User Experience Design

HTML/CSS

Lesson 10: User Experience Design

Introduction

As our technologies advance so do the complexity of the applications we use. Web applications and websites, which in the past have been quite static and bland, have evolved into rich interactive experiences.

However, no matter how the applications have changed over time, a website's success still depends largely on how it is perceived by the user. Whether the user realises it or not, they ask themselves the following questions when encountering a website:

- **Does this website solve my problem?**
- **Do I find it easy to use?**
- **Do I find it pleasant to use?**

These are the questions on the minds of users as they interact with our products, and they form the basis of their decisions on whether to become regular users.

If your website is a visually noisy, busy, slow, confusing, displays incorrect or missing data, then it will result in a poor user experience for your users and they will almost certainly go elsewhere to buy their flights, check the news, find a date or watch videos. In this lesson we will explore the principles and techniques that work together to determine a user's perception of a site and also the characteristics of these users.

Learning Outcomes

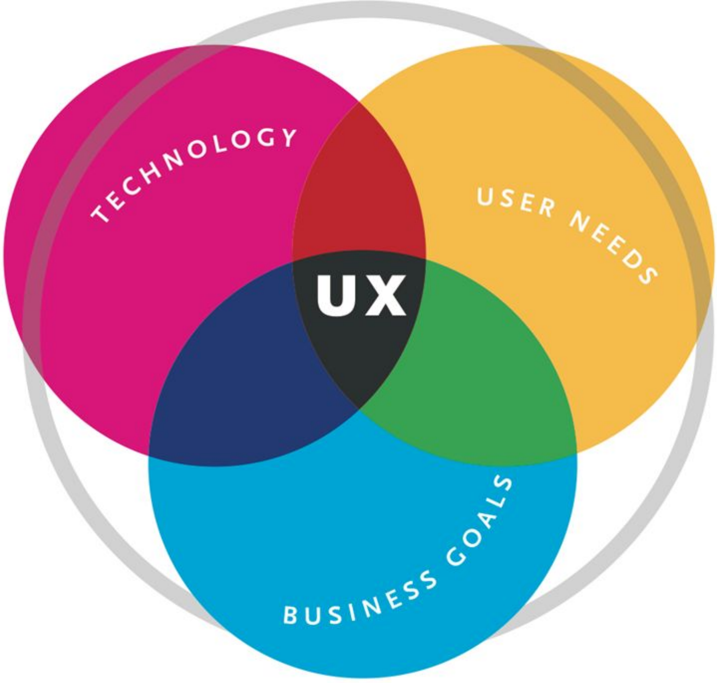
Topics include:

- User Experience
- Users
- Usability
- Visual Design

Topic 1: User Experience(UX)

"After all, usability really just means making sure that something works well: that a person of average (or even below average) ability and experience can use the thing -whether it's a web site, a fighter jet, or a revolving door- for its intended purpose without getting hopelessly frustrated."

Steve Krug, 2006



User Experience determines is how a person feels when interacting with a system. The system could be a website, a web application or - now that the *internet of things* is upon us - a fridge.

Here are some examples of good UX that instinctively appeal to users:

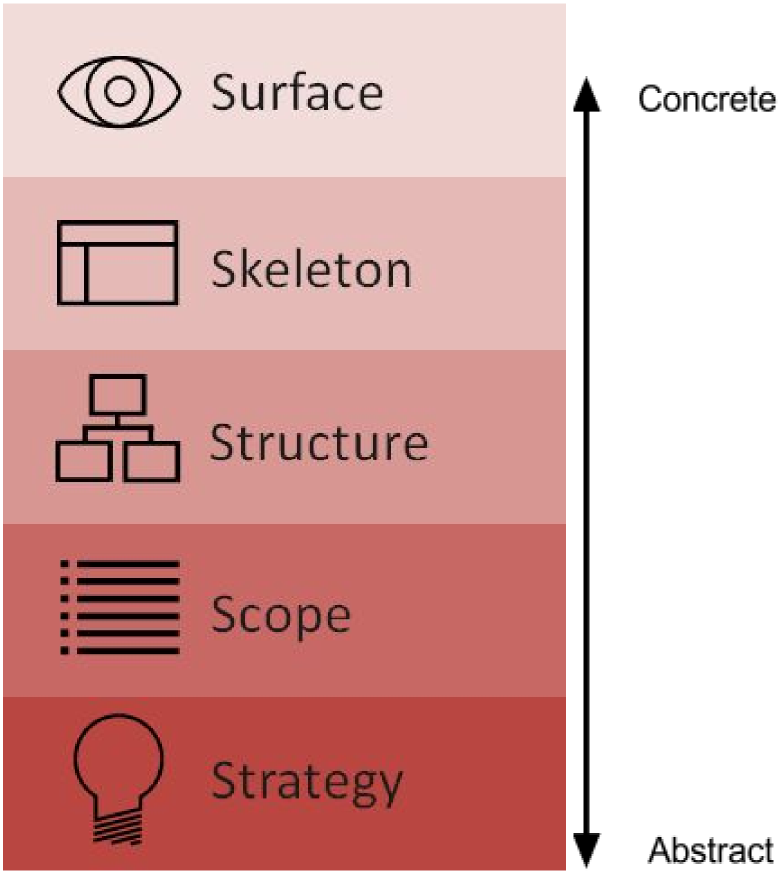
- Beautiful UX
- UX Awards

To achieve results similar to the examples linked to above let's a deeper look at what is involved.

No matter how simple or complex your site, or product is, a good user experience is essential in encouraging users to repeatedly interact with you. As a result of this recognition,a new role has emerged in that of a UX specialist, and even that role can be further broken down to more specialized roles.

