1.4: User-Centered Design Process

#### **Introduction**

Welcome back! Had enough psychology yet? You’d be surprised at how well a UX designer needs to understand the human (and the consumer!) mind in order to be successful. In the previous Exercise, we taught you how to identify and classify users’ needs according to the Design Hierarchy of Needs. We also discussed how important it is to remove points of conflict, or friction, from your product in order to make it simple and efficient.

Once you understand the importance of the user and their needs, you need to be able to identify and meet those needs. In this Exercise, we’ll discuss user-centered design and its usefulness in building products that satisfy your users’ needs. We’ll also take an in-depth look at the key phases of design. By the end of this Exercise, you’ll have good working knowledge of what goes into successful user-centered design, as well as the importance of users in the design process.

#### **What Is User-Centered Design?**

Product features are oftentimes determined by business goals and/or hardware capabilities. While this can occasionally lead to interesting, intuitive solutions, it usually ignores the most important aspect of the design process: the person who'll be using the product.

**User-centered design (UCD)** is a framework that places the user at the center of the design and development process. It focuses on who'll be using a product, the tasks they need to complete while using the project, and the environment in which they'll use the product (i.e., a product’s context of use). UCD addresses the whole user experience and aims to improve human well-being by creating products that are useful, usable, and accessible. This is achieved through focusing on usability and employing a user-driven evaluation and iteration process.

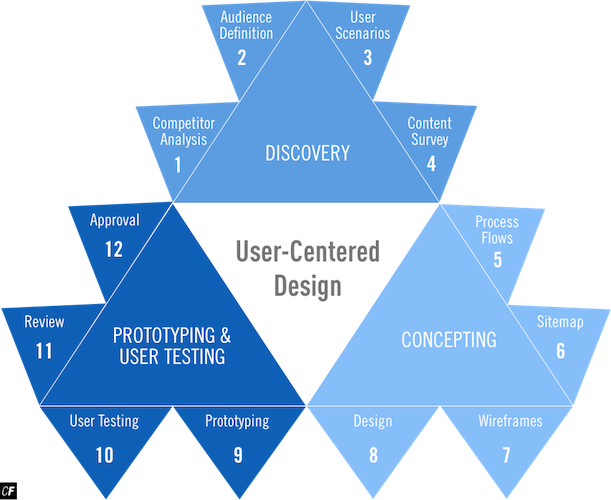
It’s important to note that you’ll often find the terms user-centered design and **human-centered design (HCD)** used interchangeably. They are, indeed, very similar, but human-centered design broadens the focus to consider the impact on all stakeholders who might be affected by the design of a system or product beyond just the end user’s direct interaction with the system or product itself.

Another important field of research and practice that's often associated with UCD is **human-computer interaction (HCI)**. HCI aims to understand how people interact with computer systems and to what degree computer systems are designed to facilitate successful interactions with humans. HCI, therefore, is a broader and separate field from UCD, though both practices share some similarities in their aim to design for human/user needs and capabilities.

#### **The Phases of UCD**

In Fundamentals, we focused on the Design Thinking Process, which we'll talk about again later in this Achievement. You should be aware, however, that there are many ways to think about design processes. The User-Centered Design Process is one of them. There have been a few different versions of the UCD process throughout the years, though the basic principles of each remain the same. Here are the typical phases of UCD:

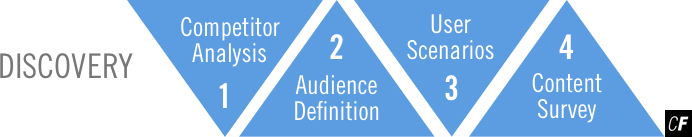
* **Discovery:** The goal of this phase is to identify those who’ll be using your product, the conditions they’ll use it under, any business requirements you might have, and finally, the user needs of your product. To put it simply, this is where you determine what it is you’re building. This phase typically consists of competitive analyses, audience definition, user scenarios, and surveys.
* **Concepting:** This phase is all about creating design solutions that address your business and user requirements. You’ll be crafting wireframes, user flows, mockups, and finalized interface designs.
* **Prototyping and User Testing:** During this phase, you’ll determine if your design solution is effective via user testing (ideally with actual users). We’ll discuss specific methods of testing later on in the course.



