an “evolutionary” process, focused on iterating and incrementally improving versus a more formal domain-driven process like Waterfall. As such, and since UX is an iterative process, aligning with an agile process should be easy... Except it’s not.

Agile has been criticized by some as being TOO rapid, without a lot of time to do the up-front analysis that is called for in a full UX process. It's also very feature-focused, and this sometimes means that the experience design is given less effort and emphasis.

Closing

Advocates of the different SDLCs will always argue that their method is the "right" way to product software. Who's right? It depends, in the end, on preference, the timelines and talents of the team, and scope and size of the software is being developed. But UX design is an important part of ALL software development processes. Not all software organizations are convinced of this, yet... but we're getting there.

Stakeholders and you!

In any design project, you are going to have one or more stakeholders. These are the people who sponsored the project, and/or are the product managers responsible for the product being design, and/or are VPs responsible for a particular division... No matter what their role in the organization, they are the people you are going to need to make happy.

And that can be a challenge.

Because everyone is different, you are going to encounter a lot of different ways stakeholders approach design projects. Some are going to be hands-off, confident in you and your team's abilities to execute and produce a quality design... and others are going to want daily updates on process. You are going to have to deal with both extremes, and smile while doing so.

Here's some hints on the "care and feeding" of the people who may be writing your paycheck:

Use "The Platinum Rule"

A favorite book of mine, long out of print, was called the Platinum Rule. The premise of the book took the Golden Rule and turned it on its head... "Do onto others as they would like to be done to." Working with stakeholders will require that you adjust your style and messaging to align with what they want and need. Don't be dishonest or "phony"... just provide the information they want/need in the way they request/desire.

"Research" your stakeholder(s)

Use some of your interview and analytical skills to get a sense of what the stakeholder's work personality is like. Are they detail oriented? Are they emotional or analytical? A good way to think about this is to research the different personality types that are

detailed in the DISC personality tests, and figure out what personality the stakeholder best aligns with. It's also a good idea to take a DISC personality assessment yourself.

If you can't guess, ask

A lot of people aren't comfortable with the simple act of asking what is expected of them, because they are afraid they will "lose face" and appear to be unprofessional. Nonsense. If you can't identify how the stakeholder wants to be kept informed as to progress, ask them what they expect or need. It's smart, and it works.

Understand the business goals (and speak the stakeholder's language)

Design and UX activities don't exist just because companies want to spend money on them; they exist because of one or more business needs. Identify what business needs the design is servicing to understand what is important to the stakeholder.

One of my early experiences was on a project where the acronym "NPS" was bandied about constantly. I learned it stood for Net Promoter Score, which was a huge priority for the company sponsoring our efforts. Armed with that insight I was able to "speak the same language" as my stakeholders, which improved communication and allowed for deeper conversations.

Educate (a little or a lot)

Some stakeholders will come to the table with grounding in user experience, either from a previous experience or some casual research – sometimes the stakeholder is a UX professional themselves. Others will not even know how to spell "UX." Again, use the engagements as an opportunity to share and "cross-train" the stakeholder. You've made the effort to understand business goals, so hopefully they can make the effort to get a better sense of UX.

Don’t compromise research results

You may need to change your communication style and approach for some stakeholders, but the one thing you should never NEVER do is compromise any user research results. Don’t change the outputs to align with a stakeholder’s worldview, to try and make them “happier” with the outcome. The results are the results, and they should not be slanted... they should speak for themselves, good or bad. Just be ready to explain negative results to stakeholders, supporting evidence in hand.

It’s not a competition

Don’t get into “fights” with stakeholders. You can disagree and debate some points, but fights... that will end badly. Probably for you. Will you get mad at a stakeholder at some point in your professional career? Absolutely. But mind your words, and bite your tongue, before what you say comes back to bite YOU.

Understand their goals

You and your stakeholder(s) should have the same goal, and if you don’t... well, you need to work through that. The simple way to that is to talk about it with the stakeholder, listen to what their goals are, and make sure that your perspective has a proper hearing. They may not understand and/or disagree... but at least you’ve had your say.

Know Your Users! Research Tactics and Processes

Who are you working for? Yes, when you have a job you have a manager who works for a company who pays you on a regular basis (or, at least, I hope you do). However, when you are a UX professional you don't really work for that person. You work for your users: All the efforts you put in should be focused on making their lives better. How can you be a better "employee" to that boss? Get to know them better...

Different Research Methods

User research is the one thing that you should never skip in a user experience process. It provides insights and understanding into the motivations and drivers of existing (and potential) customers and it is a key input into any design. How much user research you do, however, is very much dependent on your company, project, and timelines. So, be prepared to have many different tools in your quiver, to respond to these varying situations.

There are many different ways you can get user insights, and I detail some of them below.

Collaborative Design

This is when you basically "co-design" with a participant, after framing the problem space and desired functionality in a clear and simple way. The sketching and design part is not the important part (users aren't designers) – the important part is the conversation that foes along with the design activity. It allows you to get a sense of what the person likes and dislikes, and that's a key input into the "intellectual capital" you need to produce effective designs.

User Interviews

The "bread and butter" of user research... the user interview. You spend one to three hours talking to a person about their life: What they like, what they dislike, what they love. You go in-depth in the topic you are researching, even to the extent that you (if they let you) go through their home to look at their environment. The key to this, and most of these research methods, is active listening. Have an engaged conversation with the person to really understand who they are and where they are coming from.

For logistical (and in some companies, legal) reasons, you should have a colleague accompany you and take notes during the interview. Video and/or audio record the interview, so you can go

back and double-check points later (and make sure you have the participant sign a release form that they agreed to be recorded). Finally, come to the table with a pre-written set of questions and topics that you want to cover, but don't be afraid to go "off script" to get into the details of their responses.

Ethnography

This involves shadowing users for days or (if time and budget allows) weeks, to get a sense of what their lives and workflow is. This is (in most instances) work related, so you'll be monitoring several people in their nine-to-five jobs. The key to this type of research is to be a "fly on the wall" and capture EVERYTHING.

Yes, this is an odd situation at first, and many people revise their behavior because they are "on camera." You'd be surprised how often participants forget you are there, however... even if you are RIGHT THERE, four feet away.

In the many different ethnographic research projects I've done in the past, I sat and typed notes until my fingers felt like they would fall off... And even then, I went back to the video and audio I had recorded to review what I had captured to find more insights. This is a time and labor-intensive approach, so keep that in mind.

"Guerrilla" research

If you can't locate existing customers, you can have some quick "friends and family" interviews that can let you. While less formal that typical user research, you can definitely get some insights from such conversations.

Focus Group

This method has its challenges, in that it's... not very good. Yes, you will in all probably get some information by the end, but you will also suffer through a lot of "groupthink" and bickering. You may also suffer from one of the participants effectively "taking over" the conversation because of their dominant personality.

Focus groups have their place... but (in my opinion) user experience research is not one of them.

Stakeholder Interview

While not the user, many stakeholders in your company have insight into the "voice of the customer", either through market research, trade shows, or feedback. They can also provide the business perspective, which is an important input into what you need to do in the final design.

Diary Study

I have had great success with diary studies, where users record what they do for a short period of time. It provides the sort of "from the horse's mouth" insights that you may not be able to get from ethnography or interviews. It's hard to get people to commit to such an effort, though, so be aware you'll have to do spend more time recruiting.

What method should you use?

I'll once again use a phrase that is a vital part of any user experience professional's library: It depends. It depends on your research goals and the needs of the project. If the project is a very large redesign of an existing product, application or website then I would spend a lot of time doing user interviews and ethnography. If the project is a "quick hit" revision, then I would do some stakeholder interviews and some "guerrilla" research to inform your work.

Again, find the approach that fits both your circumstances and your comfort level, but always remember: you are taking time out of people's lives, so please, be kind and respectful of their help.

“Selling” research to management

User research is a vital part of any design project, because if you don’t know who you are designing for, how can you be sure you’re creating the right solution? They are your real boss, the ones who are the most affected if you do a good or a bad job... understand their needs will help you do more of the former and less of the latter.

Then there are the people who actually sign your paycheck... And many times those decision makers and managers will the need for user research. “We already know our customers, we spend X dollars a month in marketing research, we have online feedback forms, etc.” You are going to have prepare to make your case to these people, and here’s a few hints on how to do so:

Try not to play the “ROI” game

ROI stands for “Return on Investment” and it’s one of the more frustrating buzzwords in the business world. It’s often used as an excuse to take risks or spend money, a crutch middle-managers use to say no without taking responsibility.

It’s always tempting to try and force ROI arguments when you are discussing user research, but it’s a fools errand: you end up spending time and energy forcing a point that is hard to prove. Best to acknowledge that there is no direct traceable ROI that comes from user research and move on.

(I’m not saying that user research doesn’t provide value or ROI, I’m just saying it’s a very difficult thing to respond to and the time and effort spent trying to track and rationalize is better served designing.)

If you want to play the ROI game, have data to back up your argument.

User research produces intellectual property

Another buzzword, but a good and accurate one this time! Doing user research increases your knowledge of the existing or potential customers for your the company's products, and therefore will build the institutional knowledge and understanding. This is a great tact if you work in a "data-centric" culture.

The baseline is (always) shifting

We live in exciting times, with new products and technological innovations being released every day. While the core human needs of food, shelter and safety will never change, the specific needs and desires of people evolve in response to the new normal these innovations bring... a new baseline of human expectations User research – REGULAR user research – will provide insights and need standings into where people's "heads" are at, so the offerings you produce can align and meet this baseline. Doing it once isn't enough.

"You can't make bricks without clay"

This is a variation of one of my favorite quotes, from the writing pen of Arthur Conan Doyle. In a train-car on their way to investigate a case, Holmes says this statement with some exasperation. He cannot come up with a theory to solve the case because he simply doesn't have enough data... Which is why they are traveling by train to the scene of the crime. User research is very much the same thing... You cannot produce effective solutions and designs for users until you have a better sense of what the user needs.

If we don't do it, our competitors will!

This argument works a lot, and is an easy card to deal... Even easier if you can point to research results or white papers that the

direct competitor has made public. It appeals to the competitive instinct, and user research can help keep up with or beat the competition.

Quantitative data and market research isn't enough

This is an argument to rebut the "we already do consumer research" statement in the introduction... Lots of people think that marketing research is the same as user research. It's not. Market research is, in most cases, one-dimensional... and as Dr. Gregory House once said, "people lie." Direct interviews and observation will bring you deep insights and understandings that the results of a survey cannot.

(Please note that I am not "dismissing" marketing data... far from it. If you have data, use it. Analyze it from a UX perspective, study it thoroughly. I was once on a project that had extensive legacy marketing data and research, and we reviewed it until our eyes were blurry – it gave us a lot of good ideas and informed areas to explore through formal user research.)

It can identify new product or service offerings

I've done several studies that provided great ideas for new products or services, ideas that would never have been generated if not for the research. We are detached from the workflows of users, and what they do and need is often misaligned with the tools and products they use.

There you have it, a number of points you can make to build your case for user research. In a perfect world, the value and benefits of such research wouldn't need to be "sold"... but this isn't such a world.

Analysis

Once you have gathered your user data, through either user interviews or usability testing, you need to analyze it. While some may consider this a daunting task, it's not: it is, however, an activity that requires focus and objectivity. In order to make sure that you are not unconsciously letting your preconceived notions taint your analysis and final findings, do the analysis activity with at least one other person – preferably two or three. This way everyone can "check" each other's interpretation of the data and mitigate the chance that the results will contain fallacies or mistakes.

The most important thing about analysis is this: Don't reach any conclusions before you start analysis! When you do you will (unconsciously) try to make the data fit your premise and you will (unintentionally) distort the data to fit that preconceived notion.

Always remember: the data doesn't care if you're right or wrong, its just data. I have repeatedly been pleasantly surprised and delighted in research analysis sessions when the data revealed insights and understanding that were totally unexpected and cool – and we would never have identified those insights if we had come to the data with "blinders" on.

The second thing to be mindful of is "pattern recognition." We are wired to make connections... sometimes when no connections exist. Analysis involves interpreting data to form insights and findings, and a lot of that involves identifying patterns and trends in the data... but don't make false connections that aren't there just because you want to find such things. Again, having more than one person do the analysis helps to mitigate any such "incorrect" pattern recognition.

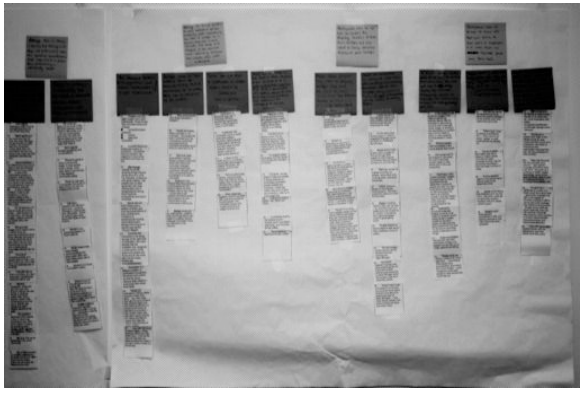
Analysis methods

When you are looking at data from user research or testing, there are multiple techniques you can use to analyze the data. No matter what technique you use, there will always be common activities you will always have to do. You will always need to review the notes and audio/video of the session to get a sense of the person and their responses. Spends some time identifying any pain-points or frustrations that were captured and the root cause of the frustration – was it the software or process being used or was it an underlying issue the person has outside of that? Finally, when you are looking at usability test data, be fair and objective – you need to identify what worked and what didn't in order to produce accurate findings.

Use one or more of the following data analysis, and be aware there are pros and cons to all of them:

Affinity diagrams (aka Card Sorting)

Card sorting allows you to write down individual data points on sticky notes and then put them on the wall. This allows you to start looking at the data in isolation, without any preconceived notions. This is a good way to identify patterns and works especially well when you are building an information architecture or creating personas based on attributes of interview subjects.



Another benefit of this technique is that you can bring in other people to look at the data and let them organize it for and with you. This is called an open card sort, when participants are asked to sort

cards with no pre-established groupings. The groups they create reflect how they think of the data and after they are done they are asked to group the cards and describe the groups they created. (A closed card sort is when participants are asked to sort cards into groups provided to them).