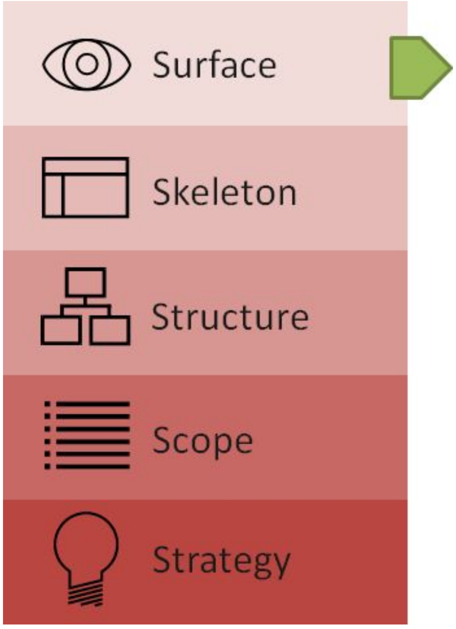
Garrett’s Elements of Ux Stack

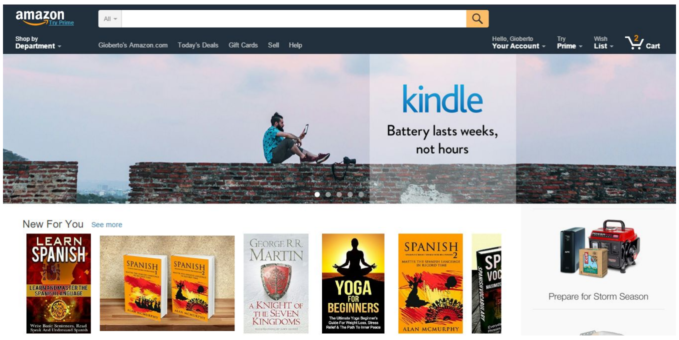
User Experience is Built From Dependent Layers



Surface

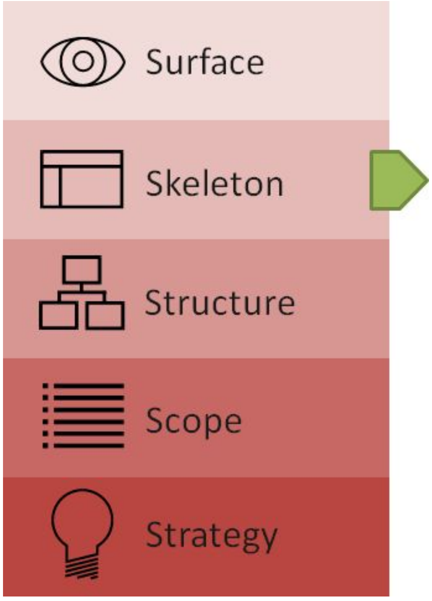
The Surface Layer Describes Finished Visual Design Aspects

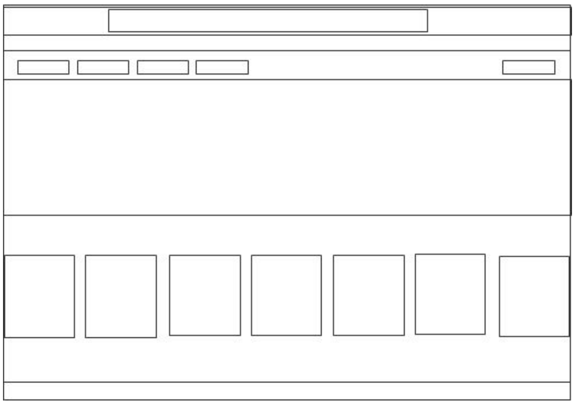




Skeleton

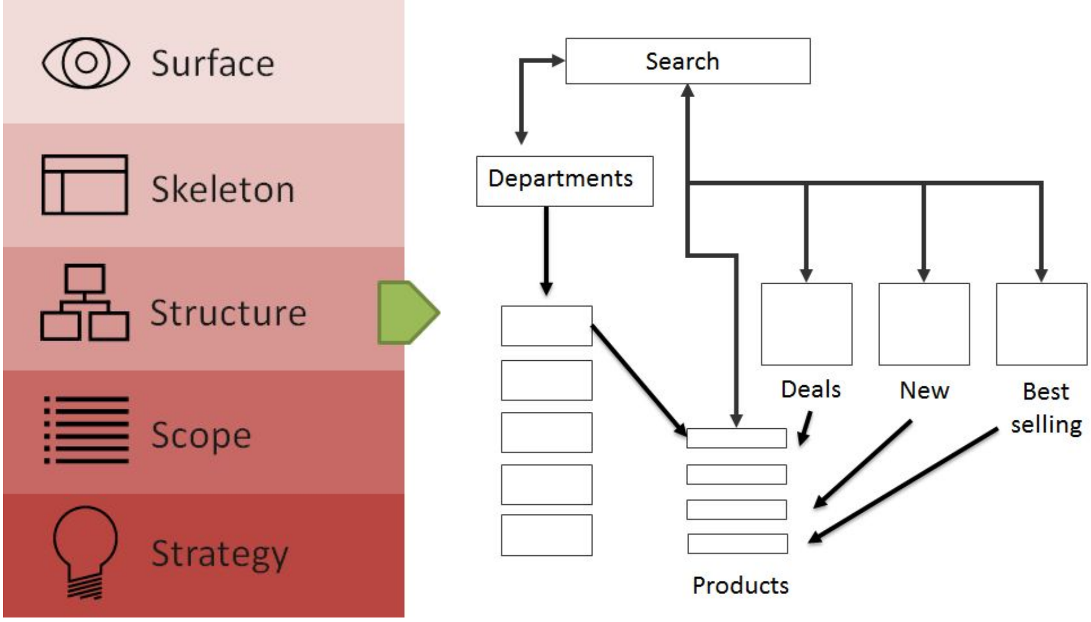
The Skeleton Describes Screen Layout and Functional Compartments in the Screen





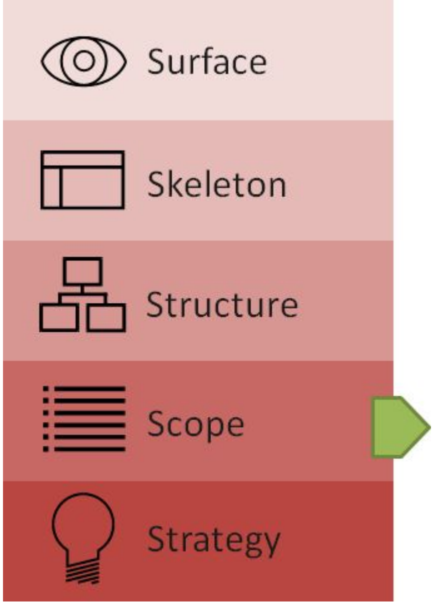
Structure

Structure Defines Navigation from place to lace in the User Interface



Scope

The Places in the User Interface are Built to Support User Task-Centric Scope.

