Introduction

Welcome

Hello, I'm Chris Nodder. Welcome to the second episode of the UX Design Techniques series. In this episode we'll discuss Analyzing User Data. I'll show you how to learn what you're user's pain points are, by watching them perform the tasks you care about in their own environment. I'll talk about observation techniques, how to analyze the data you collect, and how to extract actionable product ideas from this analysis. Now, it's time to dive in and discuss the techniques you can use for gathering and analyzing data on your users. So let's get started.

---

1. **Data Analysis in the User-Centered Design Process**

Get data to analyze

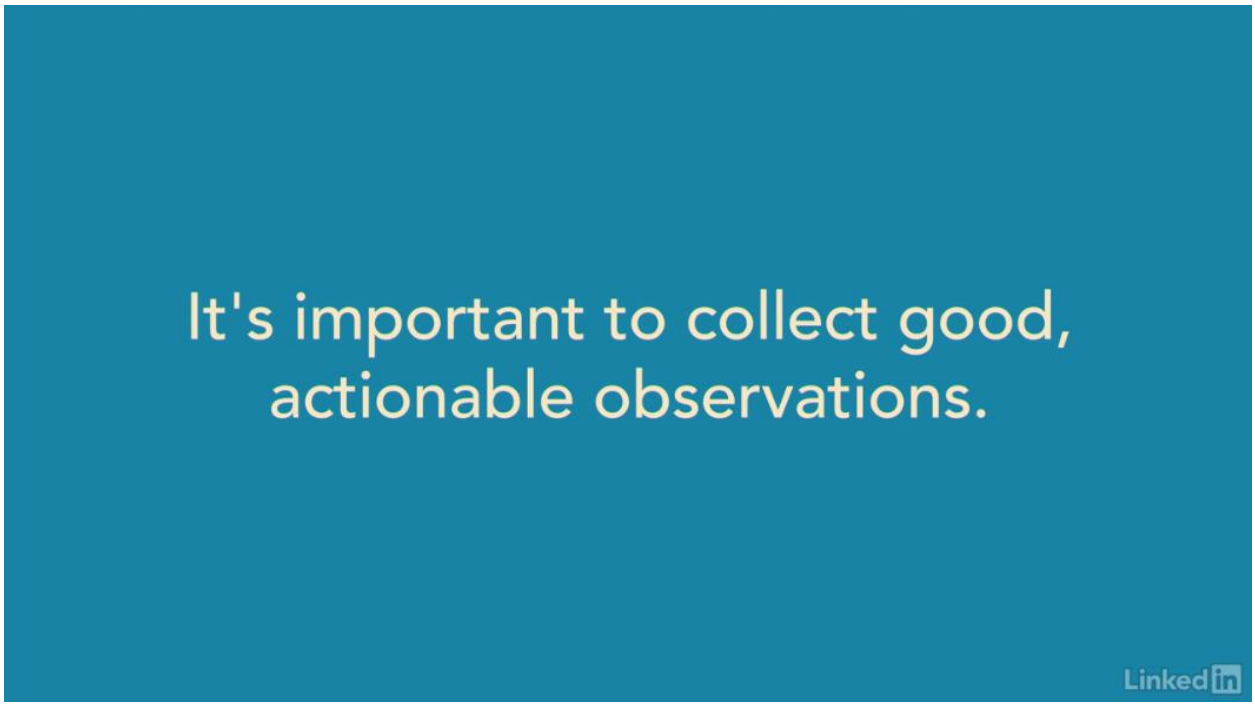
You are **not the same**
as your users.

Understanding Your Users

- Who are your users?
- What are their pain points?
- What areas of your product need attention?

