# Table of Contents

[Table of Contents](#_pcgre8j0873o)

[Meta-Data](#_wmw6vju0tvzt)

[Lesson Goals](#_y0f332pzpsd3)

[Lesson Outcomes](#_yuun61fn00lj)

[Assessments](#_qy4klj35a395)

[Lesson Plan](#_jhf5z8xjau2m)

[Script](#_x1hgyjwhsibz)

[3.3.1 Introduction](#_xgus6bu1y1vv)

[3.3.1.1 Headshot Studio](#_uwwts2eh1p4)

[3.3.2 Data Inventory](#_6buz6eptqopo)

[3.3.2.1 Headshot Studio](#_ex5fe3d0seyh)

[3.3.3 The Problem Space](#_2tvekfbhpeu3)

[3.3.3.1 David’s House (Security System)](#_rnejalot43i8)

[3.3.4 User Types](#_mmkzc4dfcjbx)

[3.3.4.1 The Park](#_k6cpirvnh85a)

[3.3.5 5 Tips: Avoiding Bias in Needfinding](#_9coakav4r8so)

[3.3.5.1 Headshot Studio](#_ryqlb9w8cttf)

[3.3.6 Naturalistic Observation](#_1ymzi73exysv)

[3.3.6.1 The Park](#_1de1acn31a2m)

[3.3.7 5 Tips: Naturalistic Observation](#_lv07ya4mbtii)

[3.3.7.1 Headshot Studio](#_xy8882oi9vb8)

[3.3.8 Participant Observation](#_lmhagm8b3nnc)

[3.3.8.1 The Park](#_8rzk1jf143q0)

[3.3.8.2 The Park](#_vmz0fg6nw2rp)

[3.3.9 Hacks and Workarounds](#_fuhz97w3h9gy)

[3.3.9.1 David’s House (Office)](#_oef6g7nfjl8v)

[3.3.10 Errors](#_exuu62gdonpc)

[3.3.10.1 David’s House (Office)](#_txrgygwprxw7)

[3.3.10.2 The Park](#_kb7o72khg18p)

[3.3.11 Apprenticeship and Ethnography](#_ndyt3yrml8m)

[3.3.11.1 Udacity Office (Atlanta)](#_x4rk0u78spwo)

[3.3.12 Interviews and Focus Groups](#_vns25fu7p41i)

[3.3.12.1 Udacity Office (Atlanta)](#_vooml7cc9z5)

[3.3.13 5 Tips: Interviews](#_3roctctcnz4t)

[3.3.13.1 Headshot Studio](#_mh3xk7rb8okw)

[3.3.14 Exercise: Interviews](#_9khp229yygb0)

[3.3.14.1 Tablet Studio](#_tbkd4ds7vlv5)

[3.3.14.2 Exercise](#_67mfff1bkvls)

[3.3.14.3 Tablet Studio](#_onyubtlqeri9)

[3.3.15 Think-Aloud](#_can7dkiwtdh5)

[3.3.15.1 The Park](#_2n42ia63r7le)

[3.3.16 Surveys](#_m60rji7p1fc8)

[3.3.16.1 Headshot Studio](#_kpwtvzpdn552)

[3.3.17 5 Tips: Surveys](#_cpke6vbvdtc3)

[3.3.17.1 Headshot Studio](#_mlatqchn4u3s)

[3.3.18 Exercise: Surveys](#_a31uuxmzklqa)

[3.3.18.1 Tablet Studio](#_72r8n6s2dhe4)

[3.3.18.2 Exercise](#_88365zm173ws)

[3.3.18.3 Tablet Studio](#_gtbie12sti4z)

[3.3.19 Other Data Gathering Methods](#_my6a1aslh58e)

[3.3.19.1 Headshot Studio](#_h25ryqmgb4n)

[3.3.20 Exercise: Needfinding Pros and Cons](#_gftxcwhsf9o0)

[3.3.20.1 Tablet Studio](#_46eeddasseyu)

[3.3.20.2 Exercise](#_pv95033r7d6u)

[3.3.20.3 Tablet Studio](#_c8zkmktew7b5)

[3.3.21 Design Challenge: Needfinding for Book Reading](#_pggwqgaas96m)

[3.3.21.1 Headshot Studio](#_vvhmju3wjfw7)

[3.3.21.2 Exercise](#_jf5s6vp3uco2)

[3.3.21.3 Headshot Studio](#_1ok7qggvt7ei)

[3.3.22 Iterative Needfinding](#_d06f3o8ydwb6)

[3.3.22.1 Tablet Studio](#_glqukz3d4w2z)

[3.3.23 Revisiting the Inventory](#_ozgnd1x40nxi)

[3.3.23.1 Headshot Studio](#_r8b63a86zrpu)

[3.3.24 Representing the Need](#_o8bgmiizuq65)

[3.3.24.1 Tablet Studio](#_sx4yup16qmxn)

[3.3.25 Defining the Requirements](#_m2abfut8vrp)

[3.3.25.1 Headshot Studio](#_3q0zqg8km90k)

[3.3.26 Design Challenge: Needfinding and MOOC Recording](#_w6x6673en6dv)

[3.3.26.1 Headshot Studio (Behind Camera)](#_ms9tlhc40x35)

[3.3.27 Exploring HCI: Needfinding](#_jx24ik3d1olz)

[3.3.27.1 Headshot Studio](#_stucbcs365bh)

[3.3.28 Conclusion](#_2mlhdqe90qw)

[3.3.28.1 Headshot Studio](#_m3yxbsu2evcc)

# Meta-Data

## Lesson Goals

* Students will understand the role of needfinding in the design life cycle.
* Students will understand various methods for conducting needfinding.
* Students will understand the types of data gathered during needfinding exercises.
* Students will understand the relationship between data gathered during needfinding and synthesized models of human tasks.