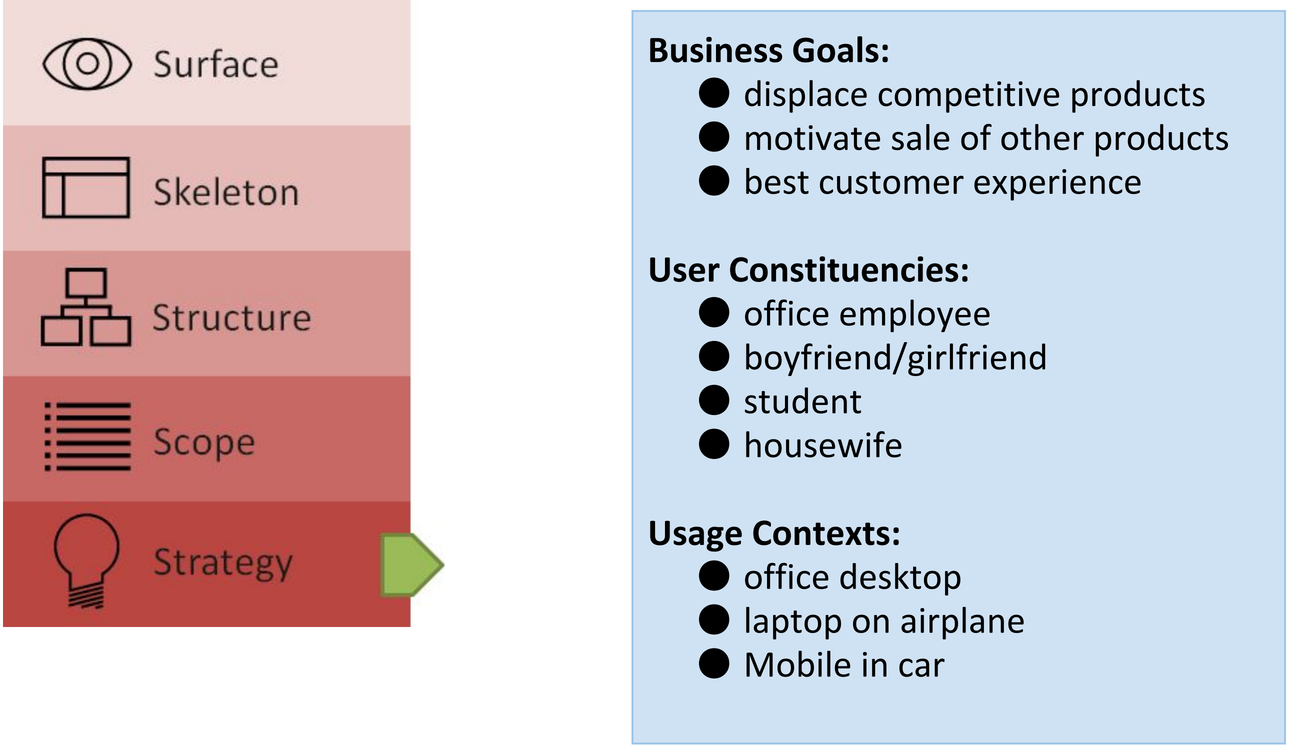


User Tasks:

- search
- buy
- checkout
- pay
- rate
- review
- wishlist
- sell
- browse

Strategy

Business Goals Drive User Constituencies and Contexts Supported To Form Strategy.



Business Goals:

- displace competitive products
- motivate sale of other products
- best customer experience

User Constituencies:

- office employee
- boyfriend/girlfriend
- student
- housewife

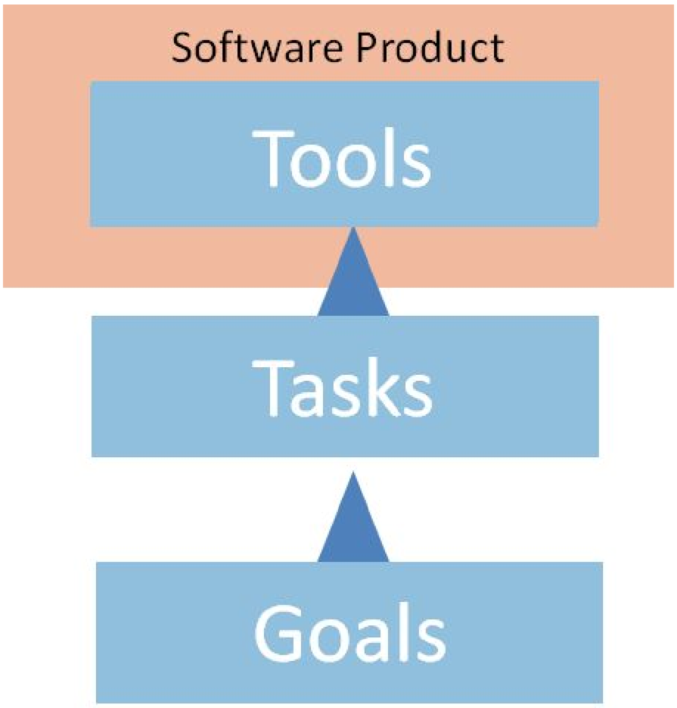
Usage Contexts:

- office desktop
- laptop on airplane
- Mobile in car

Goals, Tasks, & Tools

Understanding the Relationship between Goals, Tasks, & Tools is Critical

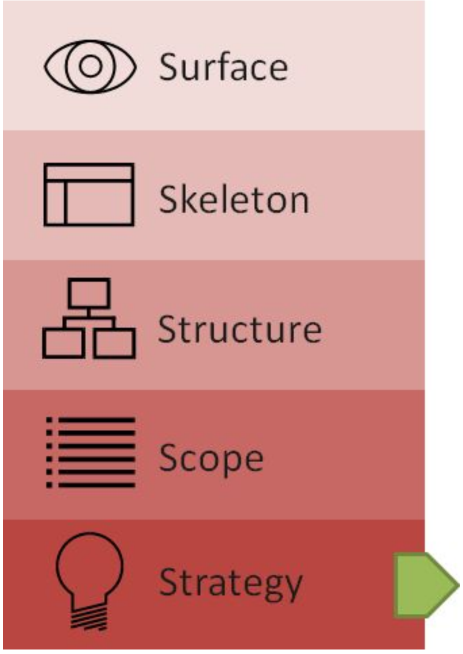
Garret's Elements of Ux Stack Can Apply to the User Experience of Other Complex Products These layers of concerns apply not only to software but a variety of products. In particular, products that support a wide variety of user tasks benefit from this kind of thinking.