The core concept of user-centered design is that giving your customers software that meets their needs is more likely to make them productive and happy. If they're more productive and happy, they'll use your product more and recommend it to more people. That, in turn, makes your product more successful. So, the first task in the user-centered design process is to understand more about your users. User research can be quantitative, that is, using statistics and metrics you gather from web analytics or instrumentation or from surveys and other market research, or it can be qualitative. That means the rich data you get from observing users performing the tasks you care about whether it's manually, or with your software, or even using the competitor's product. It's best to have a combination of quantitative and qualitative data. Quantitative data tells you what is happening. Qualitative data tells you why it's happening. Combining the two types of user research allows you to see in-depth behavior during user observations research, and then work at how prevalent that behavior is using quantitative tools. Or quantitative analytics let you see where in your product users are struggling. Qualitative techniques let you work out the reasons so you can best resolve the issues. Without a good set of user research, it's unlikely you'll be able to do user-centered design for the simple reason that you won't know how to be user-centered. If there's one thing that's always true on development teams, it's that you are not your users. Anyone working in software development is more technically adept and better able to complete system tasks than regular users. The first step in being user-centered is to get an understanding of who your users truly are and what their pain points are. What I normally suggest, is that you take the quantitative data that you have from analytics, instrumentation, and support calls, and use that to work out what area of your product you want to first try and address. Then, perform some qualitative user research by watching some representative users working with that part of the product. The best way to get your qualitative information is to take a field trip to users' place of work or their homes to watch them working. The data you get back is much richer in these situations than if you were to ask users to come to you to do the same tasks, because you get to see all the little coping mechanisms they use, like passwords stuck on Post-it notes under their keyboards, or cheat sheets stuck on the side of their monitors. Those things are missing if you bring people into an artificial usability testing environment. User-centered design is also a whole-team activity. Unless everyone on the team has a good understanding of user needs, they won't be able to design good solutions for those needs. Get the whole team to participate in user research, so that they're all aware of the pain points that users have.

Turn observations into actionable data



The part of user-centered design that most people struggle with is making sense of user quotes and actions. How do you get from site visit data to something you can use to make a better product? This course describes the techniques you can use to extract user pain points, goals, and product ideas from observation data. The beauty of this process is that it quickly provides a visualization of where users face problems in their current work. It also describes what those problems are. In a way that you can turn into goals for software improvements you'd like to make in your next product release. Describing users' current work, their behaviors, and their feelings in this level of detail also shows how much the tools they have today differ from their ideal situation. We'll use this understanding of the gap between their current tools and what they really need as we move through the user-centered design process. As we move forward, we'll start creating ideal solutions based on the understanding we gain from observing users' situation today. So it's important to collect good, actionable observations from our target audience in order to drive the rest of the design process. Without a good understanding of users' current pain points, we won't be able to create goals for product improvement, and we won't know whether the product ideas we come up with are likely to resolve the issues that users truly face in their daily lives.

Data: The key to user-centered designs

This is the starting point of the user centered design process. Any user centered design activity needs user data as an input. You'll take this data and process it so that you can extract users pain points. Fixing those pain points will mean that users are more likely to love the product you create. To do that you'll first define exactly who you're building for using personas. Then you'll use ideation techniques to generate multiple different solutions to the pain points you discover. After ideation you'll bring things back to reality by creating scenarios which you'll then use to build a prototype user interface that contains your suggested fixes. You'll usability test that UI to see whether your ideas work with your target audience. And then you'll create a plan to build this code to make the interface real. All of this process relies on you first collecting good quality data from representative users performing representative tasks. So let's look at how we should go about collecting that information.

---

2. Gathering User Data

Observe users

Why Observe Users?

- Asking people what's wrong gives you incomplete answers.
- Observing them lets you see the issues first hand.
- Knowing the issues, you can develop full solutions.

Planning Your Observations

- Who are your target user types?
- Where do they do their work?
- What type of activity do you want to observe?

How Many Observations?