## Lesson Outcomes

* Students will be able to design a needfinding process as part of the broader design life cycle.
* Students will be able to perform individual needfinding activities.
* Students will be able to synthesize the results of those needfinding exercises into an understanding of the user’s needs.

## Assessments

* Students will reflect on the application of the lesson’s concepts to their chosen area of HCI.
* Students will engage in a short design task based on the lesson’s concepts.
* Students will complete a short answer assignment in which they critique a provided interface from the perspective of the lesson’s concepts.
* Students will complete a short answer assignment in which they select an interface to critique from the perspective of the lesson’s concepts.
* Students will complete a short answer assignment in which they design a revision of one of the critiqued interfaces from the perspective of the lesson’s concepts.

## Lesson Plan

* Students will initially be introduced to the idea of needfinding as the way of gathering foundational user information.
* Students are then introduced to general needfinding concerns: defining the problem space, gathering a data inventory, articulating user types.
* Students will then be walked through various needfinding techniques, generally sorted from minimum intervention with users (naturalistic observation) to maximum intervention (interviews and think alouds).
* Then, students will be introduced to techniques for synthesizing the results of these needfinding methods.
* Finally, students will be instructed how to combine the data they’ve gathered into requirements.

# Script

## 3.3.1 Introduction

### 3.3.1.1 Headshot Studio

* [C] David talking
* [A] Clips of lesson
* [B] Topic; Needfinding
* The first stage of the design life cycle is **needfinding** or requirements gathering.
* This is the stage where we go and try to find out what the user really needs.
* The biggest mistake a designer can make is jumping to the design process before understanding the user or the task.
* We want to develop a deep understanding of the task they’re trying to accomplish and why.
* As we do this, it’s important to try to come in with as few preconceived notions as possible.
* There’s an old adage that says: when all you have is a hammer, everything looks like a nail.
* This is similar.
* If you come in having already decided what approach you want to take, it’s tempting to only see the problem in terms of the approach you’ve chosen.
* So, we’re going to go through a process that attempts to avoid as many preconceived notions as possible.
* [B] Topic; Defining questions about the user
* We’re going to start by **defining** some general questions we want to answer throughout the data gathering process about who the user is, what they’re doing, and what they need.
* [B] Topic; Generating answers about the user
* Then, we’ll go through several methods of **generating** answers to those questions to gain a better understanding of the user.
* [B] Topic;
* Then, we’ll talk about how to **formalize** the data we gather into a shareable model of the task and a list of requirements for our ultimate interface.
* Note that each of these tool could get a lesson on its own on how to do it, so we’ll try to provide additional resources to read further on the tools you choose to use.

## 3.3.2 Data Inventory

### 3.3.2.1 Headshot Studio

* [C] David talking
* [A] Items to find appearing on right
* Before we start our needfinding exercises, we also want to enter with some understanding of what data we want to gather.
* These are questions we want to ultimately answer.
* That’s not to say we should be answering them every step of the way: rather, we want to gather the data necessary to come to a conclusion at the end.
* There are lots of inventories of the types of data you need to gather, but here is one useful list:
* [B] Blooping the first questions on each line as they’re read
* 1. **Who** are the users? What are their ages, genders, levels of expertise?
* 2. **Where** are the users? What is the environment?
* 3. **What** is the context of the task? What else is competing for users’ attention?
* 4. **What** are their goals? What are they trying to accomplish?
* 5. **Right** now, what do they need? What are the physical objects? What information do they need? What collaborators do they need?
* 6. **What** are their tasks? What are they doing physically, cognitively, socially?
* 7. **What** are the subtasks? How do they accomplish those tasks?
* When you’re designing your needfinding methods, each thing you do should match up with one or more of these questions.

## 3.3.3 The Problem Space

### 3.3.3.1 Headshot Studio

* [C] David talking
* In order to do some real needfinding, the first thing we need to do is identify the problem space.
* Where is the task occurring?
* What else is going on?
* What are the user’s explicit and implicit needs?
* We’ll talk about some of the methods for doing this in this lesson, but before getting into those methods, we want to understand the scope of the space we’re looking at.
* Consider the difference between these two actions.

### 3.3.3.2 David’s House (Security System)

* <David turns off the security system>
* <David unlocks the front door holding a bag of groceries and Lucy, opens it, puts the bag down and reaches to turn the alarm off>

### 3.3.3.3 Headshot Studio

* Notice that in each of these, I’m doing the same task: turning off the alarm.
* But in the first scene, we’re very narrowly focusing on the interaction between the user and the interface.
* In the latter, we’re taking into consideration a broader view of the problem space.
* We could zoom out even further if we wanted, and ask questions about where and why people need alarm systems in the first place?
* That might lead us to things like designing security systems for dorm rooms or designing check-in systems for office buildings.
* As we’re going about needfinding, we want to make sure we’re taking the broad approach: understanding the entire problem space in which we’re interested, not just focusing narrowly on the user’s interactions with an interface.
* So, in our exploration of methods for needfinding, we’re going to start with the most authentic types of general observation, then move through progressively more targeted types of needfinding.