Software Product
Tools
Tasks
Goals

Strategy

Think about the requirements for a home.



Goals:

- live comfortably
- eat well
- stay clean
- be healthy
- keep up with Jones's

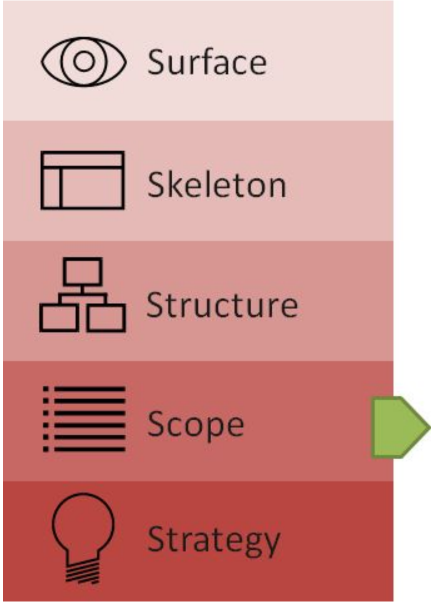
User Constituencies:

- me
- spouse
- child

Usage contexts:

- suburban neighborhood
- near good schools
- near shopping

Scope

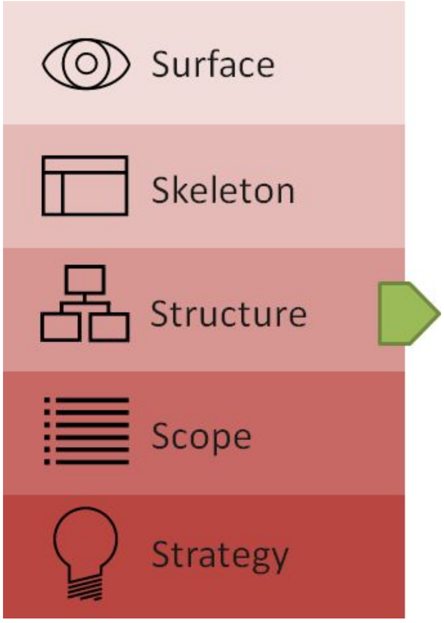


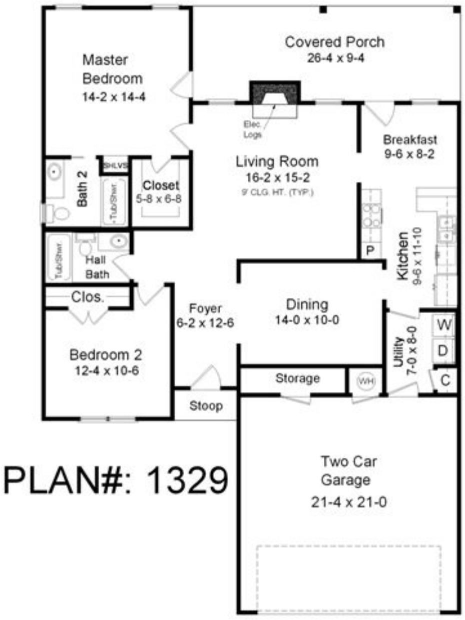
User Tasks:

- store food
- prepare food
- eat food
- sleep
- bathe
- store changes of clothing
- stay out of rain
- entertain guests