Mind Mapping

Doing a mind-map allows you to create a visual map of the information that you gathered through your research, and this "picture" allows you and your analysis team an opportunity to look at things differently. You can use a software-tool or, if you are artistically inclined, you can create a mind-map on a whiteboard or on large sheets of paper.

Visualizing the information helps identify patterns and informs insights that may not be understood otherwise (remember: many people are visual learners, and this exercise works extremely well for those type of people).

Data-crunching

This is using tools like Excel to look at the data to identify patterns. When doing usability testing, I like to use a standard spreadsheet template that has columns that list the task being tested, an area for notes, a drop-list that allows the notetaker to classify the note as they are typing ("usability issue", "participant

question", etc.) and a "flag" for whether or the person was successful in the task. This allows me to reconcile all the notes from the sessions to quickly analyze what worked, what didn't, and what issues the participant encountered.

When you are looking at notes from user interviews, you will have to spend some time "retyping" the handwritten notes into excel, so this approach has some extra effort baked in. However, this lets you have a permanent electronic effort of all the interview notes, so this is a benefit over card sorting. Another benefit is you can produce nice charts and graphs from the data and many stakeholders like that kind of thing...

Dimensions

"Dimensions" is a good tool for identifying patterns to inform personas. You look at the data you have gathered from all the user interviews and you define key characteristics that came out of these conversations. Some examples are "Tech savviness", "Confidence", "Extrovert/Introvert" and "Charitable giving". You then identify the two ends of the dimension and you place all the interview subjects on the line where they fall. This allows you to see where there are similarities and where there are differences, and this informs the creation of more accurate representative personas. Other than persona creation, however, it has limited application for other design or research activities.

Forming results and making recommendations

After you spend the proper amount of due diligence analyzing the data and forming results (you'll know you're done when your fingers are numb and your eyes feel like they are bleeding) you will need to package your results and define your recommendations. In the past I've written detailed and lengthy word documents as well as large PowerPoint presentations... and I've found PowerPoint (or Keynote if you use OS X) works best.

Business stakeholders like to see presentations and many of them have looked at my lengthy word documents with a reaction

bordering on contempt. “I have no time to read that,” they say, “give me a 10-page summary.” Do not take offense at such a reaction – most senior folks like this are like Jack Webb on Dragnet: They want “just the facts.”

You may be working on a project with an aggressive timelines, and may be tempted to just send the results in a quick e-mail to the key players and the rest of the team. Don’t do it. Spend the time pulling together a formal, professional document, because if you don’t odds are a stakeholder at some point will question the budget line item for user experience research or testing and ask, “Why are we doing this? What are we getting out of this spend?” Having results documented from all your testing will allow you to best respond to this type of question. It will also add to your portfolio, and so it’s worth doing and doing well.

Understanding bias

We are all biased... anyone who says otherwise is fooling themselves or lying. We are all biased because we are all human... We are the sum of all of our experiences, and those "inputs" influences our opinions and interpretation of facts.

As user experience professionals, we are going to encounter biases in our research subjects and in ourselves, when we analyze the information we have gathered through usability tests or research.

Understanding the type of biases that we all have will help you "get over" them and be objective when analyzing you data.

Taken from Wikipedia, here's some of the key biases to watch out for:

Anchoring: The tendency to rely too heavily, or "anchor," on a past reference or on one trait or piece of information when making decisions. As an example, I've seen this in users who won't try out mobile banking because their first experience was so bad.

Bandwagon Effect: the tendency to do (or believe) things because many other people do (or believe) the same. You see this all the time when it comes to user's opinions about technology and certain tech companies.

Confirmation bias: The tendency to search for or interpret information or memories in a way that confirms one's preconceptions. This is why people tend to get their news from sources that reinforces their world view (CNN, Fox News, talk radio, etc.)

Curse of Knowledge: When knowledge of a topic diminishes one's ability to think about it from a less-informed perspective. I've seen this in a lot of user researchers, who can't "unlearn" what they know and look at user data from a fresh perspective.

Empathy Gap: The tendency to underestimate the influence or strength of feelings, in either oneself or others. This is why empathy is a key “soft skill” user experience professionals need to have.

Framing Effect: Drawing different conclusions from the same information, depending on how or by whom that information is presented. This is why con men will always exist.

Hindsight bias: Sometimes called the "I-knew-it-all-along" effect, the tendency to see past events as being predictable at the time those events happened. Remember, kids, correlation is not causation...

Irrational Escalation: Where people justify increased investment in a decision, based on the cumulative prior investment, despite new evidence suggesting that the decision was probably wrong. (Irrational Escalation was actually the original name of Las Vegas.)

Selection Bias: The distortion of a statistical analysis, resulting from the method of collecting samples. If the selection bias is not taken into account then certain conclusions drawn may be wrong. This is why you need to be very thoughtful when it comes to recruiting participants for user research and testing.

Stereotyping: Expecting a member of a group to have certain characteristics without having actual information about that individual. This is, unfortunately, still a very persistent cognitive bias in people around the world.

(More here: http://en.wikipedia.org/wiki/List_of_cognitive_biases)

Avoid bias two ways: First by, educating yourself on the biases that we have (and hopefully this list is a good start) and second, when doing any analysis have more than three people involved so that they can “police” each other. It may not mitigate all the bias, but it will help minimize any impacts the bias has on the results.

Creating and using personas

When I discuss the word persona with people, I get a lot of blank stares and tilted heads. The film buffs I know respond with, “The Bergman film?” and the rest of them just shake their head when I ask them what they think it means in UX. So, let me spend a moment describing the term in the context of user experience.

Personas are representative users that designers, developers, and other team members can use as “targets” for designs and features.

They provide team members an “empathy point” that can be used to ensure user needs are being met when products or websites are being developed. How do you create personas? Read on...

Creating Personas

Personas should not be “imaginary friends”, they should be driven by real people and real data. Personas that are “made up out of whole cloth” don’t have the weight and realism that properly built personas have, so use them at your own risk.

How do you get that data? Talk to people. A LOT of people. Then analyze the data you gather to identify key characteristics, and understand wants/need/desires... get a sense of who these people are AS people, and then document that in personas you create.

And when creating personas, remember... less is more. Don’t try and impress people with a huge amount of information. Remember, unless you are doing a user research project, in most cases the persona is not a final deliverable, it’s a tool that informs design activities and helps communication. So define Persona details that are focused, succinct and usable. I like to keep it simple, and usually only flesh out the following:

Description: Who the persona is (a general sense of the person and the personality)

Goals: The ambitions and desires of the persona

Knowledge/Skill: Capabilities of the persona (in the domain that is being designed)

Start from that foundation, and add the details that are needed to paint a picture of the persona that the team can understand and empathize with.

A typical persona

Using Personas in User Stories and Scenarios

Once you have personas, you need to apply them in User Stories and User Scenarios. What are they? I like to look at things this way (using Title Case for key terms):

• A Persona is a character in a story.

• A User Story is a brief descriptive sentence that details the need/desire of the user. The Persona is the "subject" of the sentence.

• A Scenario is a more detailed paragraph that contains context and details about the situation and the user.

(I use the term User Story in the Agile software development process context - some UX professionals use User Story and Scenario as synonyms.)

Be purposeful and thoughtful about use the Personas properly - you don't want to have a Persona of Jane, a housewife, in a User Story about pouring concrete in a construction project.

Also, the definition of the Persona will inform the creation of new User Stories and Scenarios, because when you do the research to create the personas a natural output of that effort will be an understanding of what they want/need to do every day in their personal and professional lives.

The best way to integrate personas with user stories are to understand that personas are characters in a story, and they have emotions, drivers and motivations just like any characters in any good stories do.

I've often seen the parallels between user experience design and screenwriting, and this is another great example. Good screenplays/stories present characters/personas with situations that they have to overcome or respond to. How they do so shows the measure of the character... And helps the audience/designer understand them better.

So, write user stories and scenarios with your personas in mind and they will help you design the right solution that works for them.

Storyboards: Another way to use personas

If you are not narratively inclined, you can use personas in storyboards. This carries the aforementioned moviemaking metaphor to the next level, in that you are sketching out an experience the persona has in a “comic strip” fashion. This works great if you are an above-average artist, and really helps “tell the story” visually and effectively.

As I’ve stated previously, use the method that works for you and communicates the situation in the most effective way.

1 Child's Doctor / Emergency
However you come to Children's, know what kind of care you can expect, how to find us and who to talk to first.
2 Scheduling
Schedule your appointment in just one phone call, at a time that works for you.
3 Clinic visit
Expect that your visit goes smoothly and that you're in and out without waiting.
4 Before Surgery
Receive and understand all the information you need before your child's surgery.
5 Surgery
We'll start the surgery right on time.
6 After Surgery
Your child will move to a hospital room or go home as soon as they are ready.
7 Inpatient Care
Know when you can go home.
8 Going Home
Have all your questions answered and get clear instructions on how to care for your child.
9 Follow-up visit
Your appointment will begin on time.

Design stuff! Making your ideas a reality

Once you know who you are designing for and what they want and need, it's time to start designing! This is both easier and harder than it looks, and we'll get into specifics on that next...

Design Foundations

User experience isn't "a" skill – it's several of them. UX is a multi- disciplined domain that requires a lot of varied abilities, and the most successful UX professionals are the ones who are good at a lot of things. The phrase "Jack of all trades, master of none" comes to mind, and is a good way of looking at the profession.

This is not to say that if you AREN'T that type of person, you can't make it in UX... Opportunities abound for "single-skilled"

artisans or experts. It's just that having a variety of skills brings more opportunities and makes you more desirable to teams and companies.

We've previously covered some of the soft skills that are important to have, now let's cover some of the foundational things you need to do to become a good designer:

Expose yourself to art

You can do this formally or informally, but it's important to get a broad exposure to different art styles and examples. Get coffee table books, go to museums, and start to identify what art you like and what you don't. When you find art you appreciate, reflect on why you like it... was it the color palette, or the composition, or something else? Do the same for art you didn't like.

Design, like any other creative exercise, is like breathing – you breath in, you breath out. Exposing yourself to art will inform your own creative work, even in an indirect way.

Use technology

Why do I recommend using technology as a foundation to user experience design? Because technology is going to be the "medium" you design in, and if you don't expose yourself to tech

you won't know the conventions and standards you have to design to (or break from). This is not saying you need to be walking around with three different cell phones, two tablets and three laptops... you just can't live in a cave. Find the balance that works for you, and then use tech... a lot.

Get out of your comfort zone

Don't think you are good at drawing? Force yourself to do a sketch every day. Don't like the sound of your voice? Start a podcast. Not comfortable speaking front of people? Join a public speaking group like Toastmasters. By getting out of your comfort zone, you can "stretch" yourself and exercise the parts of your brain that you rarely use. It's like mental exercise, and the more you "limber up" the more your brain will be ready to create and design.

Know the brain you have

Speaking of brains, get a good sense of the type of brain you have. We all have a "dominant" side of our brain, and you can take tests online to see what type of brain you have. This simple insight can help you better understand what characteristics are dominant and therefore will help you "balance out" your design process and approach.

Learn design principles

Understand what the Golden Ratio is, and how artists use it. Get a better sense of composition, and the elements of design. Spend time exploring the rule of thirds, geometry and symmetry. All of these design principles are applicable to UX design.

Understand color

There are colors that work together and colors that don't... Learn about complimentary, warm and cool colors. Investigate about different color schemes, and how they are often applied in different contexts and industries. Get a sense of what color theory is, and how it's applied. Finally, certain colors provoke an emotional

response in people. Learn what these responses are to properly utilize colors in your designs.

Create something

We are creative creatures, man... we have an instinctive innate ability to produce something from nothing. Go with that: Paint, sketch, photograph, sing... be creative, and by being creative you will stretch yourself and become better by doing.

Working with existing designs

I've found that most of the educational materials on user experience design gives the impression that UX design projects will always be a "from scratch, develop a new product or application" effort. In my experience, that's rarely the case.

While it is great to work on a "blue sky" "anything goes" project – and if you get a job in UX you will get to do those type of projects - the typical design project in my experience is usually focused on tweaking something that already exists. Be it a site or app, it's an existing offering that has some history... and, unfortunately, some baggage.

When doing this type of project, there are some important steps you should take in the early days of such an effort. They are:

Learn all that is learnable

Interview stakeholders and other members of the team to find out the history... why is the design the way it is today? You'll find out a lot about not only the decisions that brought the design to where it is now, but also get a sense of the team's attitude about the design. This is very valuable "intelligence" that can help you avoids mistakes that were made before.

Also, take the time to "learn" the legacy design. Step through it, doing the tasks and end-user would. Identify the workflows that exist and understand what the key scenarios are. Study any research and information that informed the final design.

Be objective (and hold your tongue)

Don't go off half-cocked and say, "this sucks" to a stakeholder as a first impression... First off, it may not "suck", you just may not be familiar with the UI conventions, workflows and terminology to understand it at first glance. It may align with the mental models of

the end users quite well – it just doesn't make sense to YOU yet. Also, that stakeholder may LOVE the design... and any flippant "this sucks" response could close off a potential ally.

Do a heuristic review against best practices and design standards

There are some great best practices in UX and UI designs out there – some will be detailed later on in UX101 – and you can easily take them and form a review "checklist" to apply to the existing design. I personally like Jakob Nielsen's 10 usability heuristics, because they are easy to understand and communicate to team members.

Identify contributing factors

Why is the design the way it is? It may not have just been the results of the previous design team's efforts... it's likely technical or platform limitations may have had something to do with it. For example, if a dashboard looks a little "thin" in the content department, it may be because the back-end architecture isn't robust enough to serve up any more data. Talk to the developers about the technology used and any constraints they have, and this will prevent you from making recommendations that simply can't be done.