## 3.3.4 User Types

### 3.3.4.1 The Park

* [C] David talking on a bench
* Just as we want to get an idea of the physical space of the problem, we also want to get an idea of the space of the user.
* In other words, we want to understand who we’re designing for.
* This comes in a lot when doing design alternatives and prototyping, but we also want to make sure to gather information about the full range of users for whom we’re designing.
* So, let’s take the example of designing an audiobook app for people that exercise.
* Am I interested in audiobooks just for kids, or for adults, too?
* Am I interested in experts at exercising, or novices, too?
* Am I interested in experts at listening to audiobooks, or am I interested in novices at that as well?
* Those are pretty key questions:
* They differentiate whether I’m designing for business people who want to be able to exercise while reading, or exercisers who want something else to do while exercising.
* The task is similar for both, but the audience, their motivations, and their needs are different.
* So, I need to identify these different types of users, and perform needfinding exercises on all of them.
* One of the most successful products of all time succeeded because of its attention to user types.
* The Sony Walkman became such a dramatic success because they identified different needs for different types of people, designed their product in a way that it could meet all those needs, and then marketed it to those types individually.
* [B] Book information (title and authors)
* You can read more about that in a book called Doing Cultural Studies by Hugh Mackay and Linda Janes.

## 3.3.5 5 Tips: Avoiding Bias in Needfinding

### 3.3.5.1 Headshot Studio

* [C] David talking
* [B] Tips blooping
* During needfinding, there are some significant considerations that need to be made to avoid biasing your results.
* Let’s go through five of these possible biases.
* **1. Confirmation bias.** Confirmation bias is the phenomenon where we see what we want to see. We enter with some preconceived ideas of what we’ll see, and we only notice those things that confirm our prior beliefs. Try to avoid this by specifically looking for signs that you’re wrong, by testing your beliefs empirically, and by involving multiple individuals in needfinding.
* **2. Observer bias.** When we’re interacting directly with users, we may subconsciously bias them. We might be more helpful, for example, with users using the interface we designed than one others designed. On surveys, we might phrase questions in a way that elicits the answers we want. Try to avoid this by separating experimenters with motives from the participants, by heavily scripting interactions with users, and by having someone else review your interview scripts or surveys for leading questions.
* **3. Social desirability bias.** People tend to be nice. People want to help. If you’re testing an interface and the participants know you were the designer, they’ll want to say nice things about it to make you happy. That gets in the way of getting good data, though. Try to avoid this by hiding what the socially desirable response it, by conducting more naturalistic observations, and by recording objective data.
* **4. Voluntary response bias.** Studies have shown that people with stronger opinions are more likely to respond to optional surveys. You can see this often in online store reviews: the most common responses are 5s and 1s. For us, that means if we perform quantitative analysis on surveys, we risk oversampling the more extreme views. Avoid this by limiting how much of the survey content is shown to users before they begin the survey, and by confirming any conclusions with other methods.
* **5. Recall bias.** Studies have shown that people aren’t always very good at recalling what they did, what they thought, or how they felt during an activity they completed in the past. That can lead to misleading and incorrect data. Try to avoid this by studying tasks in context by having users think aloud during activities s or conducting interviews during the activity itself.
* These biases can largely be controlled by making sure to engage in multiple forms of needfinding

## 3.3.6 Naturalistic Observation

### 3.3.6.1 The Park

* [C] David talking
* For certain tasks, a great way for us to start understanding the user’s needs is to simply… watch.
* A great way for me to start understanding this is to just come somewhere where people are exercising and watch.
* This is naturalistic observation: observing people in their natural contexts.
* So, I’m sitting here in the park, just watching people.
* I want to start with very specific observations, then generalize out to more abstract tasks.
* That way, I’ll avoid confirmation bias, which basically say you see what you want to see.
* So, what do I notice?
* Well, I notice there are a number of different types of exercising.
* I see walkers, joggers, runners.
* I see some rollerbladers.
* I see some people doing yoga.
* I see some people riding bikes.
* I see bikers that seem to be a bit more leisurely, I see bikers that seem to be more strenuous.
* I’m noticing that while joggers might be able to stop and start pretty quickly, that’s harder for someone riding a bike, so I might want to avoid designs that force the user to pull out their phone a lot because that’s going to be dangerous for bikers.
* I see people exercising in groups, but also people exercising individually.
* I don’t actually know right now whether people exercising in groups would be interested in this, so I’m going to need to find that out later.
* I see that many people stretch before and after exercising.
* I’m wondering if we can use that. Maybe we can have some kind of starting and ending sequence for this, so that a single exercise session is actually a session with its own notes as opposed to the app just kind of being always on.
* By just watching people engaged in this task, I’m gathering an enormous amount of information that might affect my design.
* But note also that while naturalistic observation is great, I’m limited in that ethically, I can’t interact with users directly or capture any identifying information like videos or photographs: the only thing I can take are notes.
* I’m also limited in that I don’t know anything about what those users are thinking.
* I don’t know if people working out in groups want to be able to listen to audiobooks.
* I don’t know if bluetooth headsets would be problematic for people doing yoga.
* I need to do a lot more before I get to the design phase.
* However, this has been very informative in understanding the problem space.

## 3.3.7 5 Tips: Naturalistic Observation

### 3.3.7.1 Headshot Studio

* [C] David talking
* [B] Tips blooping
* Here are five quick tips for doing naturalistic observation.
* **1. Take notes.** Don’t just sit around watching for a while; be prepared to gather targeted information and observations about what you see.
* **2. Start specific, then abstract.** Write down the individual little actions you see people doing before trying to interpret or summarize them. If you jump to summarizing too soon, you risk tunnel vision.
* **3. Spread out your sessions.** Rather than sitting somewhere for two hours one day and moving on, try to observe in shorter 10-15 minute sessions several times. You may find interesting different information, and your growing understanding and reflection on past exercises will help your future sessions.
* **4. Find a partner.** Observe together with someone else. Take your own notes, then compare them later so you can see if y’all interpreted the same scenarios or actions the same way.
* **5. Look for questions.** Naturalistic observation should inform the questions you decide to ask participants in more targeted needfinding exercises. You don’t need to have all the answers based on observation alone: what you need is questions to investigate further.