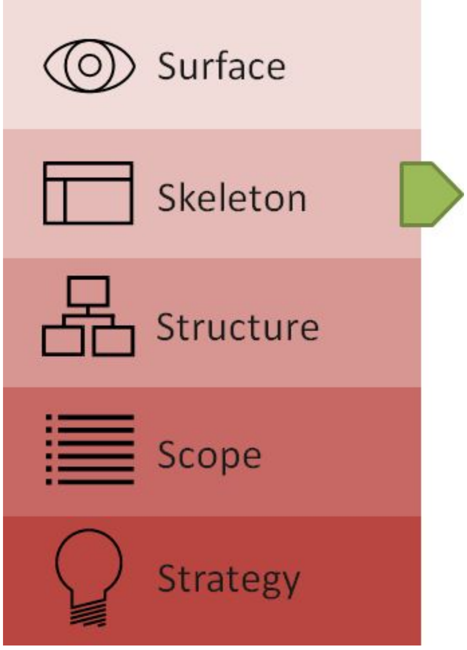
- entertain self

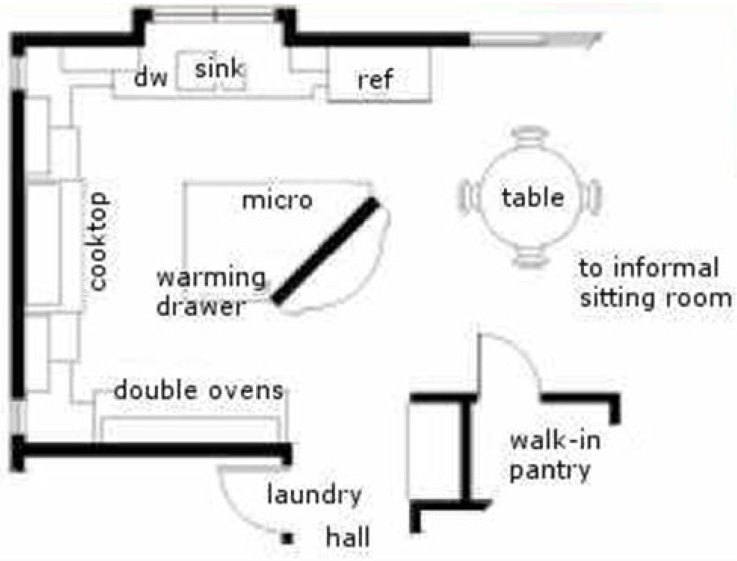
Structure



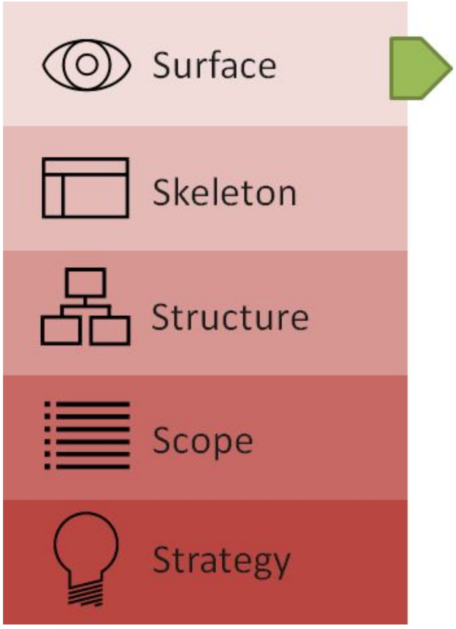


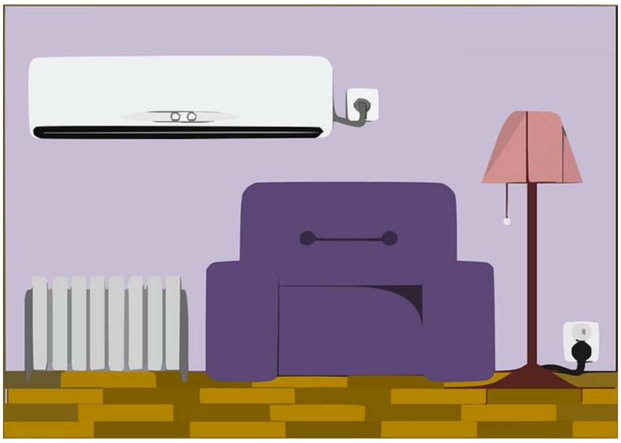
Skeleton

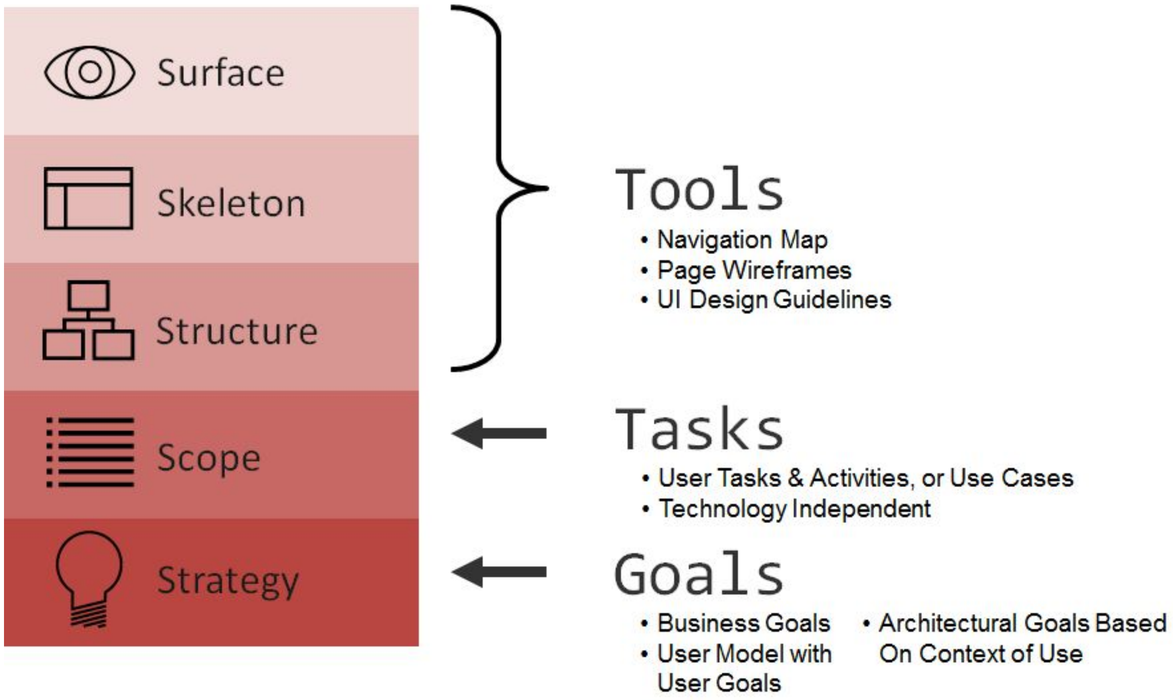

Surface
Skeleton
Structure
Scope
Strategy


dw
sink
ref
cooktop
micro
table
warming drawer
to informal sitting room
double ovens
walk-in pantry
laundry
hall