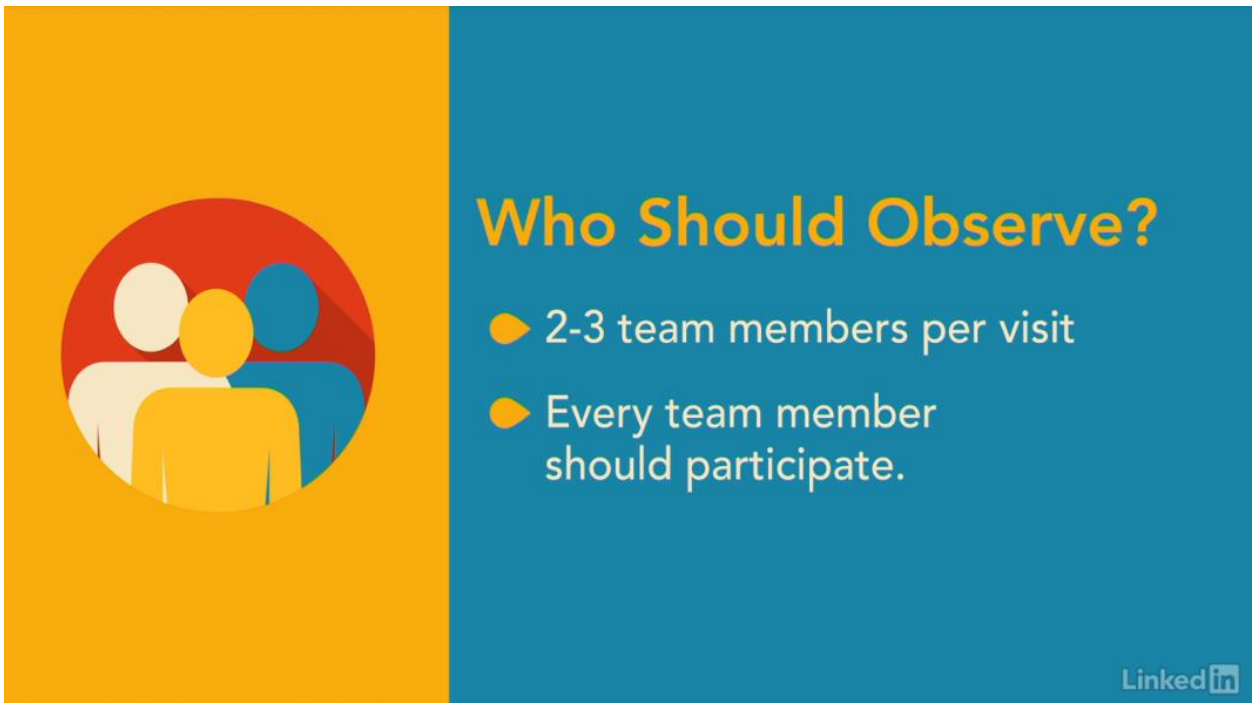
- 3+ visits **for each target user type**
- 5+ visits **in total**

More realistic

Less realistic

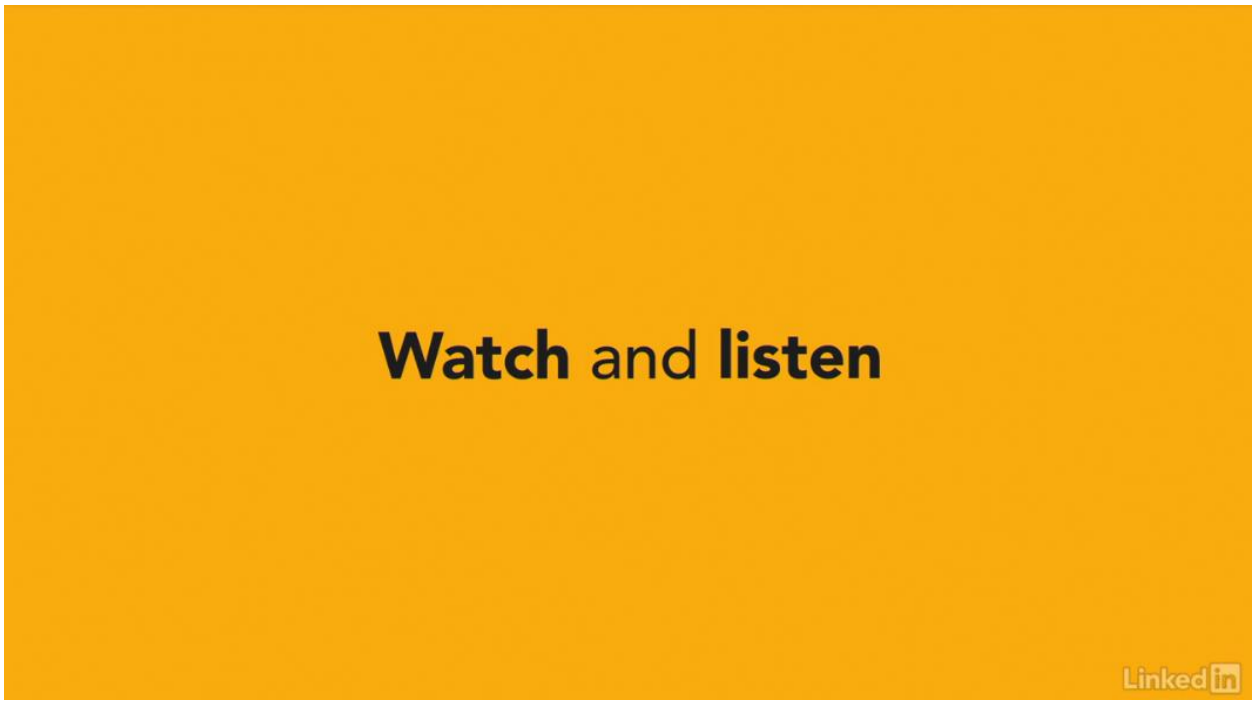
Where to Observe?