## 3.3.8 Participant Observation

### 3.3.8.1 The Park

* [C] David, in exercising attire
* Sometimes it’s not just enough to watch people engaging in a task.
* Sometimes, we want to experience the task for ourselves.
* So, that’s what I’m going to do.
* I listen to audiobooks a lot, but I really don’t exercise.
* I should, but I don’t.
* But I’m going to try this out.
* I’ve got my audiobook queued up.
* I’m going to go on a ride and see what I discover.

### 3.3.8.2 The Park

* [C] David, in exercising attire
* So, what did I learn?
* Well, I learned that I’m out of shape for one thing.
* I also learned that this app would be quite valuable for anyone doing participant observation on exercisers because I kept having to stop to record notes for myself… which I could have done if the app was already implemented.
* But aside from that, I noticed that unexpected things happened pretty often that made me wish I could easily go back.
* Or, sometimes there were just things I wanted to hear again, but there was no easy way to do that.
* I noticed that the need is definitely there for me -- I already plan to listen to everything again when I get home because there were notes I wanted to take that I couldn’t take easily.
* I also noticed that while sometimes I want to take notes, sometimes I also just want to leave bookmarks.
* Now, we do have to be careful here.
* [B] Davidface; You are not your user!
* Remember, you are not your user.
* When you’re working as a participant observer, you can have useful insights, but you shouldn’t overrepresent your own experiences.
* You should use this experience to inform what you ask users going forward.

## 3.3.9 Hacks and Workarounds

### 3.3.9.1 David’s House (Office)

* [C] David talking in his office chair
* Let’s zoom in a little bit more on what the user actually does.
* While we can do naturalistic and participant observation without having a lot of direct interaction with users, we need to get inside users’ heads a little more to understand what they’re thinking and doing.
* If we’re trying to design interfaces to make existing tasks easier, one way to research that is to look at the hacks users presently employ.
* How do they use interfaces in non-intended ways to accomplish tasks?
* Or, how do they break out of the interface to accomplish a task that could be accomplished within the interface?
* If you’re designing a task that’s meant to be performed at a desk, looking at the person’s workspace can be a great way of accomplishing this.
* I have six monitors around, and yet you still see post-it notes everywhere.
* How could I possibly need more screen real estate?
* Well, post-its can’t be covered up.
* They don’t take away from the existing real estate.
* They’re visible even when the computer is off.
* So, using post-it notes here is a hack to get around the limitations of the interface.
* When you’re looking at hacks, it’s important to note just look at what a user does and assume you understand why.
* Look at the workarounds they’re employing and ask them why.
* Find out why they don’t use the interfaces currently in place.
* You might find they just don’t know about them, which presents a different design challenge.
* Hacks are related to another method we can use to uncover user needs: errors.
* Whereas hacks are ways users get around the interface to accomplish their tasks, errors are slips or mistakes users frequently make while performing the task within the interface.

## 3.3.10 Errors

### 3.3.10.1 David’s House (Office)

* [C] David talking in the chair
* When we’re trying to make iterative improvements, one of the best places we can look is at the errors users make with the tools they currently have available.
* We can fix those errors, but we can also use those errors to understand a bit more about the user’s mental model.
* So, here’s a common example of an error, a slip, for me.
* I keep my email open on this window over here.
* I’ll frequently forget that it’s my active window while trying to type into a window over here.
* As a result, I’ll hit a bunch of hotkeys in my email interface.
* I’ll tag random emails, delete random emails, and more.
* This is a slip because there’s nothing wrong with my mental model of how this works, the problem is that I can easily forget which window is active.
* Mistakes on the other hand, a place where my mental model is weak and I’m more prone to mistakes is when I’m using my Mac.
* I’m used to a PC, where the maximize button always makes a window take up the entire screen.
* I’ve honestly never fully understood the maximize button on a Mac.
* Sometimes it seems to work like a PC maximize button.
* Sometimes it just expands the window a bit, but not to the entire screen.
* Sometimes it enters something like a PC’s full screen mode, even hiding the top task bar.
* I make mistakes there because I don’t have a strong mental model of how it works.
* So, if you were watching me, you could see me making these errors, and you could ask me why I’m making them.
* That works for both discovering hacks and discovering errors: watch people performing their task, and ask them about why certain things happen the way that they do.

### 3.3.10.2 The Park

* [C] David talking on a bench
* Discovering hacks and errors involves a bit more user interaction than just observing people in the wild.
* How might we go about that if we’re doing things like developing apps for people acting in public?
* Maybe we actually go up to people we see exercising out in public, although that’s always going to be a little awkward, and the data we get might not be great.
* So at this point, we might be better off recruiting people to come in and describe their experience.
* People experience hacks and errors pretty consciously, so our best bet would likely be to target exercise groups or local areas where exercisers frequent and recruit people to come in for a short study.
* Or, maybe we could recruit people to participate in a study during their normal exercise routine, taking notes on their experience or talking us through their thought process.
* We could take that to an extreme and actually adopt something like an apprenticeship approach, where we train to become users.