Surface

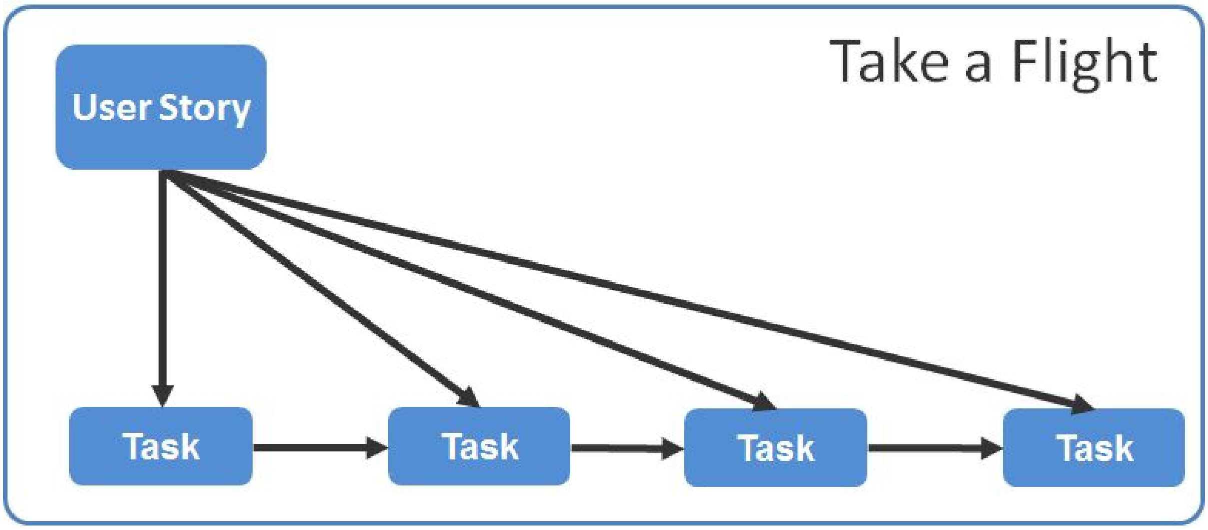

Surface
Skeleton
Structure
Scope
Strategy

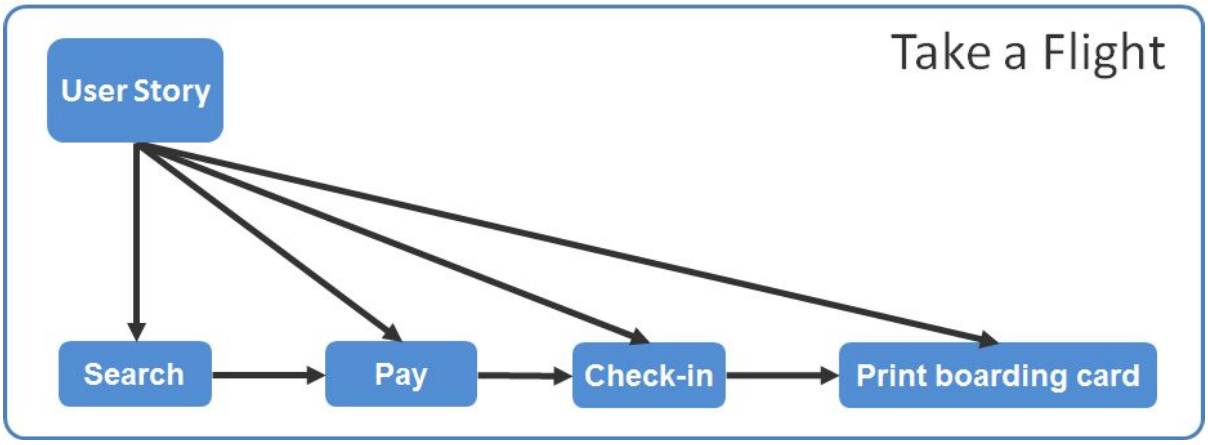




User Interface Designers Often Use “User Tasks” to Describe What People Do

- Tasks require intentional action on behalf of a tool’s user and have an objective that can be completed.
- Tasks decompose into smaller tasks.
- A User Story is used to describe a number of tasks that may be completed in an any order in pursuit of a general goal.
- “Online Check-in” is a task, “Taking a Flight” is an User Story.





Each Layer Validates The Next

- As we move up layers in Garrett's model it's easy to see how each layer validates the previous layer
- This doesn't mean that decisions on each layer must be made before moving on to the next – that's often impractical

When making a decision on one layer, consider the assumptions that must be true on the next layer down.

When information changes on lower layers – information such as business or user goals, or our understanding of tasks, consider the implications on layers above.

Challenge 1

Complete a Garett's layers for the UI stack for the company you have been given.

For Strategy include:

- Business Goals
- User Constituencies
- Usage Contexts

For Scope include:

- User tasks

For Structure include:

- Structural diagram

For Skeleton include:

- Wireframe interface (basic stetch)

For Surface include:

- Visual Design Concepts (high level)

User Centered Design

Refers to a class of methodologies where design decisions are based on some tangible user model. That user model must be based on the research of the users of the application. Rather than requiring users to adapt their attitudes and behaviors in order to learn and use a system, a system can be designed to support its intended users' existing beliefs, attitudes, and behaviors as they relate to the tasks that the system is being designed to support. The result of employing UCD to a system design is a product that offers a more efficient, satisfying, and user-friendly experience for the user, which is likely to increase sales and customer loyalty.

UX DESIGNER (USER EXPERIENCE DESIGNER)

UX designers are primarily concerned with how the product feels. A given design problem has no single right answer. UX designers explore many different approaches to solving a specific user problem. The broad responsibility of a UX designer is to ensure that the product logically flows from one step to the next.

One way that a UX designer might do this is by conducting in-person user tests to observe one's behavior. By identifying verbal and non-verbal stumbling blocks, they refine and iterate to create the "best" user experience.

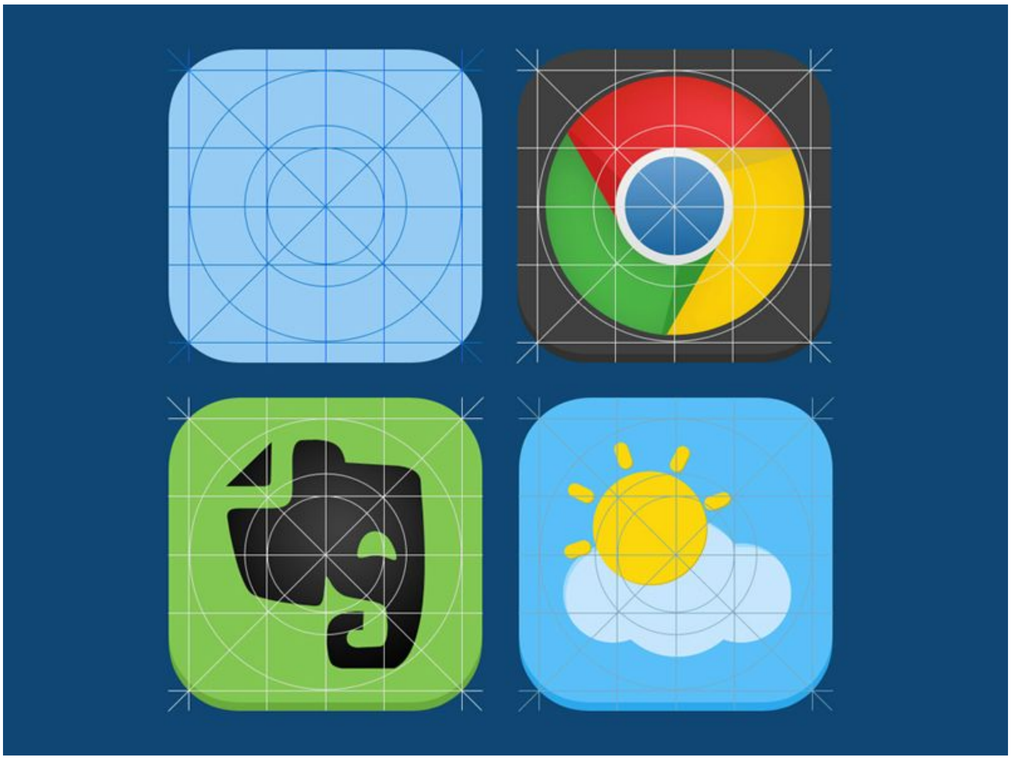
UI DESIGNER (USER INTERFACE DESIGNER)