#### **An Iterative Process**

You’ll notice that the process flow is represented as a clockwise motion. This is because the UCD process was designed to allow for easy iteration. Iteration is an important part of the design process as it allows you to constantly change and update your software to more fully meet the needs of your users. Once you reach the end of the process, a test of your solution’s effectiveness will determine if you need to start the process over again.

Let’s look at each sub-stage of the process individually to understand how each piece fits into the overall process:



* **Competitive Analysis:** Before you do anything else, you need to understand the competitors in your space to decide how your software will compete.
* **Audience Definition:** Making informed design decisions is impossible without first identifying the recipients of your design. We’ll dig deeper into this process in a later Exercise.
* **User Scenarios:** By creating scenarios in which your potential users will need your software to accomplish a goal, you can identify features and functions crucial to the success of your app.
* **Content Survey:** Surveys are a great way to learn about your audience and hone in on functions that might address their needs.



* **Process Flows:** This is where you get into the specific paths a user can take within your software, typically via user stories and flow charts (more on these in a later Exercise).
* **Sitemap:** This is a diagram that shows the navigation layout of a particular website or app. A sitemap details all of the pages contained within a website or app, showing how they're connected together and their corresponding hierarchy.
* **Wireframes:** Wireframes help you quickly lay out your designs before you begin pushing pixels around in Sketch. [Balsamiq](https://balsamiq.com/) is a great tool for quickly drafting wireframes.
* **Design:** Armed with wireframes and user flows, you can now start mocking up your actual design in Sketch (or another design program of your choice).



* **Prototyping:** Prototypes allow you to test the functionality of your design against actual users. They can be as simple or as complicated as necessary. Later down the road, we’ll introduce you to programs you can use to put together quick prototypes—no coding required!
* **User Testing:** In this phase, you’ll present your finalized designs and/or prototypes to actual users and analyze the results. Where are they getting stuck? What features are they missing? This is your chance to see where your current design can be improved.
* **Review:** Now, it’s time to take a step back and analyze everything you’ve learned. By this point, you should have some concrete data as to what was and wasn’t effective in your app.
* **Approval:** This is the final phase, which means it’s time to make the ultimate decision about what stays and what goes. If something works, it makes its way into the development phase and, ultimately, into the product. If something doesn’t work, you start the process over again and attempt to improve it based on your findings.

TIP!  
If you’re having trouble remembering the stages, try creating flashcards. [Chegg](https://www.chegg.com/mobile/#flashcards) and [Anki](http://ankisrs.net/" \t "_blank) are handy apps for creating flashcards on your phone—you may have even discovered these while completing your project for the UX Fundamentals course. You can also try highlighting the terms using our in-platform notes feature!

#### **The User Mindset**



As a UX designer, you need to do more than solve your users’ problems. You need to solve them more efficiently, more swiftly, and more easily than any other solution on the market. Discovering, understanding, and catering to the needs of your users is the basis of user-centered design.

It's important to remember that you are not your user. Never assume that everyone is like you, especially when it comes to designing websites and software. This way of thinking can lead to mistakes and inefficient designs.