## 3.3.11 Apprenticeship and Ethnography

### 3.3.11.1 Udacity Office (Atlanta)

* [C] David talking
* If we’re designing interfaces for particularly complex tasks performed by experts, we might find that no amount of observation or short interaction is enough to gain a full understanding of the way the task works.
* For complex tasks, you almost have to become an expert in order to design effective interfaces to be used by experts.
* This is informed by the domain of ethnography, which focuses on living very close to the users you’re studying.
* You bring in your expertise with design and HCI, but you have to personally develop the domain knowledge necessary to design new interfaces or improve the user task.
* So, to take an example, our video editors here at Udacity have a complex workflow from recording to combining footage to editing to adding in the final effects.
* We’re exploring the recording process in our running design challenge, but even after that’s done, there’s a lot of work that goes on in here.
* I don’t think I’d have any chance of understanding their task well enough to improve it without learning a lot more about it myself.
* <Fast-motion montage of David learning to edit>
* So, that was fun.
* I now understand a little more about the editing process, and my background in HCI pointed out some things that could be easier.
* Granted, to actually make those things easier, I need to get a job at Adobe.
* But if I had a job at Adobe, I’d now have some good knowledge for creating a highly specialized set of tools for use by a couple dozen Udacity employees.

## 3.3.12 Interviews and Focus Groups

### 3.3.12.1 Headshot Studio (Morgan)

* [C] David and Morgan talking
* Our most targeted way of gathering information from users, though, is just to talk to them.
* One way of doing that might be to bring them in for an interview.
* So, I’m sitting here with Morgan, one of the potential users for an audiobook app targeted at exercisers.
* We’re especially interested in the kinds of tasks you want to engage with while exercising and listening to audiobooks.
* So, in your experience, what kinds of challenges do you run into?
* [Morgan] Well, I think the most significant challenge is just that it’s hard to control it.
* [Morgan] Mine has a button on the headphones that you can use, but it just pauses and plays.
* [Morgan] If I want to rewind or something I have to stop, pull out my phone, unlock it, just to move it around.
* Thank you!
* Interviews are useful ways to get at what the user is thinking when they’re engaging in a task.
* You can do interviews one on one like this, or you can even do interviews in groups with multiple users.
* Those tend to take on the form of focus groups, where a number of people are all discussing something at the same time, and they can tease out different kinds of information.
* Focus groups can elicit some information and thoughts that wouldn’t come from 1:1 interviews, but they also present the risk of overly convergent thinking, so they should be used in conjunction with interviews and other needfinding methods.

## 3.3.13 5 Tips: Interviews

### 3.3.13.1 Headshot Studio

* [C] David talking
* [B] Tips blooping
* Here are five quick tips for conducting effective interviews. We recommend reading more about this before you actually start interviewing people, but these are just to get you started.
* **1. Focus on the six W’s** in writing questions: Who, what, where, when, why, and how? Try to avoid questions that lend themselves to one-word or yes-or-no answers: those are better gathered via surveys. Use your interview questions to ask open-ended, semi-structured questions.
* **2. Be aware of bias.** Look at how you’re phrasing your questions and interactions and make sure you’re not predisposing the participant to certain views. If you only smile when they say what you want them to say, for example, you risk biasing them to agree with you.
* **3. Listen.** Many novice interviewers get caught up in having a conversation with the participant rather than just gathering data from the participant. Make sure the participant is doing the vast majority of the talking, and don’t reveal anything that might predispose them to agree with you.
* **4. Organize the interview.** Make sure to have an introduction phase, some lighter questions to start to build trust, and a summary at the end so the user understands the purpose of the questions. Be ready to push the interview forward or pull it back on track.
* **5. Practice!** Practice your questions on your friends, family, or research partners in advance. Rehearse the entire interview. Gathering subjects is tough, so when you actually have them, make sure you’re ready to get the most out of them.

## 3.3.14 Exercise: Interviews

### 3.3.14.1 Tablet Studio

* [V] Exercise text
* Interviews are likely to be one of the most common ways you gather data, so let’s run through some good and bad interview questions real quick.
* Here are six questions.
* Which of these would make good interview questions?
* Mark the ones that would be good.
* For the ones that would be bad, briefly brainstorm a way to rewrite the question to make it better.

### 3.3.14.2 Exercise

* [E] “Which of these would make good interview questions?”
* [E] A. Do you exercise? [F]
* [E] B. How often do you exercise?
* [E] C. Do you exercise for health or for pleasure? [F]
* [E] D. What, if anything, do you listen to while exercising?
* [E] E. What device do you use to listen to something while exercising?
* [E] F. We’re developing an app for listening to audiobooks while exercising. Would that be interesting to you? [F]

### 3.3.14.3 Tablet Studio

* [V] Exercise text
* <<run through answers>>

## 3.3.15 Think-Aloud

### 3.3.15.1 The Park

* [C] David talking
* Think-aloud protocols are similar to interviews in that we’re asking users to talk about their perceptions of the task.
* However, with think-aloud, we’re asking them to actually do so in the context of the task.
* So, instead of bringing Morgan in to answer some questions about listening to audiobooks while exercising, I’ll ask her to actually think outloud while listening to audiobooks and exercising.
* If this was a different task, like something on a computer, I could have her just come into my lab and work on it, but since this is out in the world, what I might do is just give her a voice recorder to record thoughts she has while out running and listening.
* Think-aloud is very useful because it can help get at user’s thoughts that they forget when they are no longer engaged in the task.
* It’s also a bit dangerous, though: by asking people to think outloud about their task, we encourage to think about it more deliberately. That can change what they actually do.
* So, while it’s useful to get an understanding of what they’re thinking, we should check to see if there are places where what they do differs when thinking out loud about it.
* We can do that with a post-event protocol, which is largely the same, except we wait to get the user’s thoughts until immediately after the activity.
* That way, the activity is still fresh in their minds, but the act of thinking about it shouldn’t affect their performance as much.