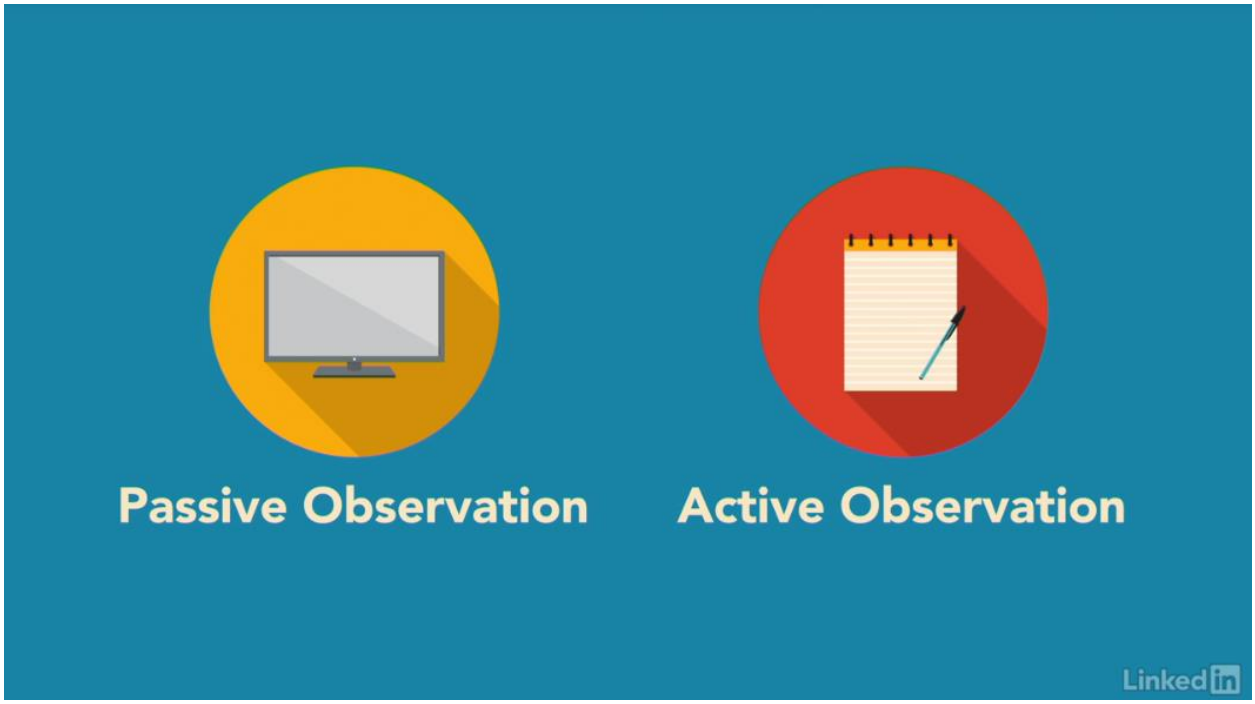
- User works from their own location.
- User brings items to your location.
- Behavioral interview with user