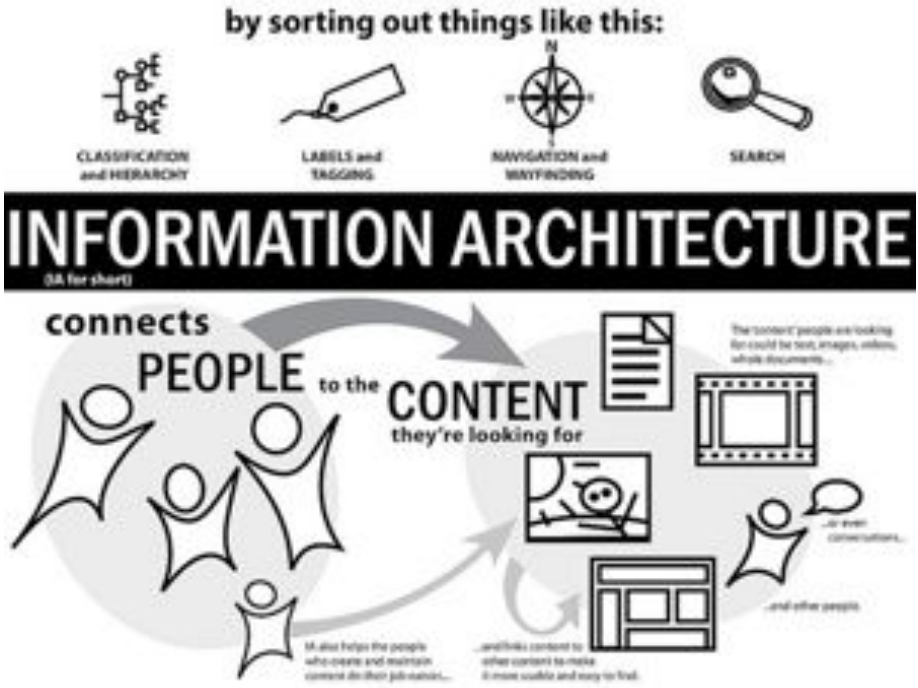
Don't go crazy (and redesign everything)

If you propose to drastic a change, you run the risk of alienating everyone on the team – "That'll take too long! That'll confuse our existing users!" The end result of such a conversation is almost always that nothing happens, which benefits no one. Be strategic, and identify "low hanging fruit"... aspects of the design and the experience that can be tweaked with minimal effort. Build to the big changes gradually, and try and make a difference as much as you can as you go.

Information Architecture

When you read about user experience and the UX design process, you will inevitably see a lot of discussion about Information Architecture. What is Information Architecture? Simply put, Information Architecture is the art and science or organizing and labeling data to make it usable and findable... though its practical application in user experience design is the subject of some debate.

This quote, by noted Information Architect Richard Saul Wurman, adds some color to the above: "I mean architect as in the creating of systemic, structural, and orderly principles to make something work — the thoughtful making of either artifact, or idea, or policy that informs because it is clear."



A key point, and one that is important to note: Information Architecture is not the same thing as "site navigation" or a site map. Navigation is a reflection of and directed by formalized information architecture, and can vary (an information architecture can have information at many levels, while a navigation is usually a "flatter" model that in most cases only has two levels). And a site map is a mapping of content – again, this should be the result of (and be informed by) a formalized IA and, even then, it isn't going to be 100% consistent.

Information Architecture can be applied in all levels of design, from form layout (What fields go together? What order should they be presented?) to Intranet development (how many intranets need to exist? And what should they contain?) and even an automated voice response system (What is the primary information tree callers access? And the second?)

Mental Models

When you are structuring the information architecture for the content and functionality you are providing to your users, you need to think of how those users think of the domain. How do they think of the information? What preconceived models do that have in their heads? A good IA aligns with the mental models of the user.

Indi Young wrote a great book about this very topic. Titled (appropriately enough) Mental Models, the book goes into some detail on ways you can get a sense of how users think of things and what intellectual constructs they have created that your design should reflect. While I will not attempt to replicate her content or detail here, I did want to share some thoughts on how you can align with how people think of things.

I was working on a project recently for a large home improvement retailer, and we discussed the different models that customers "bring to the table" When it comes to what people look for when they visit a website for such a company, what are they looking for? Are they trying to solve a problem, or are they browsing? We came up with four candidate models for the site:

Product-focused. Things, objects, items... what do people go to an e-commerce site to do? Look for things to buy. Following through on this model meant that the information should be presented in a structure that aligns with the products the company offers.

Project-focused. Do people come to the table with a project in mind? Do you structure the content products around the possible projects, and provide easy access to products and tools used to accomplish those projects?

Location-centric. Where do people live? In a home or a rented property. Why not present the information in a way that aligns with the rooms people spend their time? Kitchen, bathroom, pantry, laundry... place content and product information in the context of a familiar "place" that people know.

Service-oriented. What services does the company provide users? How can you frame the information in a way that presents the services and the associated products effectively?

These are just examples of how you can apply different models to build out an information architecture. You may find that more than one model makes sense based on the content you are presenting... a "hybrid" approach. Use this framing concept to kick-start your design conversations and focus your efforts to create the appropriate information architect for your users.

Visualizing Information Architecture

The term "boxes and arrows" is a popular one in UX circles; this descriptive term reflects the many different flow charts, task models, and IA diagrams a UX professional create. Visualizing information architecture is fairly straightforward – use the aforementioned boxes and arrows to structure the various domains of information. Keep it simple – the point of the documentation is to present the domains of information in a way that is easily understandable by the consumers of the information. For some good examples of effective IA documentation, check out the work of Edward Tufte.

Information Architecture, Applied

As noted above, Information Architecture informs the design of many facets of a user experience. It will also "level-set" expectations and help teams understand the depth and breadth of the domain that has to be designed. It can also be used to inform what features are prioritized on a roadmap and what content needs to be created. All in all, Information Architecture is an important "blueprint" that allows you to understand what needs to be done in a design project.

Interaction Design and Models

In user experience conversations you will often hear the term "interaction design" bandied about. What does it mean?

Here's the Wikipedia definition:

Interaction design, often abbreviated IxD, is "about shaping digital things for people's use", alternately defined as "the practice of designing interactive digital products, environments, systems, and services." Like many other design fields, interaction design also has an interest in form but its main focus is on behavior.

Simply put, interaction design involves the creation and definition of how users... interact with a system. What does it do? How does the system "react" to user input? Framed in that way, you can then understand that user interface design is a "sub-discipline" of Interaction Design. The UI is the "platform" that the interaction takes place on, and the objects and elements that users engage with exist on that platform.

This is much less complicated than it may appear, though you shouldn't be ashamed if you are confused. Interaction design is an abstract conceptual idea that is not easy to grasp for some. Try and think of a blank screen... then, think of placing a couple of thin black and white boxes on those screens, one above the other. Then put a button below the two boxes, and put the word Enter in the box. Add two words to the left of the boxes – Username for the top box and Password for the bottom one. You have just designed a log-in screen. That's user interface design.

Then think about how you or any other user would engage with the elements. How would they start to enter the information? Using a mouse to click the text box? Through a keyboard shortcut? Do you want the cursor to be in the first box when the screen loads? Do you want to "hide" the characters that are entered in the password field? How should the screen respond if the wrong password is

entered? Those details start to go into interaction design – how the interface interacts with users and responds to their input. The line is blurry, but hopefully this explanation helps.

Interaction Models

Part of interaction design involves using or creating interaction models. Interaction models are the “rules” around how users interact with a system. An easy way to understand what interaction models are is to look at them as the "metaphor" of how things work. "This works like..."

Why are interaction models important? Because they help users learn a new site or design, especially if the metaphors are obvious and consistent with how they expect things to work based on their experience. UX Designer Dan Saffer had this to say about interaction models:

"A device without an interaction model will likely seem disjointed and made up of pieces, instead of as a whole. Pieces of functionality will work differently and the overall concept will be hard to grasp. Many mobile phones, appliances, and consumer electronics suffer from this problem. A solid interaction model is the basis for any great device."

(From his blog: http://changeorder.typepad.com/weblog/2011/01/the-metaphor-of-the-system-part-2.html)

Good interaction models feels natural and obvious to users. They reflect the real world and align and are consistent with how people see things work and think of how things work. They are thoughtfully defined and internally consistent. Good interaction models keep things simple and (hopefully) obvious.

The most well-known interaction model is the desktop, which is used by all computer operating systems. The desktop UI reflects the real-world desktop, with files and folders representing storage

“places” for information. Since desks have exists for hundreds of years, people can “get” this metaphor quickly and get to work. Another well-known metaphor is in word-processing, where the default UI is a blank page, sized just like a piece of paper.

One site or system can have multiple interaction models present, but they should work together well and not “clash.” A good example of clashing is when an application that “breaks” an existing model – if user are used to scrolling up and down with a mouse in a certain way, don’t change that behavior in an application.

With voice, gesture, and augmented realty new and different interaction models need to be created. Designer are already defining standards and best practices for these new ways of engaging with technology – for one example, check out this article on “New Design Practices for Touch-Free Interactions” from UX Magazine. http://tinyurl.com/newhci

Whatever the future will bring, it will need to be designed... and how things work need to align with how users expect them to work. Effective interaction models will help make new technology be more usable and easier to learn.

Navigation design and best practices

Whether you are designing a website or a desktop or mobile application, you'll need to provide a way for users to get to what they want or need. Hence, you will need to provide some form of navigation.

How should you let users navigate your content or functionality? First, you should consider the conventions of the "platform" you are working in. For example, iOS applications have recommendation navigation conventions you can follow. Whether you follow these conventions or not, at least get familiar with them so that you know the "rules" you may be breaking.

There are several different ways to present navigation items, and examples of these different models conventions are easy to find throughout the web. Let's take it from the "top", shall we?

Top Navigation

This is the de facto standard in web pages for less complicated web sites, in that it provides more horizontal space for content and works well for most web sites. It is easy to scan (since most users read left to right) and provides big "targets" for visitors to click.

Side Navigation

This is more "scalable" in that it provides designers the ability to list more than seven items. It's a little less easy to read and reduces available horizontal space, but it is an effective alternative to top navigation when there are a lot of navigation items.

Drop-down/drop-list Navigation

This navigation model uses a drop-list to display the navigation items, which allows for the navigation area to have a smaller

"footprint" in a UI. The only problem is that "Hides" options from users, and many search engines don't "read" this type of navigation very well... I recommend that this model be avoided.

Icon-based Navigation

This is often seen in mobile apps, and can be presented as headers or footers or the full navigation/icon set be displayed in a "launch page" fashion. This navigation model is limiting, in a couple of ways. First off, you can only put so many navigation items on a screen (and scrolling shouldn't be used with this model).

Second, icon design is not easy – you have to represent the content/ functionality that is accessed through the icon in a way that is simple and obvious, and if this content and functionality is"obscure" it may be difficult to design an icon that represents it affectively.

Search

Yes, search is a method of navigation. In fact, the widespread use of search has "trained" users that this is the quickest way to access information, and the use of traditional navigation has been reduced. This is not to say that traditional navigation should be removed and replaced with a big search box... instead, make sure that the search results that your site or application provides are accurate, well presented, and useful.

Navigation best practices

Now that you know the different types of navigation models, how should you use the one that you have decided works best for you? Here's some "best practices" that you should follow:

Limit menu items to no more than seven (when possible)

Why seven? Because seven is less than eight, and the more navigation items you have the more complex the site or system will appear to users. This will result in "cognitive overload" for many users, and this anxiety over the number of choices provided will often result in no choice being made at all.

This is the reason why Amazon removed the top-level tabbed product category navigation several years ago: Customers were intimidated by too many options. They solved that problem by providing a prominent search box in its place and a (hidden) side navigation section. Follow their lead, and keep things as simple as you can.

Use specific descriptive terms

Don't be generic; be explicit and specific in the terms you use. When people are looking for a section or a piece of information they want to not have to "figure out" what the navigation item means... it should be obvious what it means. Make sure that navigation items align with what users say and use, so that they don't have any confusion.

Avoid jargon

An exception to the above rule: If the specific descriptive term for a content area or piece of functionality is a piece of "jargon" that is not well known to the typical user, replace it with a simpler term that is more obvious and understandable.

If you need "sub-navigation", keep it simple

Depending on the complexity of the site or system you are designing, you may need to provide “sub-navigation”: secondary navigation items that are “children” of the primary navigation. If you do this, only have one level of this secondary navigation, and keep it simple: Again, follow the “no more than seven items” for any specific sub-navigation list.

Supplement navigation with (accurate and quick) search

As previously noted, search is the new way people are accessing information. Present a prominent, accurate and quick search area to support users who ignore navigation and go straight to search.

Align with conventions

Users come to sites and systems with preconceived notions and expectations, and these conventions should be reflected in the navigation elements that are presented. For example: A "Contact" option should be towards the end of the navigation section, because that's where people expect to see it. Changing from this convention just to be “different” will result in additional cognitive load for users, which is not a good thing.

Logo should always bring the user to the home page

This is a no-brainer, and one of the “conventions” that is listed above. Users expect that clicking on the logo will bring the user back to the home page... so always do that.

Use analytics to inform navigation placement

If there is a content area that is more frequently accessed, move it to the “top” of the navigation (which, in the case of top-level navigation, is on the left of the list).

Use “breadcrumb” navigation (if it helps)

Breadcrumb navigation is a small navigation area towards the top of a content area that lists the “path” that the user took to get to the page they are on. This gives users a sense of context and “place”

and allows them to “go back” quickly. The use of this UI element has reduced over the past few years (potentially as a result of the search-based navigation that users are now doing) but is still a helpful convention to apply in many situations.

Creating and using journey maps

When we previously covered personas, we discussed how personas could be used as the subject of user stories or (more detailed) scenarios. These narratives detail what goals the persona wants to achieve and (in the case of scenarios) the detailed steps they go through in order to accomplish said goal.

While this type of "storytelling" approach is simple and effective, it lacks a visual element. While you can flesh out the details in storyboards, many design professionals prefer to detail out personas and scenarios by using journey maps.

The typical journey map is centered around a single persona and a goal, and contains the following content:

The "Story" – this section contains tasks, needs, actions, and obstacles. What does the persona want to accomplish? What do they need to know or do? What actions take place? And finally, what obstacles are in their way? This is usually presented using boxes and arrows with call-outs with further details.