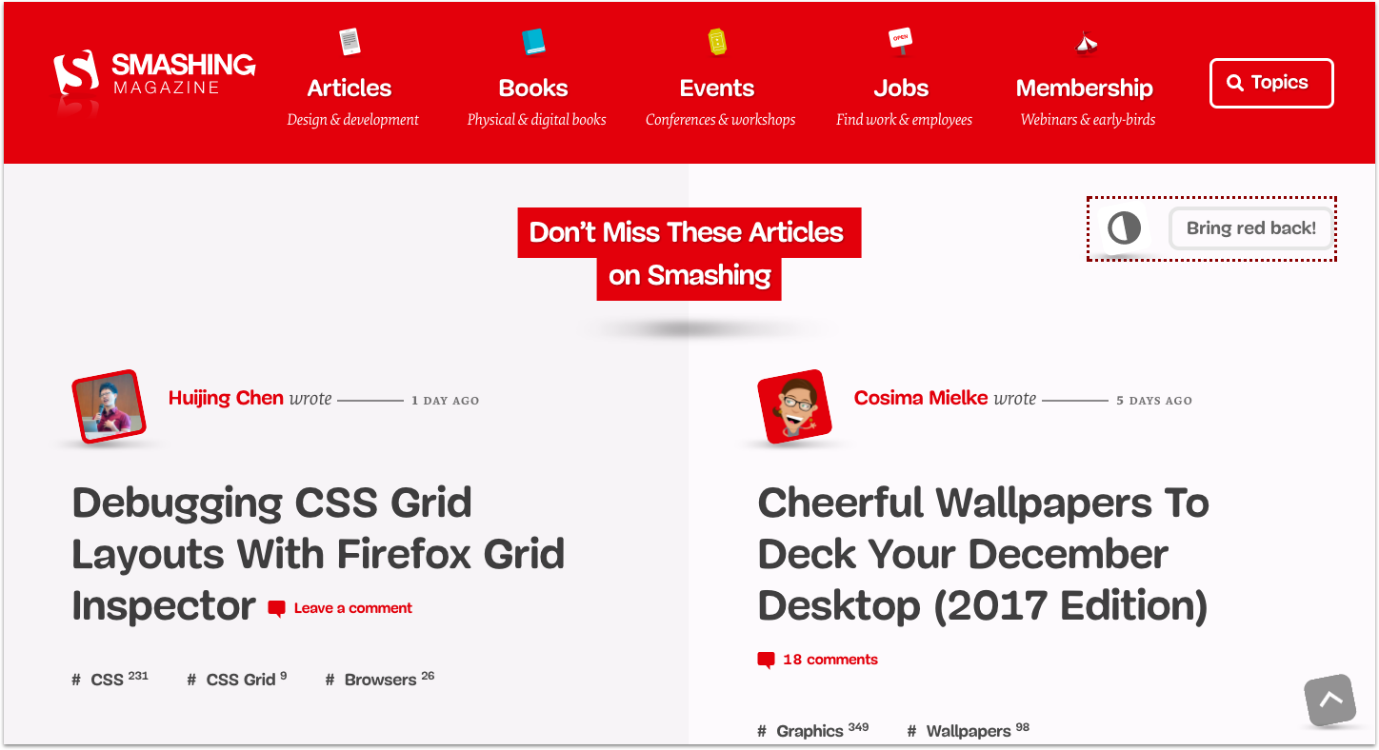
As a designer, it’s easy to grow attached to your project. It’s your baby, after all! This kind of mindset can be dangerous, however, as it can affect your judgment and bias your thinking. For example, data might reveal that a feature isn’t meeting user needs, but your personal bias may incline you to keep or refine the feature rather than scrap it.

This is why putting yourself in a user mindset is so important. Remember, you aren’t creating this software for yourself—you’re creating it for someone with completely different needs and motivations than your own.

#### **Digital Products in Context**

Part of exploring the user mindset is to understand the context in which your product will be used and to design for this context accordingly. For example, is your product a piece of business software likely to be used by professionals in an office environment, or is it a car-sharing platform for people on the go? Understanding the context in which users are likely to use your product is essential in determining the most suitable design approach. Designing an application for public transport that works well on desktop devices but not on mobile is likely to lead to a frustrating user experience for those trying to use the application on their smartphone when they’re out and about.

Taking the time to understand this aspect of your users’ mindsets will enable you to determine the most appropriate type of platform for your product, that is: a website, a web application, or a native app. Although your project requires you to design a web application, let’s take a closer look at each of these options to better understand how you might choose one over the other when designing a digital product.