## 3.3.16 Surveys

### 3.3.16.1 Headshot Studio

* [C] David talking
* Most of the other methods for needfinding, like observation, interviewing, apprenticeship, require a significant amount of effort for what is often relatively little data.
* We might spend an entire hour interviewing a single possible user, or an hour observing a small handful of users in the world.
* The data we get from those interactions is deep and thorough, but sometimes we also want broader data.
* Sometimes we want to know just how many people encounter a certain difficulty or engage in a certain task.
* If we’re designing an audiobook app for exercisers, maybe we just want to know how often those people exercise. Maybe we want to know what kind of books they listen to.
* At that point, a survey might be our most appropriate means of needfinding.
* Surveys let us get a much larger number of responses quickly, and the questions can be phrased objectively, allowing quicker interpretation.
* Plus, with the internet, they can be administered asynchronously.
* A few weeks ago, I came up with an idea for a study on Friday morning, and with cooperation from our local IRB office, I was able to send out the survey to potential participants less than 24 hours later and receive 150 responses within a week.
* Of course, the data I receive from that isn’t nearly as thorough as what I would receive from interviewing some of those participants, it’s somewhat superficial, but it’s a powerful way of getting a larger amount of data, and it can be especially useful to decide what to ask participants during interviews or focus groups.

## 3.3.17 5 Tips: Surveys

### 3.3.17.1 Headshot Studio

* [C] David talking
* [B] Tips blooping
* Survey design is a well-documented art form. In fact, designing surveys is very similar to designing interfaces themselves, so many of the lessons we learn in our conversations apply here as well.
* Here are five quick tips for designing and administering effective surveys:
* **1. Less is more.** The biggest mistake novice survey designers make is to ask way too much. That affects the response rate and reliability of your data. Ask the minimum number of questions necessary to get the data that you need, and only ask questions you know you’ll use.
* **2. Be aware of bias.** Look at how you’re phrasing the questions: are there positive or negative connotations? Are participants implicitly pressured to answer one way or the other?
* **3. Tie them to the inventory.** Make sure every question on your survey connects to some of the data that you want to gather. Start with the goals for the survey and write the questions from there.
* **4. Test it out!** Before sending it to real participants, have your coworkers or colleagues test out your survey. Pretend they’re real users, and see if you would get the data you need from their responses.
* **5. Iterate!** Survey design is like interface design. Test out your survey, see what works and what doesn’t, and revise it accordingly. Give participants a chance to give feedback on the survey itself so that you can improve it for future iterations.

## 3.3.17A Writing Good Survey Questions

### 3.3.17A.1 Tablet Studio

* Surveys are often used because of their convenience, but they’re only useful if the questions are actually well-written.
* Tips like ‘be aware of bias’ and ‘test it out’ are good pieces of general advice, but there are also some specific things that can make your survey questions better.
* There are six things I personally recommend in survey design, so let’s go through them.
* The six general words of advice are: be clear, be concise, be specific, be expressive, be unbiased, and be usable.
* First, for clarity: want to make sure the user actually understands what we’re asking about.
* If we’re using a numeric scale, we want to provide labels that explain what the scale means.
* If we’re providing ranges, we want to avoid overlapping ranges so the user isn’t confused about what to select.
* Sometimes, we might want to provide additional detail to ensure that the question is correctly interpreted.
* Part of being clear is also make sure that the user has a clear foundation for answering the question.
* If we’re asking about the frequency of a task, we probably want to time-box it so they aren’t reflecting on an impossibly long period of time.
* Second, be concise.
* Ask questions in plain language.
* Note that conciseness is sometimes at odds with clarity: sometimes we add extra details to make a question more clear, but that harms the conciseness of the question.
* Use your best judgment on when more information improves clarity enough to justify the loss of conciseness.
* Third, be specific.
* Avoid questions about super-big ideas; instead, break those questions down into smaller, specific questions that get at that big idea.
* Avoid what are called “double barrel” questions: these are questions that ask about two things at once.
* Avoid questions where the user could have conflicting ideas at the same time: if they could have conflicting ideas about different parts, break that into multiple short questions.
* Fourth, be expressive -- or perhaps that should be, allow the user to be expressive.
* Most of your survey questions will be asking your users for their opinions, so emphasize in the question prompt that they’ll be providing an opinion -- this makes the user more comfortable giving their thoughts.
* When providing opinion ranges like ‘agree’ vs. ‘disagree’, always provide an odd number of options so that the user can respond neutrally, and I’d generally recommend at least 5 options so users feel more comfortable differentiating their level of agreement.
* When asking a multiple-choice question, if there’s a chance a user could have more than one thought, let them choose more than one.
* Instead of asking yes or no questions, give the user a range of options from which to choose to add more dimension to their expression.
* Fifth, be unbiased -- we mentioned already the need to avoid bias, but how do you do that exactly?
* One way is the same as allowing the user to be expressive: giving an ‘other’ option limits bias towards your pre-selected options.
* You should also note that users will often read your options and take them as suggestions. What a user selects may not be what they would have said if you hadn’t given them options, so it can be beneficial to make questions like these free-response.
* Watch for leading questions.
* Watch for loaded questions.
* Be careful with closed-ended questions: provide an ‘Other’ option
* And sixth, be usable.
* This refers less to the questions you ask and more to the way you design your survey as a whole: use HCI principles in designing the survey itself.
* Some of these will be dictated by the survey tool you use, but others are up to you.
* Provide a progress bar so that the user can evaluate how far they are into the survey.
* Make the pages approximately consistent lengths so that the user has an accurate gauge for what it means to be on page 3 of 5 or something similar.
* Order your questions logically: group questions about demographics, questions about prior experience, questions about future desires, etc. such that they follow a natural flow.
* Alert users when questions are unanswered, but don’t require them to be answered: some users will feel uncomfortable answering some questions, so it’s good to leave them unrequired, but you also want to avoid users unknowingly skipping. So, tell them if they’ve skipped, but don’t force them to answer.
* And finally, preview the survey yourself. Your users might not tell you if there are errors, so make sure to take it yourself.
* That was a lot of information, but hopefully it’s all practical enough to be immediately applicable -- and when in doubt, you can always ask for feedback!