Tools and Touch-points – What tools are used? What touch-points (with customer support representatives) occur?

Emotional Experience – How does the persona respond to each of steps in the experience? Are they thrilled or annoyed?

The creation of journey maps requires a bit of imagination and creativity, and it helps to work from (and understand) any existing processes to inform this effort. Often, if an existing process is being refactored, two journey maps are created – one that details the current state, and the second that envisions an "optimized" experience.

Always remember to use the appropriate persona as the "subject" of the journey map. And be aware that, depending on the level of complexity of the experience, different personas may require

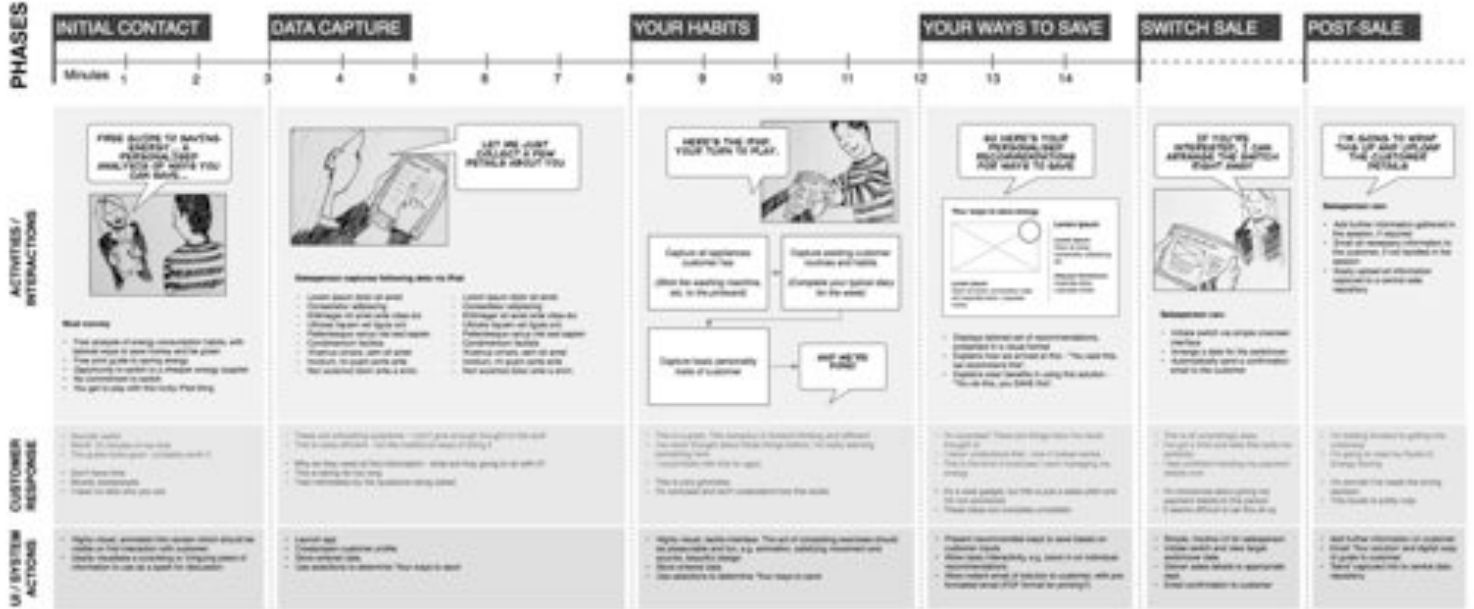
different journey maps for the same goal (based on the roles they have in the process).

Journey maps allows stakeholders to see a visual representation of an intended experience, and how users interact with the proposed system and solution. It's a great communication tool, but is usually not the final deliverable. It is a means to an end – a way of letting people know what the planned design will be and how (optimally) customers will respond to it.

How many journey maps do you need? You'll need one for each major feature that is being designed and implemented. These journey maps will all be "happy path" representations of what the optimal experience is, though you may want to create "unhappy paths" that detail how exceptions or mistakes are handled.

So, what's the best tool – user stories, scenarios, or journey maps? It depends on timelines, talents, and stakeholders. If you have an aggressive timeline, you won't have the time to flesh out full journey maps, and you'll have to rely on simple user stories. However, if you or your team can quickly visualize things through journey maps, then you can still use journey maps. Finally, if your stakeholders are visually minded, you will need to create journey maps to respond to their needs to "see it."

As with other deliverables, always keep in mind that these documents are a means to an end – they are documents that inform the final design, and while it is important that these documents be clear and useful, we don't "ship" design documents – we produce applications and tools that help people get things done in a effective, usable way. Focus on the experience, and let the design documentation support the definition of the best experience possible.


PHASES
INITIAL CONTACT
DATA CAPTURE
YOUR HABITS
YOUR WAYS TO SAVE
SWITCH SALE
POST-SALE
Minutes
1
2
3
4
5
6
7
8
9
10
11
12
13
14
ACTIVITIES / INTERACTIONS
FREE GLOSS TO SAVING ENERGY + PERSONAL TIER ANALYSIS OF WAYS YOU CAN SAVE...
LET ME JUST COLLECT A FEW DETAILS ABOUT YOU
HERE'S THE IPAD YOUR TURN TO PLAY.
AS HERE'S YOUR PERSONALISED RECOMMENDATIONS FOR WAYS TO SAVE
IF YOU'RE INTERESTED, I CAN ARRANGE THE SWITCH RIGHT AWAY
I'M GOING TO WRAP THIS UP AND UPLOAD THE CUSTOMER DETAILS
Capture all appliances customer has
(Mirror the washing machine, etc. in the printboard)
Capture washing customer routines and habits
(Complete your typical day on the week)
Capture basic personality traits of customer
AND HE'S DONE!
CUSTOMER RESPONSE
UI / SYSTEM ACTIONS

Interface design 101

All you have learned about user research, information architecture, and interaction design is the foundation for the next step – execution. Taking all you have learned and defined and then designing screens that let users accomplish tasks and get things done. User interface design... it's where the rubber hits the road.

But compared to everything else you have done so far in the design process, it's actually the easy part.

Whenever I say that many of my peers are flabbergasted by the comment. "Interface design isn't easy!" one once exclaimed in response. "If you do it wrong the whole experience falls apart!" True enough, but it's not like we are starting from a blank slate... We already KNOW what works and doesn't.

How so? Because user interface design isn't a "new" thing. Designers have been designing screens and "dashboards" for decades, and this has built a wealth of knowledge and standards we can apply in the act of interface design. Heck, Paul Fitts defined the famous law named in his honor (on control size and placement) over seventy years ago!

There are best practices like Fitts' Law (many that I have covered in other articles here) that we UX'ers can leverage and use. On top of that are multiple design heuristics that we can follow and compare any proposed UI designs against.

When I look at younger designers try and be "original", they often come up with user interface designs that don't work BECAUSE they don't use that collective knowledge to inform their designs. I state this not to discount original thought, but to properly direct it. If there are best practices on how to build a dashboard, why ignore them and come up with your own? Designers, focus your creativity to better use.

Understand your users, then start designing

So, how do you get started? A clear understanding of the user and their needs is key. This is why you did user research in the first place – don't lose sight of that understanding. You may have already defined and documented user needs though journey maps or storyboards, but if you haven't... take the time to do so. Even a one-paragraph scenario can give you basic context and understanding.

To make sure you are ready to start fleshing out your screens, make sure you have the answer to the following questions:

- Who is this design for?
- What will they be doing?
- When will they use it (how often)?
- Where will they use it?
- Why would they use it (as opposed to another process or application)?

This grounds your design and lets you focus on the user rather than get bogged down in trivial details. The time to sweat those details will come later – focus on getting a good first attempt down.

I prefer to do my "first cut" at an interface design on paper or on a whiteboard, because it's less "permanent." By sketching, you can erase or throw away ideas without much angst – if you are building out designs on the computer many people invest a lot of emotional and intellectual activity in that and so they are more reluctant to "throw away" work done there. As I've said before, use the tool that works for you, but be careful not to be too satisfied with your initial effort... because it will change.

As you are doing this first set of sketches (also known by many as "concept designs"), leverage your experience and knowledge to create what you think works – if you want to look at how other sites or applications do things, feel free. As I started earlier, there are a lot of really good designs in the world, and you should take full advantage of what's out there.

When it comes to what you are designing, there is a difference between content sites and applications. To state the obvious... they are not the same thing. There are different user goals and focus for content sites – it's about consuming vs. doing. While content sites will still have functionality, these interactive elements will be to support and supplement consumption, and should be placed and prioritized as such. Again (and obviously), support the appropriate user focus with your design.

Once you have done a first cut at the conceptual design, this is where you get user feedback. You can formally or informally test your designs with people, and the key focus at this point is to make sure the interface design aligns and supports what users want to do and "makes sense" to them. If you show a design to someone and they can't tell you what they can do with it or how it works, you need to go back to the drawing board.

Detailed design

After gaining some confidence that your concept design is understandable and usable, this is when you flesh out the design and add additional detail. There are multiple tools this can be done in, from Axure to Irise to Visio and Omnigraffle. Again, use the tool that works best for you (and if you are part of a larger design team, the tool that is compatible with what they have or need).

As you flesh out your detailed interface designs, leverage one of the many design pattern libraries that exist. These design patterns detail out scores of UI elements and rules on how to use them. These patterns are usually informed by best practices from web sites and applications, and so when you apply these elements in you are design you are going to be providing elements and

behavior that are familiar to users, reducing confusion and cognitive load.

The most well-known pattern library comes from Yahoo, and some design tools even allow you to import these UI elements into your designs. Depending on the size of the project, you can even “roll your own” and build out your own pattern library as a reference for other designers and developers. As noted above, leverage what already works and repurpose design elements to flesh out your design.

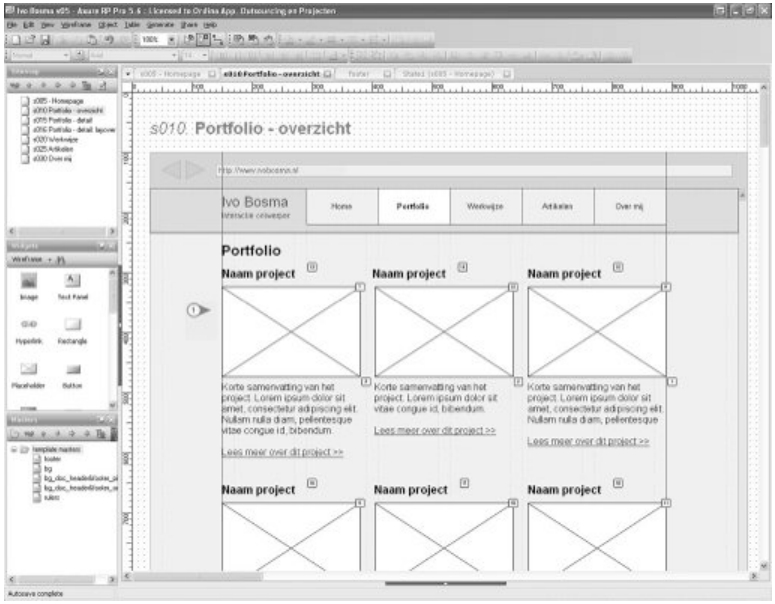
After you have fleshed out your designs in detail, you will want to test them again, this time making them a interactive as possible. Many of the design tools listed above will allow you to create an interactive prototype from your designs, but if you are not comfortable with doing this it may be the time to request development build-out a functional prototype from your designs (Of course, if you are in an Agile software project they may have already started doing this).

After you have such a prototype, you test it – in a much more formal way than the earlier conceptual design. More on how to do that next ...

Tools of the Trade

There are quite a few tools that you can use to create designs with users, and in my many many years as a UX professional I've tried out most of them. Here's the most popular design and prototyping tools:

Axure RP



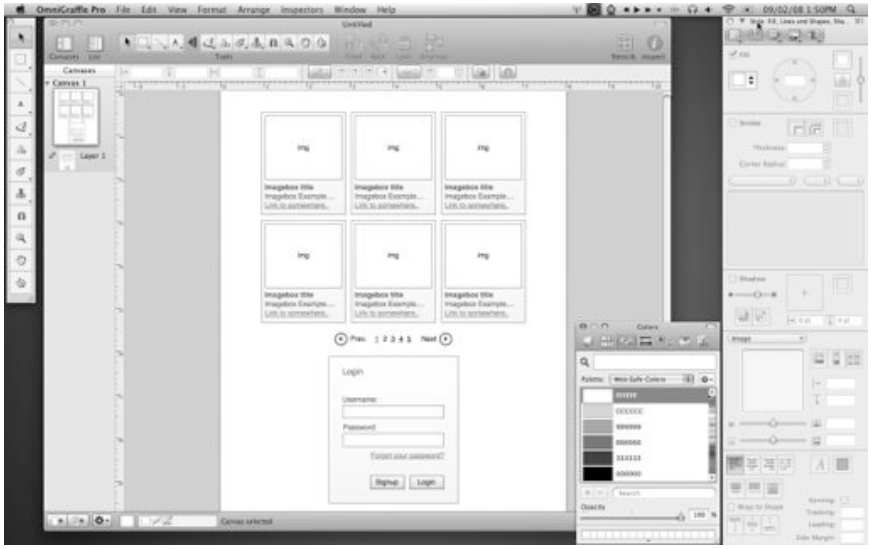
Axure is one of my personal favorite design tools, because it allows you to do many things at once: You can create and document the UI design and also add interactive elements to make a functional prototype (which is rendered in HTML). It's available for both the Mac and the PC, and is fairly easy to use and learn. It's not cheap, but for what you get from the tool it's worth it.

Adobe Creative Suite

My “go-to” design tool, the Adobe Creative Suite is a “must-have” tool for design professionals for a reason – it lets you do everything, from print and web design to interactive and sound editing. If it only had Photoshop, Illustrator and InDesign... it would be indispensable. It’s a lot more than that. Depending on what “set” you buy, it’s either expensive or VERY expensive, but it’s worth it.