###### **Smashing Magazine website**

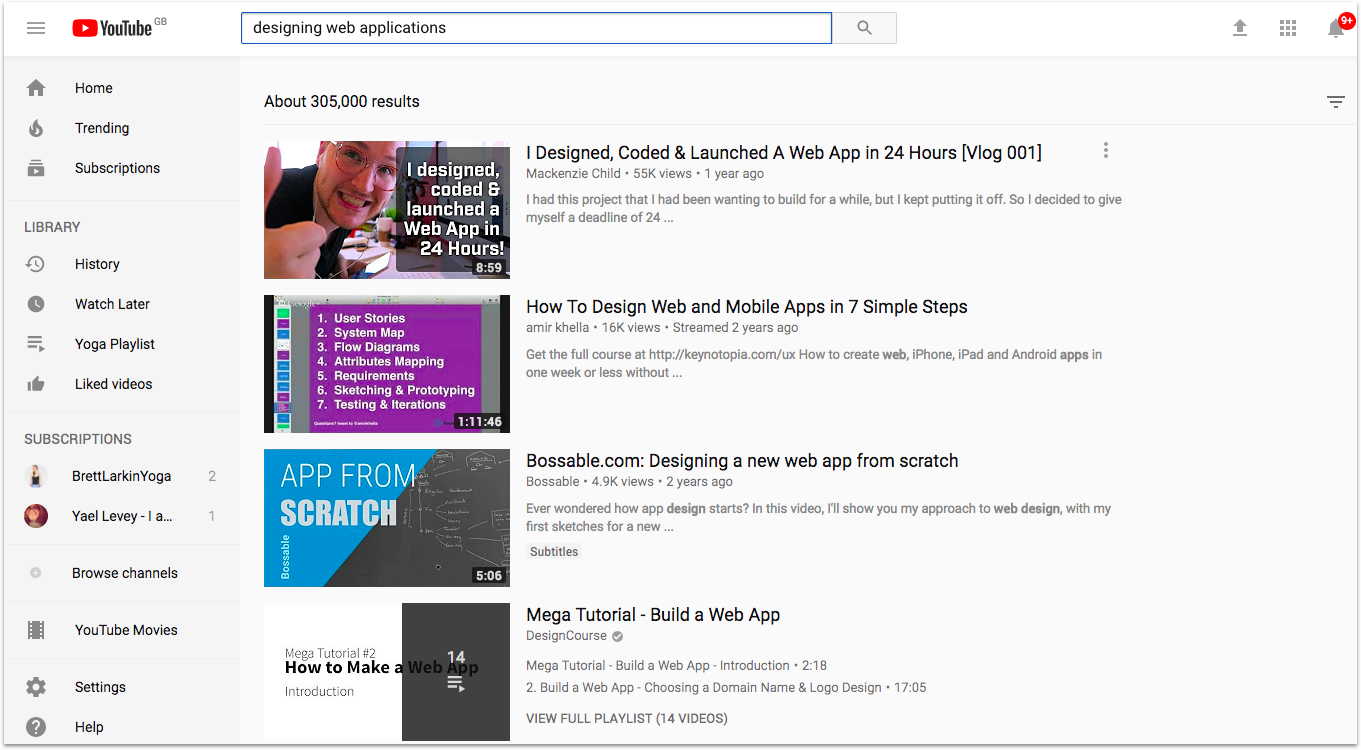
##### **Website**

A website typically provides static information with minimal interaction from users. The main goal of a website is to provide users with informational content they can read, watch, or listen to. News websites such as [BBC News](http://www.bbc.com/news) or [CNN](http://edition.cnn.com/) are good examples, as are online magazines such as [Smashing Magazine](https://www.smashingmagazine.com/). Websites don't usually require users to log in or register, and the information provided tends to be generic (i.e., the same for everyone who visits that website). There may be small elements of interactivity such as entering your email address to sign up for a newsletter or changing the language in which you view the website, but, otherwise, websites provide a one-way stream of information to the user.

##### **Web Application**

A web application, or web app, unlike a website, is characterized by the ability to interact with and manipulate information and data. An example of a web app is an online banking portal, which enables customers to view and filter their personal information as well as carry out financial transactions. Other examples include ecommerce sites such as [Amazon](https://www.amazon.com/), where customers can search, filter, create wish lists, purchase products, and view personalized recommendations. A web application will also require users to register and log in to use the application and, as such, will provide users with personalized data based on the information provided. Both web applications and websites run in the browser (e.g., Firefox or Chrome) and will usually work with multiple operating systems such as Windows or MacOS.

Some other examples of web applications you might have used include YouTube, Google Docs, and Dropbox.