Unlike UX designers who are concerned with the overall feel of the product, user interface designers are particular about how the product is laid out. They are in charge of designing each screen or page with which a user interacts and ensuring that the UI visually communicates the path that a UX designer has laid out. For example, a UI designer creating an analytics dashboard might front load the most important content at the top, or decide whether a slider or a control knob makes the most intuitive sense to adjust a graph.

UI designers are also typically responsible for creating a cohesive style guide and ensuring that a consistent design language is applied across the product. Maintaining consistency in visual elements and defining behavior such as how to display error or warning states fall under the purview of a UI designer.

VISUAL DESIGNER (GRAPHIC DESIGNER)

A visual designer is the one who pushes pixels. If you ask a non-designer what a designer does, this is probably what comes to mind first. Visual designers are not concerned with how screens link to each other, nor how someone interacts with the product. Instead, their focus is on crafting beautiful icons, controls, and visual elements and making use of suitable typography. Visual designers sweat the small details that others overlook and frequently operate at the 4X to 8X zoom level in Photoshop.



Understanding the Users

To understand our users more we can investigate them to try to understand the motives and needs. We can group the in a general manner or drill right down for create **personas**. Lets look at some of theses groupings.

Actor & Goal

Often a job title or the common name for the type of user in a system.

- **Example:**
- **On-line Shopper**: browse and purchase merchandise online

User Role

Short name describing a user in pursuit of a goal – users change roles as their goals change.

- **Casual Browser:** pass time by browsing products online
- **Comparison Shopper:** compare price and features for items I wish to buy
- **Gift Shopper:** find a gift for someone that likes the types of products this website sells
- **Impatient Buyer:** find what I need and get through the checkout process quickly

User Profile

Adding summary information about the types of users who fill a role or perform as an actor begins a process of "profiling".

- Users: 50,000 customer visit this sporting goods website monthly
- Activities: browsing, price comparing, gift shopping, handling returns
- Computer Skills: vary wildly from first time users to expert – although moderate computer skills are typical
- Domain expertise: typical customers are avid outdoor enthusiasts

Persona

Choosing specific characteristics of a person and compiling those into a archetypal description of that person creates a strong design target.

Building Personas

Personas are based on research, e.g. interviews, contextual enquiry, observation, focus groups, surveys, literature reviews, etc. They can be useful in personalizing the target user for our application and can provide a focus for the development process. A persona may not be one person we have met in the requirements phase but a representation of a target user.

Personas

- Describe users’ behavioural patterns, goals and motivation
- Are represented as individual people.
- Express and focus on the major needs and expectations of the most important user groups
- Give a clear picture of the user's expectations and how they're likely to use the site
- Aid in uncovering universal features and functionality

From the user research we can construct **personas** and a **User Story** which allow us the recreate the user experience for a very specific user. Personas are focused, concise user models that we can use to base our decisions on. They will be based on the users we have met. We must ask some questions of our users to figure out what our personas might look like:

- What attributes do our users have in common?
- What attributes vary within our user group?

We can group common attributes into roles and then convert the roles into specific personas:

- Fictional name
- Demographics such as age, education, ethnicity, and family status
- Picture can be very helpful
- Job titles and major responsibilities
- Occupation
- IT/ domain experience
- Personality

- The goals and tasks they are trying to complete using the site
- Frequency of Use: how often are they likely to use the site?
- Their physical, social, and technological environment
- A quote that sums up what matters most to the persona as it relates to your site

If, say, we name our persona as ‘Sarah’. We can continually ask ourselves:

- what would Sarah think of this ?
- are we meeting Sarah’s goals ?
- will Sarah be able to achieve her goals ?
- does the application meet Sarah’s aspirations ?

The scope of the project will determine how many personas are useful. There is usually a **primary persona** and a number of **secondary personas.**

Challenge 2

Identify the following for your project:

- Actors
- Roles
- Profiles (if possible)
- Personas
  - Create a Primary and secondary persona for your project
  - Use all the steps in **Building a Persona** (above)

Usability

Usability refers to the ability of a specific type of user to be able to effectively carry out a task using a product. Usability is usually measured through testing. Given a number of test subjects that reflects the type of user that will use the application:

- how many successfully complete a task.
- on average how quickly do they complete that task.
- on average how many user errors are made while attempting to complete that task.

Design and Usability Principles

There are existing and evolving standards for interaction design. One of the 'Gurus' of usability and interaction design is Jakob Nielsen. He has developed '**10 Usability Heuristics for User Interface Design'.** They are called "heuristics" because they are broad rules of thumb and not specific guidelines.

1) Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2) Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3) User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4) Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5) Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

(Read full article on preventing user errors.)

6) Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

(Read full article on recognition vs. recall in UX.)

7) Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8) Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9) Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10) Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

See examples of 10 principles in web applications

One of the best books on creating web interfaces is Designing Web Interfaces: Principles and Patterns for Rich Interaction which is available online . In this book the author divides design patterns and best practice into six design principles.

- Make it direct
- Keep it lightweight
- Stay on Page
- Provide and Invitation
- Use Transitions
- React Immediately

Review each of these principles and note examples (Hover over each icon to reveal examples)

Prototype Design

Do not fall into the trap of coding early in a project which can lead to a huge waste of resources. Better to invest in scissors, paper and some coloured pencils. Create a paper version of the interface and walk through the user experience. The designs should focus on the user's flow throughout the application rather than specific controls .