(Adobe has recently gone to a “subscription” model – which I don’t like. I recommend buying the packaged software, which should still be available through services like Amazon, etc.)

Omnigraffle



A mac-only design tool, Omnigraffle allows you to design user flows and journey maps as well as user interfaces. The ability to add interactivity is limited, and only when you export the design to a PDF. It, too, is pricey but incredibly versatile.

Balsamiq Mockups

Balsamiq Mockups has got a lot of fans in the UX community, and rightly so: it’s a simple easy to use design tool that allows you to

quickly mockup wireframes and, because the default design looks like a sketch, stakeholders don't get fixated with visual design elements. It's also inexpensive and well worth trying out.

PowerPoint and Keynote

Yes, you can create interactive prototypes in presentation programs such as PowerPoint and Keynote. It won't be as engaging or as interactive as some of the prototypes you can create in Axure but if you are designing a simple kiosk-like UI then these tools can definitely help you build out something to test with.

iRise

An extensive suite of functionality in iRise allows you to make a very interactive prototype. That deep feature set comes with a price, however: A steep learning curve and a high price. I've worked a little bit with the tool, and it's not a bad product... it's just too pricey for me.

Expression Blend

Now a part of Visual Studio 2012, Expression Blend is a great tool for mocking up screens if you are designing Windows Apps. It also allows you to work with versioning and developers to execute that design. It's got a learning curve, but if you have experience developing it may be the right solution for you.

There you have it, some of the many design tools available for you as a user experience professional. As I have said before, there is no "right" tool... use the tools that work for you and best aligns with your needs. And no tool will make you a better designer – only talent, practice, and time will do that.

Best practices in UI design

When it comes to designing user interfaces, there are a lot of factors to consider – the standards and conventions of the medium you are designing in, the usage patterns of the features you are implementing, and the goals and needs of the user. While this section cannot be a "paint by numbers" guide on how to properly design a user interface, hopefully it will detail some practical guidance on how to do the best job you can.

Consuming vs. doing

First, know what type of interface you are designing. There are two primary ways users engage with an interface. Either the user is consuming information (even if it is a single data point) or the user is doing something (typing, drawing, saving, etc.). To state the obvious, they are not the same thing and how you structure the information and the options on your UI will be different depending on this mode of use.

Support this different focus with your design by removing superfluous information and controls when users are consuming information, and place key options and controls in an easy-to scan and use way when users are doing.

When creating a UI that is for consumption, keep the following in mind:

Design for "scanability." When most people read text on a screen they don't read in the same way they would consume a physical book – they tend to scan, looking for keywords and phrases. Support this by keep non-article content short and using bullets.

Break up long paragraphs. It's always a good idea to keep content short and digestible – having a lengthy paragraph results in a "wall of text" effect that discourages reading and intimidates

some users. Try and keep paragraphs less than four sentences, and be succinct.

Place important content "above the fold". "The fold" is the line where the screen is split between the content and functionality that initially displays and the additional information that displays when the users scroll. As screen resolution and sizes have changed over the years, the location of the "fold" has changed, and many designers have debated the importance of this rule in recent years. I still think it's good advice, and worth following.

When creating a UI to support help users accomplish a task ("doing"), keep the following in mind:

Hide secondary controls (and provide them "on demand"). Too many UI designs throw every option available on the primary screen for an activity, even though there are usually only two or three primary controls that are absolutely necessary. Be thoughtful about what controls should be displayed and what are "secondary" to the primary action the user needs to accomplish.

Make sure labels are understandable and clear. Labels and controls should be intuitive, obvious and user friendly. Not doing this will result in user confusion and frustration so it's important to pay close attention to this detail.

Leverage pattern libraries. Pattern libraries are sets of UI elements that are considered "standard" ways of doing things. Different pattern libraries exist for different platforms and operating systems, and these patterns reflect the conventions and approaches that exist in those mediums – standards that users have learned over time. Using these libraries helps you leverage this user understanding and reduce the learning curve for users of the screens you design.

When placing controls, align with users expectations. Users have expectations when they engage a screen, based on their previous use of technology and similar systems. A good example is the placement of submit and cancel controls on web forms – most

users expect to see cancel on the left and submit on the right. Don't "break" from those expectations unless you have a good reason too, and then make sure you test the design to confirm you have made the right decision.

The Yahoo pattern library is a great resource.

Design for diffused attention

As a UX designer whose job is to create screens that users interact with, you need to remember your user base is often distracted, detached and multitasking.

And it's not just the environment that's a distraction. Users are preoccupied and distracted all the time, not just by what are around them, but by what is in their heads. They are worried about their kids, or wondering what to cook for dinner. You can't assume that the UI you have designed is their sole focus. Ever.

So, how do you design for a world where users are distracted and preoccupied? Well, understanding the fact that users ARE distracted and are often not going to be 100% "at attention" is a good place to start. It brings to your mind the context of the situation, and accommodating context and situational understanding is a key to refining a design to support those factors.

Practically speaking, and as good as you may be as a UI designer, distraction will still be there for many many users of the solutions you create. The best you can hope for is that you provide a clean UI that doesn't make things worse. So the key is this: keep things simple, and serve up information and functionality as cleanly and as well as you can.

Finally, a few words on "dashboard" design

Many systems have a "dashboard" that users access to get a sense of how things are going in a particular area. It could be a secure home page for a bank broker or stock trader, or a set of sales or productivity charts ... whatever it is, a dashboard contains vital information that users will need to get at a moments notice. Here are some key best practices to follow when designing a user dashboard:

Location, location, location! As people read left-to-right and top to bottom, prioritize the top left area for the most important information on the dashboard and the bottom right for the least important information. (Of course, if you are designing a dashboard for a culture that doesn't use this Western reading convention, place the information in a way that confirms with how they read information).

Layer information appropriately. Identify the appropriate level of summary information to display, and show only the "tip of the iceberg" on the main dashboard, and allow the users to dive deeper into the details as they want/need to (and, obviously, provide an obvious mechanism to take that deep dive in the UI).

Avoid flashy visualization. If you render information in flashy charts and graphs, it could diffuse the information's meaning and confuse the users. When doing visualizations, make sure they are understandable and align with the user's need to quickly "get" what is being displayed. Also, provide easy and obvious access to see the underlying data that is driving these visualizations. Test these renderings with users as early as you can.

Define different dashboards based on roles and usage. What roles the user has in an organization will influence the information and functionality they need to have on their dashboard – there is usually no "one size fits all" approach. And how often they access a dashboard will influence the design as well – if they use it all the time you can potentially have a steeper learning curve than if they will only look at it once a month (because they will be using the dashboard as a "sovereign" app they will have time to learn by repeated use). To identify what users need on their dashboard and how often they will use it, do the obvious: research users and ask them.

Best practices in form design

Keep things on a “need to know” basis

What information do you need to have the user enter? Let me restate that: What is the LEAST amount of information you need from a user? Keep in mind the different business cases and situations are a huge factor: You’ll need to ask a lot more questions from a user who is opening a bank account online than you would somebody who is buying a DVD from an electronic storefront.

Even still, ask only what is necessary – and don’t add irrelevant questions that could frustrate the customer.

It’s not a paper form

The number one thing is to always keep in mind the obvious point – a web form is NOT a paper form. Users have a lot more patience filling out a paper form than an on-line form, even if the goal of the application is exactly the same. I've done A/B user testing with paper vs. online forms and that lack of patience surfaced time and time again... even if the online form took a lot less time to fill out, people still expressed frustration. So don't make the web form "look" like a paper form.

Split and chunk!

Split out any long form to multiple pages, and clearly label each form. "Chunk" like information (example: if you are asking a lot of personal information, put all those questions on one page and label it "personal information.")

How to do this? Take all the form values you need to capture and group “like” elements through an affinity exercise, using sticky notes. If you can, have two or three different groups do it to provide different perspectives. Any fields that don't naturally "fit" become candidates should be considered candidates for removal.

Tell the user how long it will take (and what you will be asking)

Tell the user before they start filling out the form how long it will take and what the information is used for (both points reduce abandon rates when used). Provide a subnav for a multi-screen form/process.

Identify what the requested information will be used for

One of the first things I would do is get a better understanding of is usage. What is the information entered going to be used for? A good sense of what is important and what is not will quickly come from that. This will help you take your first cut with all the data fields you ask.

What are the business and user goals?

What is the business driver behind the form? Is it to let users signup quickly? Is it to ensure that the right information is being captured? Is it to have as little “abandonment” as possible? Clarity around what the business AND the users wants is key to balance both groups’ needs.

Do you need a form at all?

Twitter, LinkedIn, Facebook, and Google (and many others) have open APIs that allows for “single sign on” – instead of filling out a form to open an account, you simply connect with one of these services. It allows for quick access and removes the need for a conventional “sign-up” form.

Stagger the input

Finally, consider asking only the bare minimum, and "staggering" the user input to ask the information ONLY when it is needed in the full experience. Amazon does a great job with this technique,

baby-stepping users through a potentially complicate process. Another idea: "pre-filling" fields with "intelligent default" values so that people can edit the information if it's wrong instead of entering it from scratch.

Redesigning existing (and potentially long) forms

Often we are not designing a form, we are redesigning one that already exists. The first thing I would do, when looking at that, is question why it was long in the first place. Do you really NEED to ask all the questions that you are presenting? Is there any way you can remove several of the questions to simplify and streamline the form? Sometimes business rules and legal constraints may mean you HAVE to ask certain questions (such as the aforementioned online bank application form) but it's always worth questioning and trying to reduce and simplify.

After you have reviewed and (hopefully) streamlined the form fields mock up the restructured data entry page in a way that you can test it, and take it to users – if you are redesigning a form that a company may be using internally, take the proposed redesign to the people at the company who used the old form. Having them actually USE a new version of a form will prompt a much different and frequently more specific feedback than looking at the information in an abstract way.

Best practices for mobile design

Keep context in mind

Whatever you are designing for a mobile device will be used "on the go" and you need to identify a "context model" that is applicable. I had a good example recently when I was debating a mobile design with a colleague. The colleague had created a very simple stepped approach at data entry which on the surface was fine... but it was designed without thinking of the context of use....when the user would be using the screens, and what their focus would be. This context pointed towards a different design approach.

Align with device conventions

If you are designing an app for an Android phone do not use iPhone like controls, and vice-versa. Users have "learned" the interaction patterns that is used on the device, and forcing a convention different than what they understand will produce confusion and difficulty where it can be avoided.

Test on the device whenever possible

Either create a clickable prototype or have one of the development team code a thin front-end. Testing interactions on a touch screen requires you to actually TOUCH the screen... any other test method will only give you a partial and hollow result. And test "in the wild" if possible.

Don't try and mirror the online experience

There are functions that, frankly, don't make sense to provide in a mobile app. On a recent mobile project we had requirements that mandated that a comprehensive administrative set of functionality should also be available on the mobile app - I'm talking about

account permissions, rules around who gets to use what functionality, you name it.

We interviewed users and found that if we had designed all these screens they would be hardly be used, and that the more appropriate mobile design would be a much smaller subset of what was requested. We COULD have spent a lot of time and energy supporting the initial requirements... but understanding usage patterns and context lead us to a more appropriate mobile solution.

Know your platform

You can't design for the iPhone if you don't know how it works... And how it doesn't. Every platform has its strengths and weaknesses, and you should know what you are working with.

Design for the "Immediacy of Now"

One of the major design projects I worked on a few months back involved redesigning a financial management web application. We created a concept design that used a calendar as the system's primary navigation. Users could add or move financial transactions on the calendar by dragging and dropping, or see projected balances on certain days – the idea being that users could manage their finances through that screen.

The design was very excitedly reviewed by stakeholders, as well as my peers... even our CEO loved it.

Then we tested it with users.

The reaction? Meh.

We soon realized the problem with the design. It wasn't that it was bad – it was cool, interactive and very Web 2.0. The problem was it didn't match the mental model of our users. We were forcing an inappropriate metaphor on our users, and were trying to make "list users" into "calendar users." But, more than that, we misunderstood the context of use.

Our users did not plan – they reacted, usually to inputs that were not available in our system. Giving a holistic calendar view to them did not help – in many ways it reduced clarity and focus... They were overwhelmed. They couldn't react to the UI affectively because there was too much to react too and it didn't "fit" the way they worked.

The system did not support what I call the Immediacy of Now – the need for users to quickly accomplish an urgent task. It may not be urgent to you or me, but it is to them.

Support this behavior by keeping things simple. Don't add needless abstractions or features – support the core task and stay out of the way. And if you are designing mobile applications, reduce the process steps to one or two at most. You'll be quite surprised how well your design is received when you don't over do it.

Best practices for voice interaction design

One of my first jobs in user experience was designing an interactive voice response system for my company's customer support line. it was more challenging that you might think... we had to understand the mental model of users who would call such a number, what their priority needs would be, how much "information" the caller can keep in his or her head at one time, etc. Such systems are still around, obviously, with some far better designed than others (I think the one I had a hand in was pretty good).

Now, however, we have an even greater challenge – how to design true voice interface systems, systems like Apple's Siri that lets users do anything. I've been thinking a while about some "best practices" that can be applied to such interactions. Here's my first swag at some good advice on how to make a good voice interface:

Be conversational

Respond in a way that is personal and polite, not monotonous or robotic. Just because it's a computer doesn't mean it needs to be a computerized voice. Be friendly, but...

Don't get too personal

Avoid words or phrases that are overly critical or praising... It can come off as cloying and phony. While it was design in a different domain, I once evaluated an ATM UI that used phrases that were completely informal... It was off-putting and inappropriate.

Recover gracefully

When mistakes are made, learn from them. Apologize, and remember. Design a learning system. Yes, I know, this is easier said than done, but it's important.

Respond quickly

Latency is DEATH to a voice control system, because the user is expecting the response immediately. Every time the system responds with "thinking" is a moment where the user's confidence is reduced.