###### **YouTube desktop website**

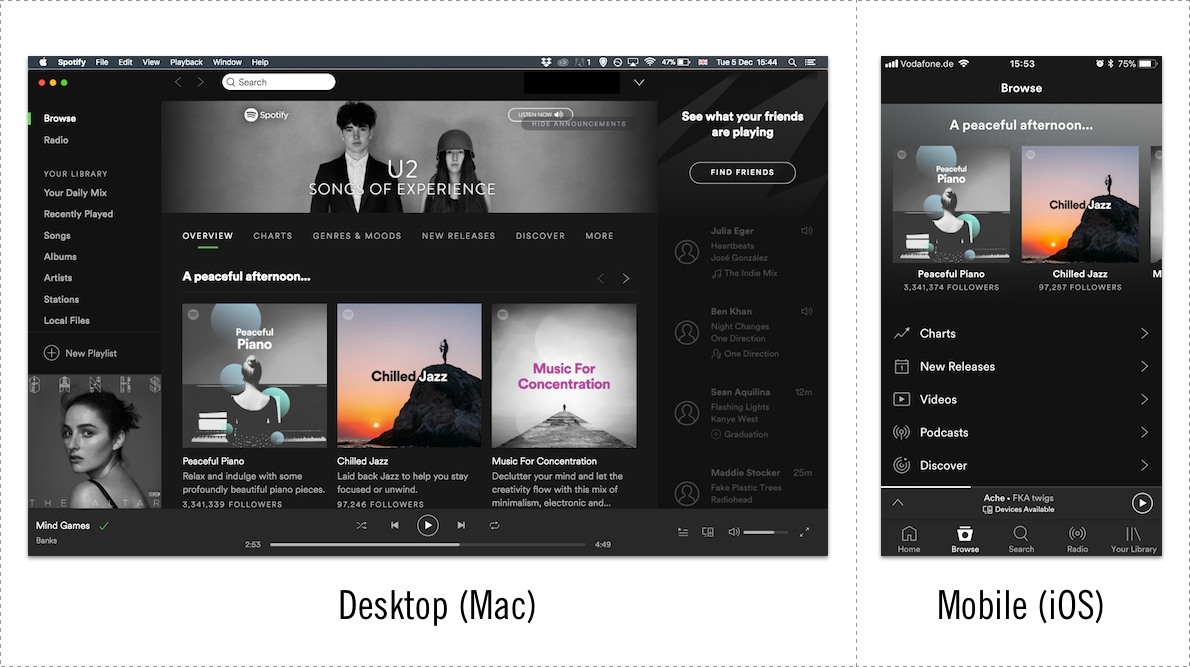
##### **Responsive Web Design**

Before we go any further, let’s briefly explore the concept of responsive web design. Both websites and web applications can be built using this technical approach that enables webpages to re-organize their content based on the type and size of the devicethey're being viewed on. In the project brief, you’ll see that you're being asked to create a “responsive web application.” This simply means that you need to design a web application capable of altering how it displays its content based on the type of device it's being viewed on. Don’t worry if all this sounds a bit complicated right now. We’ll be looking at this concept in more depth later on in the course.