Prototyping

The **prototype** is a simple working version of the project in which basic functionality can be shown without the full features.

Why Prototype ?

- First, prototypes are much faster to build than finished implementations, so we can evaluate them sooner and get early feedback about the good and bad points of a design.
- Second, if we have a design decision that is hard to resolve, we can build multiple prototypes embodying the different alternatives of the decision and get feedback.
- Third, if we discover problems in the design, a prototype can be changed more easily, for the same reasons it could be built faster. Prototypes are more malleable. Most important, if the design flaws are serious, a prototype can be thrown away.
- It is cheaper than building a fully developed product.

What is Prototyping ?

The development of incomplete representations of a target system for testing purposes and as a way of understanding the difficulties of development and the scale of the problem

Prototype can broadly be split into High-fidelity & Low-fidelity prototypes

Low-fidelity

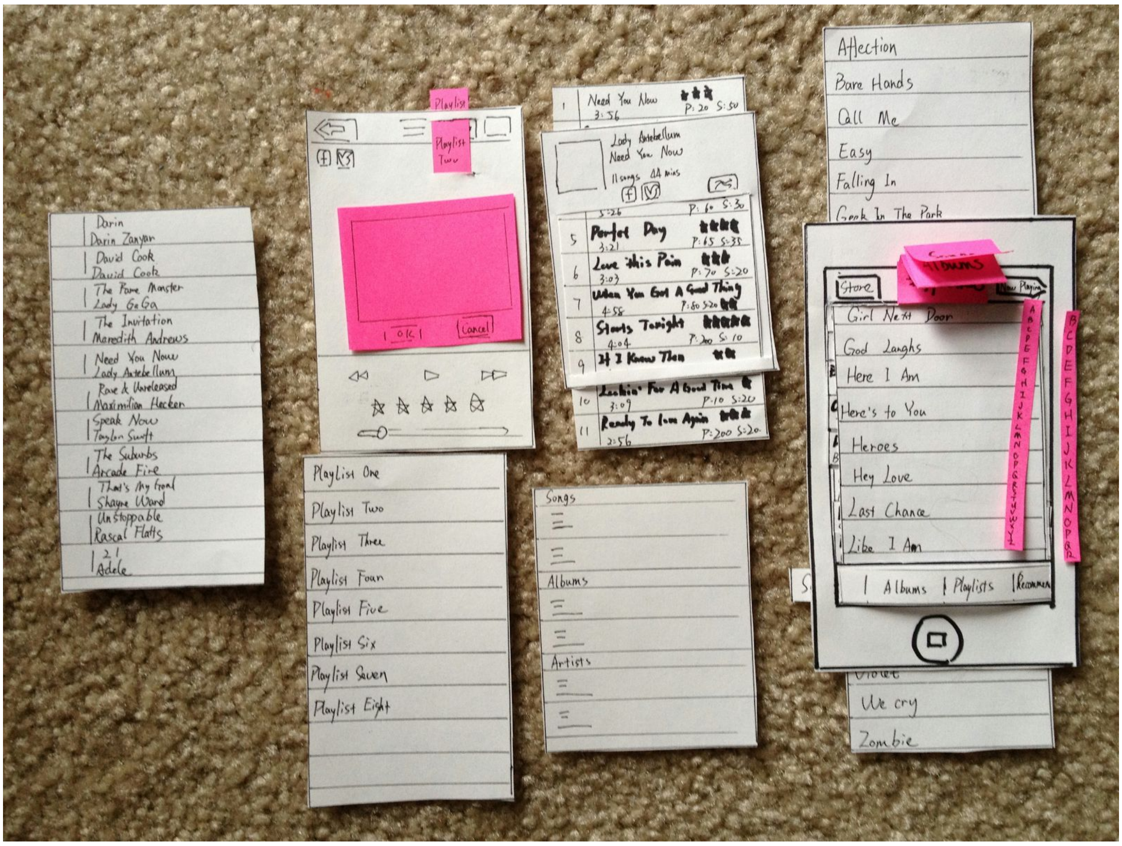
–Sketching

–Wire framing

Sketching

Roughly drawing first ideas can help to communicate with customers, users or other stakeholders

An advantage of paper prototyping: no special skills are required. So graphic designers, usability specialists, and even users can help create prototypes and operate them.



e.g. http://www.wireframeshowcase.com/wireframes/detail/iplagg_webshop

High-fidelity

mockups

working Prototypes

Challenge 3

Create a wireframe of your interface with **balsamiq**

Allow the user to complete a specific task while interacting with the wireframe.

Challenge 4

Perform a Usability test on the website assigned to you.

Download the usability form the lesson folder.

Keep in mind the topics of the lesson.

Summary

The early phases of **user research** and **prototyping** should set us on the right path, but it takes a concerted effort to stay on track. For most of the rest of this bootcamp we will concentrate on the development phase of a project. But it's important to do the groundwork before embarking on the development phase.

Paying attention to user research, prototyping and adhering to design guidelines and principles will set you down the right path, keep you focused and keep your users engaged .

The next steps in the lifecycle of the application are developing, testing and deployment which is beyond this introductory lesson. User Experience is a vast subject which we have just dipped our toes in and and our aim is to give you an appreciation for it's importance when developing Web applications. There are many resources on User Experience on the web to continue your research

http://www.smashingmagazine.com/2010/10/what-is-user-experience-design-overview-tools-and-resources/

Common design mistakes in web design

Prototyping Tools http://wireframes.linowski.ca/category/tools/

UX Mag http://uxmag.com

Additionally some books which may be helpful:

Cooper, A., Reimann, R. & Cronin, D. (2007). About Face 3, Indianapolis: Wiley; chapter 5: Modeling Users

Sharp, H., Rogers, Y. & Preece, J. (2006). Interaction Design. Chichester: Wiley; chapter 9: The Process of Interaction Design

Scott B. & Neil, T. (2009). Designing Web Interfaces. Cambridge: O'Reilly.

Tidwell J. (2006). Designing Interfaces. Cambridge: O'Reilly.