Be ready for ANYTHING

Build out "trees" that account for as many eventualities as you can think of. Be thorough. Defaulting to a web search is not a positive outcome. In creating a voice interface system you should aim for a complete solution, and deliver it... and users will use the system with great confidence

Be reliable

A voice response system that doesn't respond... well, that's not a very good thing, is it? Make sure that the system is like the dial tone should be when you pick up the phone – always on, always ready.

Provide alternate inputs, but use voice first

One of the things that Siri does badly is when users has to confirm something. Most of the time the user has to click a "yes" or "OK" button. Why? You're already talking to the device, the device should KEEP LISTENING.

Avoid doing HAL 9000 jokes.

I know it's tempting, but it's played out. But do pay attention to how movies like Moon, 2001, and Star Trek presents voice control systems, because that is the "training" that all of us have. Our expectations are often set by pop culture, setting our mental

models around how we expect technology to work. Understanding the expectations that users have of these systems can help you design them better.

Get out of the way whenever possible

Interacting with a voice system is cool now, but in the future it will become as normal as driving a car or watching TV. Once the novelty wears off, people will be more results and outcome-focused... so don't be cute. Provide an interactive experience that obeys commands and gets out of the way, with no editorial comments.

In closing, I have always said that we are living in the future, it just wasn't the future we were expecting. The very fact that I can write this type of article as a serious piece of guidance to designers instead of a blue-sky puff piece... well, that just proves my earlier point. The future has arrived.

Best practices for gestural interaction and controls

Gestural interaction has been around for a while now. Inspired by movies such as Minority Report, engineers and designers have made the ideas seen in that film a reality for millions of users. From the simplest "shake to play another random track" to complicated game-based gestures, the way we engage with technology has advanced greatly from the traditional keyboard and mouse.

The release of the Leap Motion controller in July 2013 has brought incredibly advanced and sensitive motion control tech into homes for less than $90, and the Xbox One game system promises an additional "leap" in the commercial application of this new way of doing things.

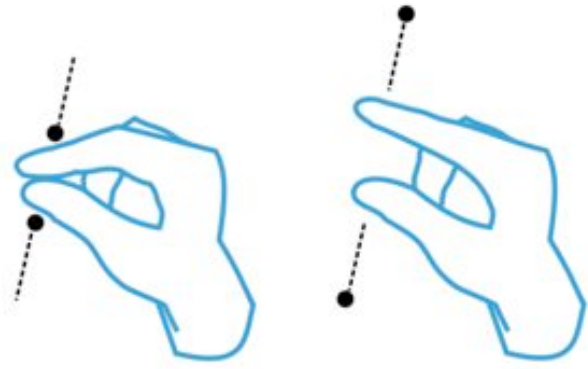
As this "new way of doing" looks to not be going away anytime soon, here's some "best practices" on how to do gestural design for your application.

Decide how and when gesture control should be used... and choose wisely

One of my favorite pieces of movie dialogue is from Jurassic Park, when Malcolm (played by Jeff Goldblum) questions the whole premise of the theme park: "You realized you could do it, you never thought if you should." Applying gestural control is a great example of technology that could be applied... but should it?

Don't just decide to do gesture control because you can, apply and use it in thoughtful ways... Make it align with what you have designed and don't make it a "bolt-on."

Keep gestures simple



I've seen some examples of gesture control that, well... are too cute and complicated to work. People don't want to look foolish, even if/when they are alone. Don't have users doing the "chicken dance" from Arrested Development in order to accomplish a simple task. Be thoughtful and keep things simple and straightforward.

Gestures have to make sense

Don't do a gesture that doesn't align with how people do things in real life. An example: If you are paging through "cards" in a UI, use a swipe to the right to dismiss the top card on the stack. This is like dealing playing cards, and this gesture will align perfectly with how people do things in the real world. If you do the opposite (swiping to the left), it will be unnatural to people and will reduce the "learnability" of the app using such a gesture. Remember: The more natural and obvious the gesture, the easier the application that has these gestures is to use.

Use common gestural "affordances"

Pull, push, swipe, twist: we all use our hands every day to do things with gestures like these. This follows through on the point made above: Leverage how people do things to make the gestures make sense to users. Watch how people use their hands and then reflect that understanding in the gestures you apply in your designs. Don't try and be "creative" when you have the right answer staring you in the face: Align with what people already do, and you will be successful.

People are lazy

I spoke about future UI concepts and gestural computing at a conference several months ago, and as part of that speech I asked everyone to raise their arm above their head and keep it raised until I said so. Within one minute many of the people in the audience were getting fatigues, and for good reason: Most of us are not fit enough to maintain a gesture for more than a few seconds.

In fact, when Tom Cruise was filming his gestural sequences for Minority Report, he was dripping with sweat within an hour of shooting the scenes... and he's in a lot better shape than most people (like me). Good gestural computing design aligns with the ergonomics and limits of its users, and understands that people usually need "restful" moments where they are not gesticulating. I

labeled this section “people are lazy” but that’s not really true – they are human. Understand and align to their human limitations.

Test, Validate, and Iterate

If you can get everything 100% right the first time, congratulations! You are the world's best designer! Odds are, you won't get everything right the first time, and that is why it's important to test your designs with users and iterate based on that feedback.

Usability Testing

Once you have created an interface, you need to test it, and this is known as usability testing. How do you do that? Well, it's easier than you may think.

First, let's define what we mean by the term "usability test." A usability test is a structured session that involves a person engaging with a design and providing feedback on said design to a facilitator. Usability testing allows for an open objective conversation about a design to identify what works and what doesn't. Why perform usability tests? By testing designs early, you can quickly identify potential problems with the interface before it is coded and implemented. This saves a lot of time, money, rework... and potential embarrassment.

I've met several UX professionals that think you need to have a huge usability lab with thousands of dollars worth of equipment, a two-way mirror, specialized software, and more. While I've used (and set up) a couple of usability labs, and there's several benefits from having such a location for user research and testing... you don't need most of that stuff. What do you need? The only things you absolutely need are:

- You
- The Design
- People to test the design with

That's it. You can do usability testing almost anywhere, and you don't even need a computer. Now I know that there's an entire cottage industry out there of people who sell usability-testing equipment, and many of them would prefer I didn't state this. Sorry, but I call it like I see it... though I will spend some time detailing some usability test tools I have liked and used later. While you don't need a lot of James-Bond style gadgets to do a

usability test, you do need to plan things out so you can get the most out of the effort. So, let's talk about how you can plan a usability test.

You can do usability testing (almost) anywhere.

There are five things you have to do beforehand to have an effective usability test session:

- Define test goals
- Formalize the test artifacts
- Define the type of test and the test method
- Identify who (and how many people) to test with
- Write the test protocol

Define test goals

What do you want to get out of the testing? What do you want to find out or understand? Define clear goals to focus on, and then make sure that the test details don't lose sight or "muddle" those goals. If you don't have any clear goals, then focus on getting answers to three key questions:

• Do they "get" it? If they can grasp the purpose and utility of the design, you're in a good position.

• Can they use it? When given tasks to accomplish using the design, can they do it.

• Can they explain it to you? Can they describe what it is and how it would work?

Formalize the test artifacts

Where you are in the design process informs what design artifacts you create and use for usability testing. If you are doing early testing, the paper sketches or "prototypes" are fine. You may need to sketch out different "screen states" for some screens to represent the process you need to test, but that depends on how you are testing and what your goals are.

If you are testing more "mature" designs you are going to want to make the test artifacts more interactive, so the test participant can engage with the design more. Creating an interactive prototype is fairly straightforward if you can code, or have access to a team member who can develop it for you. If you don't have those options, tools like Axure RP, Omnigraffle or iRise can help you make clickable prototypes (we went over these earlier).

Define the type of test and testing method

There are generally three types of usability tests: Formative, Summative and A-B Testing.

Formative tests are used early in the design process to assess the effectiveness and usability of a preliminary design, as well as to capture users' thought processes and conceptual understanding. It

can be done with sketches or more “formalized” designs, and can be task-driven or an open-ended conversation.

Summative tests evaluate more detailed designs to determine the satisfaction, effectiveness, and overall usability. It usually takes place after an earlier formative test has occurred and is usually very structured and task-driven. You can perform these types of tests with existing systems to evaluate the current state of a design.

A-B testing compares two or more products or designs to identify and distinguishes the strengths and weaknesses of each. If you have two differing approaches to solve a design problem, this is a good way to evaluate them. This can be done at any time, and can also be used as a competitive analysis tool (test two existing sites with the same tasks to identify what site better supports the user). What you will want to do is have two sets of participants, with one set testing design A and the other set testing design B.

Now that you have identified the type of test you want to do, how do you want to execute the test? Again, there are three different ways you can do it.

In-person, facilitated tests provides the opportunity to gather the most comprehensive feedback from participants about a design. It allows for any type of testing, and provides the most flexibility of what and how you test. In my experience, I’ve found it’s the best method to test designs for mobile application

An in-person facilitated test takes about an hour and you’ll expect to pay participants at least $100 for their time (usually with a VISA or Amex gift card).

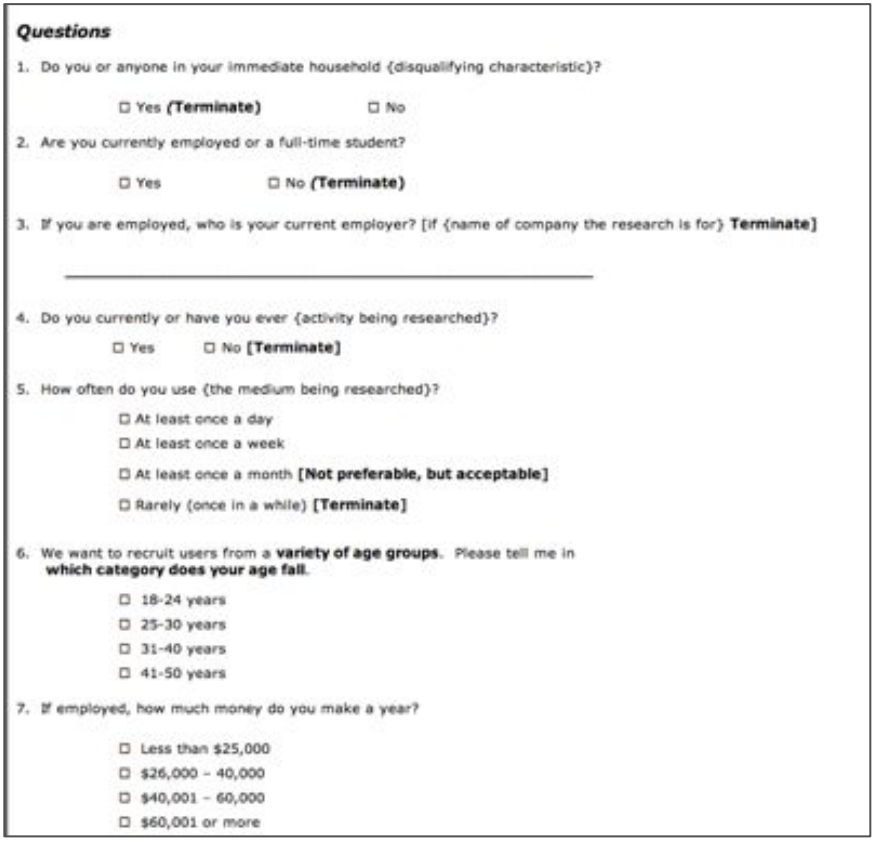
Remote usability testing allows you to get feedback on designs without having to facilitate any sessions. It will cost less than in-person testing, in that the services charge a flat fee of less than $50 per participant. However, it doesn't provide for extensive conversations or in-depth testing of designs, and the results can be hit=or-miss. In my opinion, this method is good for getting

feedback on early concepts and if you need really quick user feedback.

If you're strapped for cash or time, consider in-person friends and family testing, AKA "guerrilla" testing. While not as useful as formal in-person testing, it's still a good way to get "casual" feedback or do some simple A/B testing. It has costs besides people's time (and maybe some snacks or a lunch).

Identify who (and how many people) to test with

Who do you test with? People who reflect the key characteristics of your personas, of course. Use the information that fleshes out these personas to inform the creation of a recruiting screener. Be sure to have some "disqualifying" questions in this screener to prevent the wrong type of people to get in the test group (an obvious disqualifying question: (Do you work for a competitor?")



Questions

1. Do you or anyone in your immediate household {disqualifying characteristic}?

   ☐ Yes **(Terminate)** ☐ No

2. Are you currently employed or a full-time student?

   ☐ Yes ☐ No **(Terminate)**

3. If you are employed, who is your current employer? [if {name of company the research is for} **Terminate]**

   _____

4. Do you currently or have you ever {activity being researched}?

   ☐ Yes ☐ No **[Terminate]**

5. How often do you use {the medium being researched}?

   ☐ At least once a day
   ☐ At least once a week
   ☐ At least once a month **[Not preferable, but acceptable]**
   ☐ Rarely (once in a while) **[Terminate]**

6. We want to recruit users from a **variety of age groups**. Please tell me in **which category does your age fall**.

   ☐ 18-24 years
   ☐ 25-30 years
   ☐ 31-40 years
   ☐ 41-50 years

7. If employed, how much money do you make a year?

   ☐ Less than $25,000
   ☐ $26,000 – 40,000
   ☐ $40,001 – 60,000
   ☐ $60,001 or more

A typical screener

When you test a design, you need to get a lot of different people to look at it. The question, of course, is how many is "a lot"? Well, it depends on the type of test you are doing. If it's an early test and you want to get high-level feedback, you may decide to have a larger group of people. If it's a task-driven test that is focused on identifying potential design or usability issues, then you can "get away" with only five people.

Why five people? Because the law of diminishing returns applies. Jakob Nielsen, after his company spent years doing formal usability tests, looked at the data and identified that after the fourth person you will (on average) identify over 80% of the usability issues in the design tested. With 5 people, that percentage goes up

to over 95%. So, unless you REALLY want to test with lots of users, five is enough (details here).

When it comes to recruiting participants, there are usually local companies that do market research who can find people for you. Expect to pay $100 “finders fee” for each person (not counting the compensation you will need to pay participants for their time).