## 3.3.18 Exercise: Surveys

### 3.3.18.1 Tablet Studio

* [V] Exercise text
* Writing survey questions is an art.
* Let’s take a look at an intentionally poorly-designed survey and see everything we can find that’s wrong with it.
* On the left is a survey. Write down in the box on the write everything that is wrong with this survey.

### 3.3.18.2 Exercise

* [E] Likert-style survey questions with: even number of options; leading questions; changing scales; yes/no questions.

### 3.3.18.3 Tablet Studio

* [V] Exercise text
* <run through answer>

## 3.3.19 Other Data Gathering Methods

### 3.3.19.1 Headshot Studio

* [C] David talking
* So far, we’ve discussed some of the more common approaches to needfinding.
* Depending on your domain, though, there might be other things you can do.
* [B] Existing UI evaluation
* First, if you’re designing for a task for which interfaces already exist, you might start by critiquing the interfaces that already exist using some of the evaluation methods we cover in the Evaluation lesson.
* For example, if you wanted to design a new system for ordering take-out food, you might evaluate the interfaces of calling in an order, ordering via mobile phone, or ordering via a web site.
* [B] Product Reviews
* Second, and similarly, if you’re trying to develop a tool to address a problem people are already addressing, you might go look at user reviews and see what people already like and dislike about existing products.
* For example, there are dozens of alarm clock apps out there, and thousands of reviews. If you wanted to design a new one, you could start there to find out what people need.
* [B] Data logs
* Third, if you’re working on a task that already involves a lot of automatic logging, like web surfing, you could try to get some logs of user interaction that have already been generated.
* For example, say you wanted to build a browser that’s better at anticipating what the user will want to open next. You could grab data logs and look for trends both within and across users.
* You can get creative with your data gathering methods. The goal is to use a variety of methods to paint a complete picture of the user’s task.

## 3.3.20 Exercise: Needfinding Pros and Cons

### 3.3.20.1 Tablet Studio

* [V] Exercise
* In this lesson, we’ve covered various different methods for needfinding.
* Each method has its advantages and disadvantages.
* Let’s start to wrap the lesson up by exploring this with an exercise.
* Here are the methods we’ve covered, and here are some potential pros and cons.
* For each row, mark the column to which that pro or con applies.
* Note that these might be somewhat relative, so your answer may differ from ours.

### 3.3.20.2 Exercise

* [E] Exercise: columns featuring ‘Naturalistic Observation’, ‘Participant Observation’, ‘Errors and Hacks’, ‘Interviews’, ‘Surveys’, ‘Focus Groups’, ‘Apprenticeship’, ‘Think-Aloud’
* [E] Row: “Analyzes data that already exists.”
* [E] Row: “Requires no participant recruitment.”
* [E] Row: “Requires no synchronous participation between participant and experimenter.”
* [E] Row: “Investigates the participant’s thought process.”
* [E] Row: “Occurs in the actual task context.”
* [E] Row: “Cheaply gathers data from lots of participants.”

### 3.3.20.3 Tablet Studio

* [V] Exercise
* <run through answers>

## 3.3.21 Design Challenge: Needfinding for Book Reading

### 3.3.21.1 Headshot Studio

* [C] David talking
* The needfinding exercises we’ve gone through focus on the needs of exercisers.
* What can they do with their hands, what is the environment around them like while exercising, and so on.
* However, that’s only half the picture for this particular design.
* Our ultimate goal is to bring the experience of consuming books to people while they exercise, which means we also need to understand the task of book-reading on its own.
* Now, our problem space is still around exercisers, so we wouldn’t go through the entire design life cycle for book reading on its own. We don’t need to design or prototype anything for them.
* But if we’re going to bring the full book-reading experience to people while exercising, we need to understand what that is.
* So, take a moment and design an approach to needfinding for people reading on their own.

### 3.3.21.2 Exercise

* “Click to continue” exercise

### 3.3.21.3 Headshot Studio

* [C] David talking
* We could apply pretty much every single needfinding method we’ve discussed to this task.
* We could go to the library and watch people reading, see how they’re taking notes.
* We’ve all likely done it ourselves, and we can reflect on what we do while reading -- although we need to be careful not to overvalue our own priorities and approaches.
* Reading is common enough that we can easily find participants for interviews, surveys, think-alouds.
* The challenge here will be deciding who our users really are.
* Books are ubiquitous. Are we trying to cater to everyone who reads deliberately?
* If so, we need to sample a wide range of users.
* Or, we could choose a subset.
* We might cater to students studying, or busy businesspeople, or people that specifically walk or bike to work.
* We might start with one of those groups and then abstract out over time.
* We might eventually abstract all the way out to just anyone who is unable to read and take notes the traditional way, like people driving cars or people with visual impairments -- but that’s down the road.
* The most important thing is that we define who our user is, define the task in which we’re interested, and deliberately design for that user and that task throughout the design life cycle.

## 3.3.22 Iterative Needfinding

### 3.3.22.1 Tablet Studio

* [V] Design life cycle
* We’ve noted that design is a life cycle from needfinding to brainstorming design alternatives to prototyping to evaluation, then back to needfinding.
* Needfinding on its own, though, is an iterative cycle.
* [V] Needfinding web
* We might use the results from our naturalistic observations to inform the interviews that we design.
* Those interviews then might inform the surveys we create to sample a larger audience.
* The results of those naturalistic interviews and surveys then might dictate what further naturalistic observation we might want to perform.
* We now might understand things about the task or users that we didn’t understand previously, so we can go back to naturalistic observation with a keener eye for what we want to observe.
* One source to this is even the evaluation process itself.
* Creating prototypes and evaluating them gives us data on what works and what doesn’t, and that may inform what we want to observe to better understand the task.