Unfortunately, users aren't very good at vocalizing their tasks, and they've often become so worn down by the annoyances they face that they don't realize there could be an alternative. So, if you directly ask users what you should fix, or worse still, how you should fix it, they'll give you answers that are incomplete and don't touch on the root cause of the problem. Instead, go and watch them. Yes, it's that simple. Going on just a few site visits shows you the real pain that users face everyday. There are however some guiding principles to follow so that you get the most out of your visits. First, work out who you care about. Who is your target user? You need a well-defined user in order to have a well-defined product. Now, arrange to visit some people who match your target user profile. You must visit them in the place where they perform the activity you care about. You'll need to arrange at least three different visits for each target user type you uncover. You should perform at least five visits in total. This way, you'll meet users with different skill levels and so you'll be more likely to observe different workflows and coping strategies. It's important to visit users when they'd normally perform the tasks you want to watch. If that's impossible, ask them to save the tasks up for you, but remember that you're less likely to see their true behavior. For instance, if you want to see mobile phone behavior, you may have to follow someone on their commute to work or at lunchtime, rather than visiting them while they sit at their desk. Going to visit users at their own location works best, because of the richness of the data you can collect. If that's not possible, ask them to bring relevant things with them to your location so that you can see something as close as possible to their true behavior. Interviewing is less beneficial than watching behavior. If you must interview, make sure it's behavioral, focused on peoples' actual behavior in the past rather than forward looking, asking people to talk about things they've never actually done. Two or three team members should go on each visit. Having two people makes it easier to find the place you're looking for, makes it safer to visit people in their homes, and makes it easier to take good notes. More than three people is likely to get crowded in almost any environment. It's better to do more visits than to take more people on each visit.

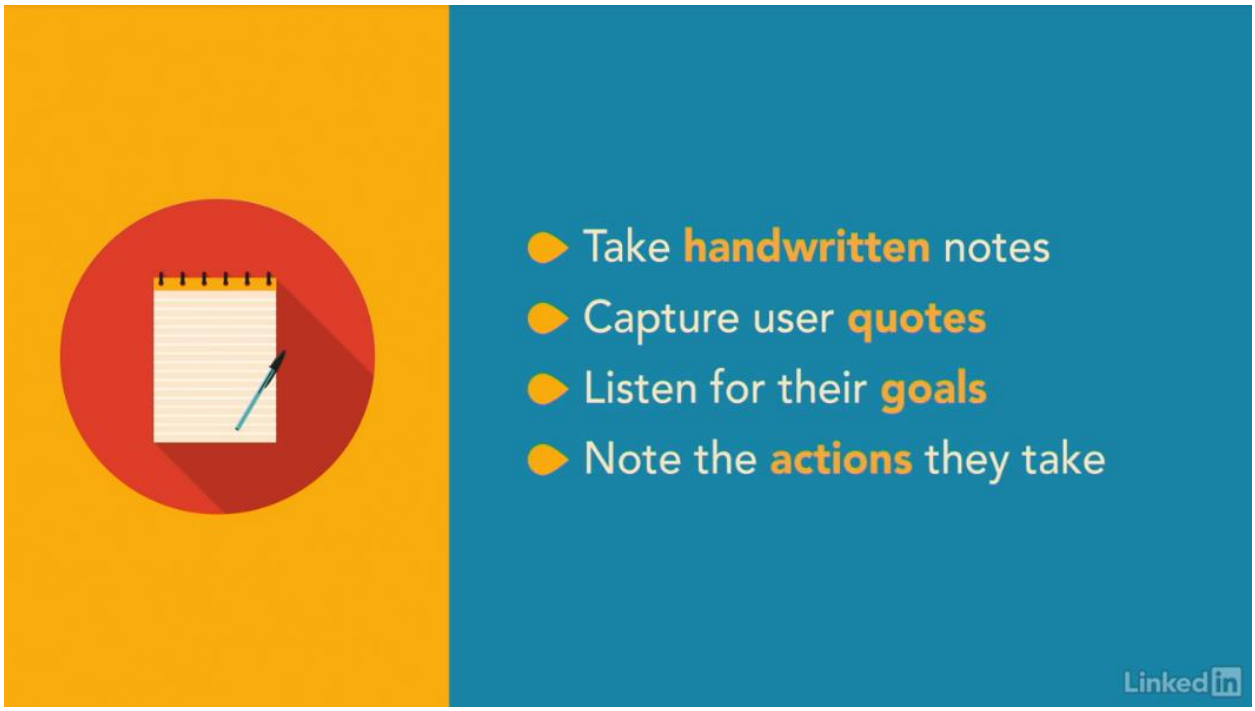
Understanding good observation techniques