Writing the test protocol

When writing the test protocol, you need to include the goals you have defined earlier as well as the list of materials/designs you are going to review. A lot of people like to script out everything – the questions, the introduction, the whole thing. I leave this to you, but the key is to present the design artifacts in a consistent way to all participants.

If you defined user scenarios to inform your designs (through either a narrative story or a journey map) then you will already have a list of tasks to use in your test, making this job easier. Using these scenarios in a usability test will also allow you to “sanity-check” these design artifacts to make sure that you have correctly understood user needs and intent – if many of your test participants say “I’d never do that” in response to the task they are given, you may need to rethink some previous decisions.

Closing

A decision you need to make at some point is whether your or some one on your team will do the testing or whether you should “farm it out” to an independent UX consultancy. The answer to that depends on some key questions you should ask yourself. Can you be objective? Do you have the proper skills in-house? Do you have confidence in your team’s ability to do the testing? Consider doing both – maybe you do formative tests internally and then supplement that with an outside consultant to do the summative testing.

Now that you have fleshed out your plan, identified what you are testing, defined participants, and have anything ready, it's time to test. Which is what we'll cover next...

How to facilitate a usability test

When you are facilitating a usability test, there are many different factors that you should keep in mind. These techniques will help you get the most out of your test and will prevent most awkward uncomfortable moments. Here are those tips.

Relax

A lot of people who are facilitating a usability test feel they are “on stage” and they have to get everything perfect. Stop worrying. We are all human beings, and no matter how practiced and focused you may be, something will inevitably go wrong - either you will misspeak, or technical problems will occur. Roll with it.

It’s not about you

I’ve observed many usability tests where the facilitator does most of the talking, and these (novice) facilitators ended up providing a walkthrough of the design instead of a task-driven test of the ideas that the design reflects. This is the wrong approach - the facilitator should not be the focus of attention and discussion, the participant (and their reactions to the design) should be. Ask questions, but listen more than you talk.

Follow your script, but don’t read it out loud

I’ve watched people new to UX look at the test script as “sacred scrolls,” and they follow it to the letter. This has lead to awkward situations, when the facilitator ends up talking to the piece of paper instead of to the participant. The key is to learn and be comfortable with the test and the planned scenarios, but don’t read it verbatim... reading the script will get in the way of you from having a real interactive conversation with the participant. Those type of conversations often lead to interesting topics and insights that are usually “off script.”

Use Active Listening techniques

I first heard about active listening years ago, when I was a practicing journalist, and I think applying those techniques will absolutely help any usability test go smoothly. Among the many ideas discussed in Active Listening courses are:

Pay attention. Read body language to get a better sense of the person's reaction.

Focus on the speaker. Show you are listening by nodding and leaning forward.

Let people finish. Don't interrupt, let them finish their thought.

Don't take notes. It prevents you from engaging fully in the conversation (Have a notetaker nearby or observing remotely)

More details on Active Listening are here: http://www.mindtools.com/CommSkll/ActiveListening.htm

Reassure participants they are not being judged

One of the things I always make certain to do is to emphasize at the beginning of the test that the participants are not being judged or tested - what is being judged and tested is the design. This helps to reduce any possible stress they will be feeling (especially if they had never done any usability testing before - and most people haven't).

This will also prevent them from being self-conscious about their feedback, and when they don't understand something they'll be more comfortable saying as such.

Remind them that you aren't the designer

Let the participant know that you did not design what is being tested, so that they won't hold back negative feedback - many

people are uncomfortable expressing criticism to someone they think did the work, but will freely criticize the work to someone else. By emphasizing that you are not the creator of the work, people will be more open and provide more feedback.

Keep things on track

Be mindful of the clock and try and keep the conversation on track. I always like to do a "dry run" of a usability test protocol with a colleague, to make sure I have allocated enough time for the topics and designs that are being covered. This way you can adjust the test protocol before the first test and get a sense of "timing". That being said, people are different, and because some people being more talkative than others... you may have to actively facilitate some tests more than others.

Don't ask yes or no questions

If you ask yes or no questions, you will get... yes or no answers. You will gain no insight or understanding as to what you are covering. Ask "Why?" questions instead, to open up the topic to detailed conversations and understanding.

Be grateful

Even though you are usually compensating test participants, thank them for their time - they had to change their schedule to make time for your usability test, and you should show gratitude. Don't overdo it, though... you don't want to make the participant think every piece of feedback is wisdom plucked from Mount Olympus. Thank them for their time and their feedback, but don't gush.

Don't flirt

This is an obvious point, but I have to mention it. If you find the participant attractive, do not act on that response. Be objective and don't treat the usability test as a blind date (I've actually seen facilitators aggressively flirt in sessions, and it's inappropriate and distracts from the task at hand).

Don’t beat yourself up when a test “goes south”

About every ten participants I encounter are what I call a “dud.” You either don’t get any additional insights, the participant lied to get into the test and are only interested in the compensation, or the person expresses opinions that are so “out there” they are a clear outlier. It happens. Don’t fret or get angry - just understand that the nature of usability testing will expose you to all types of people and some of them will not add a lot of value to your usability testing.

There you have it, some techniques that will help you run better more effective usability tests. The key is to start applying these (and other) techniques and START TESTING. It will make the work your team has done better, and will in turn make lives better for users.

Best practices in mobile usability testing

Recruit people who use the device, not just people who “own one”

Sometimes you will encounter a participant who got through the screening process and has NO IDEA how to do anything with his mobile device other than make or receive calls. You won’t get a lot of valuable data from this person, because a

good half of your time with them will be spent explaining how to do things with his phone. Make sure your screener has questions to prevent these type of recruits from showing up.

Understand usage and context

In the pretest conversation, get a sense of how people actually use their mobile devices. Asking them what they do with their phones allows you to get a better sense of the priorities and features the user needs and provides valuable context that can inform your design goals.

Capture Everything

I’m a big fan of Morae from TechSmith, an app designed to record and analyze usability tests, and the latest version supports two video sources. The setup is simple: Use two cameras, one that is on a mounted tripod pointed down at the device, and the second monitoring the participant. Morae captures both video signals and records them, as well as transmitting the video of the session to a note taker via your work network.

Get comfortable with the devices you are testing

Even if you are recruiting users that are familiar with and/or are "power users" of a particular type of phone, you need to spend time getting used to how the device works. You may encounter a person who is not accustomed to the eccentricities of how the phone operates in the space you are testing (for example, a downloadable app on a blackberry). Be ready to work through such bumps by knowing how to do

When possible, test no more than two devices in one round of testing

If you are going from one device to another, you tend to lose focus on what you should be doing which is observing how the participants uses the app you are testing – you will focus more on how the device works than you should. So, test two device types in each usability test round, and try and have all the participants of Device A test on Day one and all of Device B test on Day 2.

Remember to separate the device's usability issues from the application

On some phones you are going to see some very painful interactions that the participant has to go through to accomplish the task you have laid out for him or her. More often than not that is caused by the awkward data entry mechanism that the phone has and NOT the app itself.

Always remove the usability problems the device causes from issues observed with what you are testing. If you'd like to catalog those issues and send them to LG or Nokia, fine, but that's NOT your priority.

Make sure what you are testing supports the phone's native controls

A couple of show-stoppers came up in a previous test when the participant used the native buttons/controls while accessing the app (the keypad's "Back" button was one example). The app was not

able to support this and logged the users out. If you have at least one participant do this, odds are that many more will being doing this as well. Make sure to note this as a MAJOR issue that the dev team needs to fix.

Getting the prototypes on the phone

You can use PDF readers to open PDF prototypes. For iPhone, I've used Keynotopia and for Android I've used a couple of different apps (whose name escapes me right now). Some PDF readers for Android don't "pick up" the jump to links in the PDF, so be sure to test the prototype on the device before bringing in users. You can either copy the files to the phone or if that is not possible you can access the files through Dropbox on the device.

Designing on the device itself

A couple of apps have come out over the past few years that allow people to design or test paper sketches on the device itself. One iPad app called Blueprint allows you to design iPad and iPhone apps and export the results to a special reader/viewer. Another app called POP allows you to photograph sketches and then embed links that users can use to jump to different pages and views.

Holding the device

I've used – and like – Mr. Tappy (http://www.mrtappy.com). It's a very solid metal handle for mobile and tablet testing that holds and elevates a webcam above the device (you can use software like Morae to "flip" the image if you need to). It's pricey (at $295) but it's good.

Tools of the Trade

Here's a list of the most popular usability testing software programs:

Morae

My favorite tool for usability testing, Morae allows you to capture the desktop of the computer and an additional video source (either the built-in webcam or an external camera) and the tool will transmit the video and audio of the test session in almost-real- time to another computer on the local area network.

The software allows for remote note taking and "tagging", so that you can have a log of perceived usability issues to analyze later (with some great built-in analysis tools). It's a great suite from TechSmith, and a great alternative to an expensive usability lab.

Silverback

This is the OS X equivalent of Morae, sans most of its features. It allows you to record the webcam and desktop from a computer and that's about it. Please note it is also MUCH cheaper than Morae, so you get the core functionality both share at a lower price.

Usabilla

Usabilla is an on-line "remote" usability testing tool that allows you to have three active tests at a time. The $89 a month fee does not include participant recruiting and incentives, so you'll need to pay those additional costs. It's a fairly simple service, but it may align with your needs better than...

UserTesting.com

UserTesting.com costs $39 per participant, and that cost includes recruiting AND compensation, so it's a flat rate for everything. The drawbacks are you will have to host the design you are testing on your own server and they only offer limited screening capability. Once the participant tests your designs, you will have access to the video of their feedback to review afterwards. It's the best solution for the "budget-minded" design team.

Loop11

If cost is no object, then Loop11 is the best remote testing solution for you. It costs $350 per project, or $9,900 for an unlimited license. It offers much more interactive and richer user testing, and allows you to host your designs on their servers. Be aware that, like Usabilla, you will have to pay for your own recruiting and participant compensation.

Ethnio

Ethnio is an online recruiting tool that pays participants Amazon Gift Cards. The service costs $49 a month and allows you to post a link to screening questionnaire from Twitter or other social media sites. You can recruit up to 250 participants a month for the one flat rate.

The best thing about the service is you can integrate with Usabilla, Optimal or UserTesting.com to redirect participants directly to the tests you are housing through those services. It's a great alternative to paying an outside agency for participant recruiting.

Pulling Everything Together

Now, it's about execution... finishing the project and promoting UX in your organization and how to lead a team of designers...

Producing Design Documentation

How much design documentation do you need to produce? Do you make pixel-perfect wireframes that detail every single interaction and behavior? Or do you create mid-level documentation and leave the details to the development team? The answer, of course... depends.

I know many designers who feel they HAVE to flesh out every single aspect of the experience, and so they spend hours and hours making sure everything is spelled out and documented. They produce hundreds of pages of design specifications, which are then handed over to the development team to execute from (and the QA team to test with). This reflects the classic "don't make them think" attitude that some UX professionals have towards developers – and, to be fair, many developers have the same attitude. This extreme level of specificity, though... I think its overkill. It ends up increasing the design lifecycle as well as increasing development and QA time.

On the opposite end of the spectrum, I know some designers who do a cursory job of design documentation and do only the bare minimum. It's not that they are lazy, they just don't like doing design documentation (full disclosure: I'm not a fan of design documentation, either). When this design documentation is delivered, developers often have many questions about use cases or exception handling that is not detailed out. Just like documentation overkill has issues, "underkill" like this produces lots of problems.

To me, the best design documentation is like the best design: It aligns to the needs of the users and solves problems for them. For developers and QA team members, this means getting to know them, understanding their process, and aligning with what they need. This means having a certain degree of flexibility about how

you do things (and there may be some negotiation). Basically, document the design as much as you need to in order support your team and ensure the vision of the design is properly executed.

Additionally, there are some design documentation "best practices" I have identified over the years, and hopefully these will help you in your work:

Document standards instead of only design documentation

I'm big into efficiency, and one approach that helps you be productive is to define design standards that should be followed throughout what is being designed. These overarching rules can be used to answer any questions and resolve any ambiguity. Try and

make these standards specific enough that they aren't platitudes, and are practical and actionable. Look up examples from Google, Microsoft, Amazon, and more to help come up with your standards.

Build (or leverage) a pattern library

Once of the great principles of development is the goal of "write once, use everywhere." This means coding or leveraging standard libraries that save time and effort. You can do the same thing with design documentation by leveraging existing pattern libraries and/or creating your own. This way, you will be documenting a UI element once... instead of every time it appears on a screen.

Produce design documentation that is used and useful

Don't create design documentation that you think is what is needed, based on what you have done before... Make documentation that aligns with what the team will actually use. Don't spend hours building out a complicated sitemap that will look really nice framed on the wall that no one will ever apply. Make documentation that matters, and apply user-centered design

techniques to ensure that what you produce is usable for the recipients.

Don’t replace communication with documentation

Don’t throw design documentation “over the wall” and hope that documentation will be enough – walk through the design with the team and explain how it is supposed to work. If you are in an Agile team, this is a constant process and it results in a better design. If you are doing “waterfall”, you will have to set aside the time to do these walkthroughs... Do it as soon as you can, to understand both the level of documentation needed as well as to get feedback on the design that can you can apply to make the work better.

Focus on design, not on documentation

While you should always put aside and plan for sufficient time to document what you have designed, understand that your focus and most of your time should be sped designing solutions and focusing on user needs. Spending all your time documenting instead of designing will result in a false sense of accomplishment – you may have a big pile of documents at the end of the day, but if the design spelled out in those documents don’t help people accomplish their tasks and do things faster and better... well, it’s a hollow victory.

Bringing UX to your organization

You're passionate about UX, you've "drank the Kool-aid" and you know how much value user experience design can bring to your company. Now what? How do you promote UX to decision makers and stakeholders? here's some thoughts and advice as to how to work within your organization to get UX more attraction and interest.