## 3.3.23 Revisiting the Inventory

### 3.3.23.1 Headshot Studio

* [C] David talking
* During these needfinding exercises, you’ll have gathered an enormous amount of data about your users.
* Ideally, you’ve combined different sets of these approaches as well. You’ve observed people performing the tasks, you’ve asked them about their thought process, you’ve tried it out some yourself.
* Pay special attention to some of the places where the data seem to conflict: are these cases where you as the designer understand some of the elements of task design that the users don’t? Or are these cases where your expertise hasn’t quite developed to the point of understanding the task?
* Once you’ve gone through the data gathering process, it’s time to revisit that inventory of things we wanted to gather initially.
* [B] Inventory items on the right (first question only)
* 1. **Who** are the users? What are their ages, genders, levels of expertise?
* 2. **Where** are the users? What is the environment?
* 3. **What** is the context of the task? What else is competing for users’ attention?
* 4. **What** are their goals? What are they trying to accomplish?
* 5. **Right** now, what do they need? What are the physical objects? What information do they need? What collaborators do they need?
* 6. **What** are their tasks? What are they doing physically, cognitively, socially?
* 7. **What** are the subtasks? How do they accomplish those tasks?

## 3.3.24 Representing the Need

### 3.3.24.1 Tablet Studio

* [V] Thumbnails of various representations
* Now that you have some understanding of the user’s needs, it’s time to try to formalize that into something we can use in design.
* There are a number of different ways we can do this.
* [V] Example task outline appears
* For example, maybe we create a step-by-step task outline of the user engaging in some task.
* We can break those tasks down into sub-tasks as well.
* That helps us focus on places where we might be able to solve a task through different steps.
* [V] Hierarchical network appears
* We can further develop this kind of task outline into a hierarchical network.
* This might involve more complexity than simply a linear series of actions.
* [V] Structural diagram
* We might further augment this with a diagram of the structural relationships amongst the components in the system and how they interact.
* [V] Flow chart appears
* From there, we might develop this even more into a flow-chart equipped with decision-making points or points of interruptions.
* Notice how these representations are very similar to the outcomes of the task analyses we talk about in the principles unit of our conversations.
* We can similarly use the data gathered from here to summarize a more comprehensive task analysis that will be useful in designing and prototyping our designs.

## 3.3.25 Defining the Requirements

### 3.3.25.1 Headshot Studio

* [C] David talking
* [B] List of types of requirements appearing on one side, examples appearing on the other
* Finally, the final step of needfinding is to define our requirements.
* These are the requirements that our final interface must meet.
* They should be specific and evaluateable, and they can include some components that are outside the user’s tasks as well, as defined by project requirements.
* In terms of user data, we might have requirements regarding:
* [B] Blooping requirements
* **Functionality**: what the interface can actually do.
* **Usability**: how certain user interactions must work.
* **Learnability**: how fast the user can start to use the interface.
* **Accessibility**: who can use the interface.
* We might also have some that are generated by external project requirements.
* **Compatibility**: what devices the interface can run on.
* **Compliance**: how the interface protects user privacy.
* **Cost**: how much the final tool will cost.
* And so on.
* We’ll use these to evaluate the interfaces we develop going forward.

## 3.3.26 Design Challenge: Needfinding and MOOC Recording

### 3.3.26.1 Headshot Studio (Behind Camera)

* [C] David talking
* <<to be scripted>>

## 3.3.27 Exploring HCI: Needfinding

### 3.3.27.1 Headshot Studio

* [C] David talking
* How might needfinding work in your chosen area of HCI?
* If you’re looking at designing for some technological innovation like augmented or virtual interactions, the initial phase might not actually be that different.
* Your goal is to understand how people perform tasks right now, without your interface, so initially you’ll want to observe them in their naturalistic environment.
* Later, though, you’ll need to think about bringing participants to you to experience the devices first-hand.
* If you’re interested in something like HCI for healthcare or education, you have a wealth of naturalistic observation available to you.
* You might even have existing interfaces doing what you want to do, and you can try to leverage those as part of your needfinding exercises.
* Remember, no matter your area of application, you want to start with real users -- that might be observing them in the wild, talking to them directly, or looking at data they’ve generated.

## 3.3.28 Conclusion

### 3.3.28.1 Headshot Studio

* [C] David talking
* [A] Clips of the lesson on the right
* [B] Topic; Needfinding
* Today we’ve talked about **needfinding**.
* Needfinding is how you develop your understanding of the needs of your user.
* What tasks are they completing?
* What is the context of those tasks? What else is going on?
* What are they thinking during the task? What do they have to hold in working memory?
* All these things feed into your understanding of the user’s needs.
* [B] Topic; Needfinding techniques
* We’ve discussed a number of different **techniques** to approach this, ranging from low intervention to high intervention.
* [B] Topic; Naturalistic observation
* [B] Topic; Participant observation
* On the low side, we can just **observe** our users in the wild, or we can **become** users ourselves and participate in the task.
* [B] Topic; Errors and hacks
* Working up, we can try to look more closely at users’ areas to find **errors** or hacks, or peruse the data that they’re generating.
* [B] Topic; Surveys, interviews, and focus groups
* We might interact with them directly through **surveys**, interviews, or focus groups.
* [B] Topic; Ethnography
* Or, we might choose to **work** alongside them, not just participating in the task independently but learning from them and developing expertise itself.
* Once you’ve gained a sufficient understanding, it’s time to move on to the next step: design alternatives.