##### **Native Application**

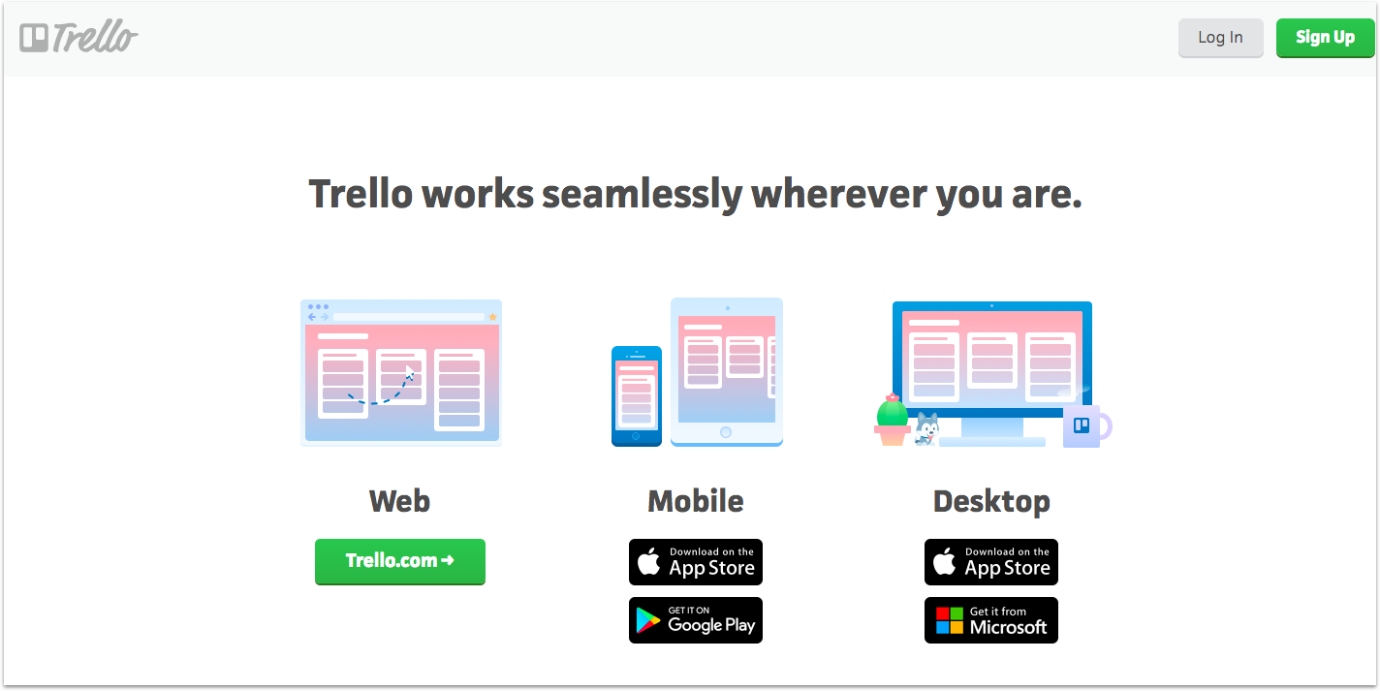
Lastly, there are native apps. Unlike websites and web applications that can run in the browser on any device, native apps will only work with a specific operating system and must be downloaded to a device running that particular operating system. Apps in the Google Play Store, for example, can only be downloaded to smartphones and tablets that use the Android operating system. Native apps are also available for use on desktop devices; for example, MacBook users can download native apps from the Apple App Store to use on MacBook computers. Native apps vary in the nature of their content and interactivity; what differentiates them from websites and web applications is their specificity to a certain operating system and inability to run in the browser.

Some examples of native apps that you might be familiar with are Spotify, WhatsApp, Instagram, Slack, and Trello (though these offer responsive web app versions that can be run in the browser, as well).



###### **The Spotify native app for both desktop (Mac) and mobile (iOS)**

When you use these apps, do you generally download the app to your phone or device? Or do you use them through your browser? Check out [Trello](https://trello.com/), in particular, that shows examples of what their app looks like in all its versions: web, mobile, and desktop.



#### **Summary**

Successful UX design leads to satisfied users, but the only way to achieve successful UX design is to understand what the user wants. This is why user-centered design is so important. By putting yourself in the shoes—or minds—of your users, you might discover that aspects of your app or website aren't as efficient or easy as you first thought. Thorough research and feedback during the design process is integral in designing a system that appeals to as many users as possible.

As you work your way through the project for this course, put these methods to the test! Apply these principles to your own design process to get into the user-centered design mindset.

In the next Exercise, you'll learn another very popular design methodology useful in UX but which is heavier on the ideation side. But first, the Task!

#### **Resources**

* [Benefits of UCD](https://www.usability.gov/what-and-why/benefits-of-ucd.html)
* [Design Is a Process, Not a Methodology](http://www.uxmatters.com/mt/archives/2010/07/design-is-a-process-not-a-methodology.php)
* [The Building Blocks of Progressive Web Apps](https://www.smashingmagazine.com/2016/09/the-building-blocks-of-progressive-web-apps/)
* [Progressive Web Apps: A New Way to Deliver Amazing User Experiences on the Web](https://developers.google.com/web/progressive-web-apps/)