Know your organization's "UX Maturity"

Johnny Holland published a great "UX Maturity" chart, that measures the levels of understanding and application of UX in an organization. It goes from Level 1 ("Unrecognized") to Level 6 ("Embedded"). Knowing where your company is in this continuum will help you know the level of discourse and education you need to have, so you can speak to the value of UX at the proper level to the right people.

Nothing succeeds like success

The easiest way to get attention to UX is to be successful applying UX to a project. After you succeed, you then have a story to tell people and this case study can help increase interest in applying more UX principles. If you are not in a position to apply UX to a paying project, do it on an internal effort - solve some process problems or improve deliverables. Even if you can't make a big impact, the key is to build your case and have a story to tell - as noted earlier, we are hard-wired to respond to stories and this makes things easier.

Don't try and do it all

Take baby steps, and use only one or two UX techniques to solve problems. This way you can be more focused and won't be

"swinging for the fences" on your first at bat. Build your skills, and successes, one step at a time.

Understand how things are done today

Spend some time learning current processes to get a good sense of potential "insertion points" where UX can add value or save time. This will show management you know what you are talking about, and therefore increase your credibility when you propose changes.

Get an executive "cheerleader"

If you have the buy-in of an executive, that is worth its weight in gold. Executive support can get you funding, break down barriers, and help you in many ways. I have seen just how important such support is first-hand... when I had it, and when I didn't. Reach out to an executive to make your case and get that support if you can.

Apply UX in the early stages of a project

If you are working on projects, bring UX practices into the early part of projects instead of later. When you are in a project's early days, the tolerance for failure is higher and deadlines are (usually) not as aggressive... so if the UX practice you apply doesn't work, there is less chance the application of same will blow up in your face and give UX a bad reputation to management.

Don't be discouraged

Organizations large and small are established in their ways, and many organizations are reluctant to change. When you try and "rock the boat" and do things differently, you will see first-hand that hesitancy. Sometimes you will think you are banging your head against a brick wall... don't let it get to you.

Keep trying to make UX a priority at your company, and be positive as much as you can. Positive attitudes are attractive to people, and this will help your cause.

A few words about Accessibility

When you are designing, you should strive to make your work usable to as many people as possible - and that includes people who may have disabilities. Doing this means you make your site or application “accessible.”

When you create an “accessible” UI, you are designing and developing a screen that can be read and accessed using assistive technology (screen readers, high-contrast Screens, etc.) How accessible the application or site you and your team will be producing depends on a number of factors - the project timelines, the medium the solution is being designed in, and so on.

It’s a good idea to produce an accessible solution not just because it’s the “right” thing to do - it also helps mitigate risk. Several sites have been sued in the past because they did not provide an alternate accessible version of their site. And if you are developing a program for the government, you have to strive for 100% access – it’s the law.

A lot of the accessibility effort will be technical - developing the screens in a certain way. But quite a lot of it is also about how the screens are designed. Some examples (taken from Section 508 Guidelines and W3C Standards):

- Web pages shall be designed so that all information conveyed with color is also available without color, (e.g., from context or markup).
- Avoid overly complicated data tables - Complicated data tables are difficult for users to follow.
- DO NOT use tables for layout – use style sheets.

- When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

There are a lot more standards, but thankfully modern operating systems (both mobile and desktop) have "baked in" accessibility tools that both users and developers can leverage.
While this book lists accessibility "at the end", designing accessible solutions should be something that shouldn't be put off until the last minute. Users with disabilities need to be supported and accommodated for, and it's an important matter that should be a priority.

The Next Level: Managing a UX Team

If you apply all you have learned from this book (as well as expanded your abilities by reading other great resources in the domain) and have success, you may end up leading a UX design team.

I've managed a few teams in my time (cue crodgety old man music), and so I have some insights that can help your design team run (semi) smoothly. Some of these recommendations are applicable to ALL types of teams, and I list them for the sake of completeness.

Have a (realistic) shared vision and direction

Have a mission statement and a set of principles the team should follow, but make sure its pragmatic and achievable. If you pump up your team with lofty aspirations and blue-sky visions they will quickly become demotivated when the hard fist of reality shatters those dreams.

Set a rigid, yet flexible, process

Set a standard design process, based on best practices, so that junior designers can have a guideline to do their jobs and work from. But be flexible about the details of the process if team members come up with better ideas on how to do things.... keeping in mind that "flexible" is not the same thing as "disposable" - process is important, and a repeatable process will increase efficiencies over time.

Spread the love around

Don't play favorites; let everyone have a shot at the big projects. Even if you don't think a member of the team has the ability to do

the job well, you have to give them a chance to step up. Just pay close attention and help them when they need it.

Listen and pay attention

You need to be there for the team, and a lot of that involves listening to a lot of complaints and gossip. Encourage and support, be ready to intervene if your team members need help. And try not to "gossip back" if you can help it.

Allow for disagreements, but don't let them get personal

Design teams are passionate, and they will often wear those passions on their sleeves. Let them debate, but don't let it get out of hand (and get in the way of work)... and don't let people insult each other or their abilities.

"You are not the work."

This is one of the most important points I can make, which is that designers have to separate themselves from what they have done, because what they have designed will change before it's finally implemented... and you may find through usability tests that the ideas/designs don't work at all. Continuously reinforce that point to designers in your team.

Don't compete, mentor

As a leader of a design team you should never put yourself in a situation where you are "competing" with anyone on your team. When I say that, I mean don't be working as a designer and a manager at the same time, because that could cause jealousy and discord. If you have to step in and work, do it by partnering with a junior team member and use it as a mentoring opportunity as well.

Set up "peer designing"

If you have the ability, partner designers on projects. Having two or more people working together allows for the generation of more ideas, the ability for one designer to support or encourage the other when he/she needs it, and allows for "cross-training" while the work is getting done.

Credit the team, not yourself, for successes

If a design project goes well, give full credit to the people who did it. And if things go badly, don't throw anyone under the bus. You are where the buck stops.

Don't play favors (and don't hire friends)

Don't spend all your time with just one member of your team, be equitable with how you invest that time with everyone. if you don't, people will be left out and they may not give their all when you need it from them.

It's very tempting to build a team with colleagues you worked with before, because they are a known quantity and you enjoyed working with them before. Don't do it. In my experience, brining in friends to work for you will end up resulting in bad things happening. Because they have worked with you as a peer they'll not respect you as much when you are the boss.

Appreciate that people are different

You are going to have a mix of different skills on your team, and not everyone is going to be the "perfect designer" that you can just throw at a project and let them go. Pay attention and balance out what people do with what they are good at and the most passionate about whenever possible.

Mentor, don't command

You cannot mandate design direction to your team. They are going to have their own ideas; they are individuals with their own perceptions of the problem space. You may "know" the right

approach is to do A, but sometimes the best thing you can do is let the team decide that A is the way to do it on their own (with a little coaching form you).

Allow for regular personal "research time"

Unless you are under a crushing deadline which requires "all hands on deck," give your team time to become better designers and researchers by reading articles and books. Make continuing education a necessary part of everyones time. And allow for time for the team to share what they learned with each other (preferably over drinks at a social hour).

Have fun

Captain Kirk once said that the more complex the mind the more important the necessity of play. User Experience Design teams are usually made up of some very smart "complex minds" indeed - always allow for opportunities to explore and play.

Letting go and finishing

As important as iteration is when it comes to UX design, it's also important to know when to stop. Too many times I have worked with designers who didn't want to finish, who tweaked and changed and refined so much that (sometimes) the work had to be pulled away from them to be completed.

This is not the way to do it, for many reasons. First off, it makes you look indecisive and insecure... which does the whole UX discipline no favors, either. Second, it takes time away from the developers who have to execute your design, which is unfair to them.

And finally, any delay caused by excessive iteration endangers the project and could prevent ANY solution from reaching end users. The whole point of user experience design is to create solutions for people... to make their lives better. If nothing is ever "shipped", then no one will get any benefit from what you have designed.

(This final point is especially important to me. I have worked on many projects in my time, and sometimes the project was halted before it was completed due to business decisions or budgetary reasons. Yes, the designs look good in my portfolio, but I'd rather the work benefit other people than just sit there unused.)

There's a lot of reasons that some designers are like this... here's some of them.

"Mr. Blue Sky"

Some of this excess iteration is driven by excessive optimism, where the designer is incredibly aspirational and wants to "change the world." While I'm an incredibly optimistic person, I also have some perspective. The world is a very big place, and I've been involved in some design projects that did, indeed, make a big

impact in the lives of a lot of people... but those are few and far between. We need to be realistic and temper our aspirations.

Designing for the next big thing

Lots of UX designers are, like me, fans of technology. Some of them are such fans that their designs aren't feasible... yet. They design applications that would only work with "future tech" - advances that are being worked on in the lap, but are still months or years away from being released in the marketplace. This is not necessarily a bad thing, because we all need to dream big sometimes... but it's not very practical if you are on a deadline-driven project. Doing this too often could also result in a reputation as having a "head in the clouds" and may impact opportunities and your career.

"Great artists ship."

This quote from Steve Jobs is one of my favorites, because it combines art and commerce in the most simple elegant way. User experience designers need to heed those worlds, and focus on results and outcomes rather than process.

While I spent a lot of time in UX101 talking about process, I also tried to emphasize outcomes and deliverables as well. You have to finish, so that your visions and designs can become reality. Endless iteration means that you will never produce a final product, and that benefits no one.

"Perfect is the enemy of good"

This is another favorite quote of mine, an aphorism attributed to Voltaire. I have seen designers who iterate too much because they want everything to be perfect. Well, perfection will never be achieved, no matter how many times you tweak the details. Starting is hard... but for people with this attitude, finishing is even harder.

Conclusion

I'm sure if I spent more time writing I'd come up with lots more words to detail out the user experience domain... content that would (hopefully) explain more advanced techniques and information. But that would be #UX201, wouldn't it? The point of this work is to provide foundational understanding and advice about UX to people who are new or interested in the domain. I think I've done that, and so I'm following my own advice: I'm letting go, and finishing.

(In fact, you can replace the words "user experience design" in the above section with "writing", "painting", "filmmaking" or any other creative endeavor, and most of the thoughts would still be appropriate and apt.)

This work was a labor of love - I don't expect to get rich off of it, and in fact will be giving it away as much as selling it. The reason I did it, my "mission statement," was to educate and inform, to hopefully get more people into the UX discipline... More practitioners mean more ideas, and more ideas means better ideas. Ideas that can raise the bar, making the quality level of the software and hardware we use better... resulting better experiences for all.

Will this happen? As noted above, I'm a practical man and I know my place in a (very big) world. Any impact will be minimal... but if this work helps even one person get into UX, or get better at their craft, then it will have been worth it.

If you have read this far, thanks! It's up to you, now. Take these ideas, use them, and hone your skills. Design that cool UI or application that will make people's lives better.

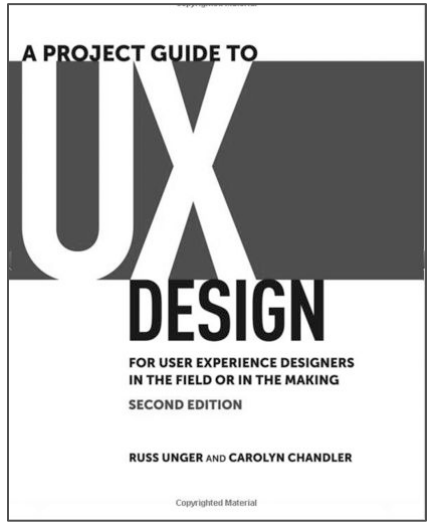
In the words of one of my favorite starship captains: Make it so.

Recommended Reading

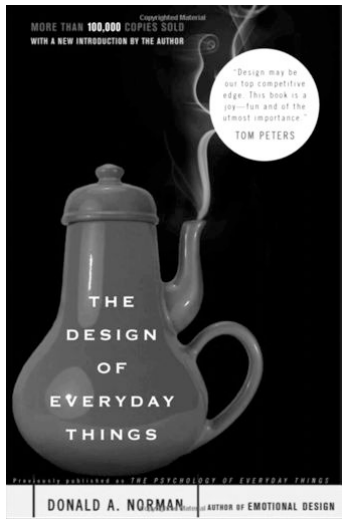
We have walked through processes, best practices, tools and techniques that will (hopefully) get you to the next level of understanding about UX design. Now, it's up to you.

Here are my "Top 10" books on UX design, resources that will give you a well-rounded and deeper understanding of the domain. In addition to these, obviously extend your reading to areas that you are interested in and topics that you want to do a deeper dive into (such as cognitive psychology, mobile design, ethnography, etc.)

Designing for the Digital Age: How to Create Human-Centered Products and Services by Kim Goodwin is a favorite of mine, in the fact that it is just so comprehensive. It's THE book to me.



A Project Guide to UX Design: For user experience designers in the field or in the making is a great primer on UX and a good book around project planning.



The Design of Everyday Things by Don Norman is the seminal work on UX design and should be the first book in any library.

The Persona Lifecycle: Keeping People in Mind Throughout Product Design (Interactive Technologies) is THE definitive reference to creating personas.

Don't Make Me Think: A Common Sense Approach to Web Usability is another "classic" in the domain.

About Face 3: The Essentials of Interaction Design is Alan Cooper's magnum opus on interaction design.

The Elements of User Experience: User-Centered Design for the Web and Beyond by Jesse James Garrett because it's awesome.

Sketching User Experiences: Getting the Design Right and the Right Design by Bill Buxton, because it's really good and practical.

Mental Models: Aligning Design Strategy with Human Behavior by Indi Young because it's a great foundational book about understanding how users think of certain domains.

Inspired: How To Create Products Customers Love by Marty Cagan, because desirability is something that is not covered by most books on UX.

About the author

Joseph Dickerson is a writer, technologist and User Experience Consultant who has lead multiple research and design projects for several Fortune 500 companies. He currently works as a UX Lead at Microsoft.

Dickerson authored an earlier book on UX, Experience Matters, as well as many other books and articles. He is a regular contributor to UX Magazine (http://www.uxmag.com) and editor of This Week in UX (http://www.thisweekinux.com).

He writes about technology, design and pop culture at his website http://www.josephdickerson.com.