The goal of the observation is to gather as much information about how users approach the task you care about as possible. Any time you spend talking or interrupting the participant is time when you aren't gathering data. The best thing to do is let the participant get on with their life while you watch. You might ask them to think out loud as they're working. Your role is to be an active observer. The only tools you need for a site visit are a notebook and a pen, and a roll of duct tape. The notebook and pen are for taking notes and sketching the environment you see users working in. Don't record video of the interaction. Video recording changes how users react to you. Also, you'll never have time to watch the tapes. Video recording also makes you lazy. You feel like the video recorder is watching so you don't have to. Also, don't use a laptop for note taking. It's too intrusive, and it's a barrier between you and your participant. The duct tape is for your mouth. Yes, you need to remember that you're an observer not an interviewer. After the introductions, do your best to be a quiet apprentice. Ask the user to describe what they're doing and feel free to ask clarifying questions, but try to ensure that 90% of the words come from them, not from you. If you're engaged in conversation, you're not observing. It's very hard to be a good observer. We don't practice active observation very often. Most of the time, we're passive observers, like when we watch TV. Active observation involves taking copious notes, almost a verbatim transcript of what the participant says and does. If you go in with a superior attitude, after all you probably developed the software they're using, then you start the relationship off in a way that makes it hard for your participant to be open and honest with you. My best tip for you if you're new to observation is to be humble. Take on the role of apprentice to the user. It's okay to ask them to explain things to you to help you understand their actions. My course on usability testing is a good place to learn more about how you should behave while you're interacting with users.

Gather the information

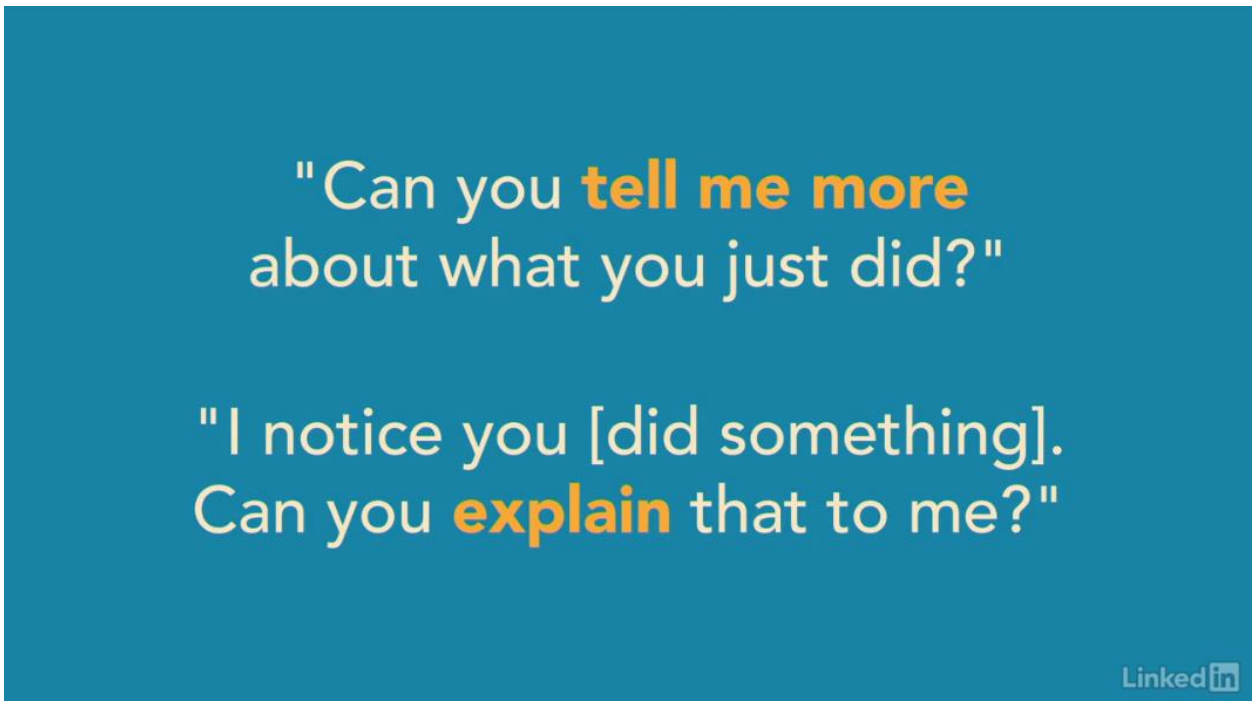


Do

- Write down what people say and do
- Take photos of the environment (if allowed)
- Engage with people
- Ask open-ended questions
- Ask for examples

Don't

- Engage in conversation
- Write down solutions or bug fixes
- Turn the visit into a sales pitch
- Ask participants to predict the future



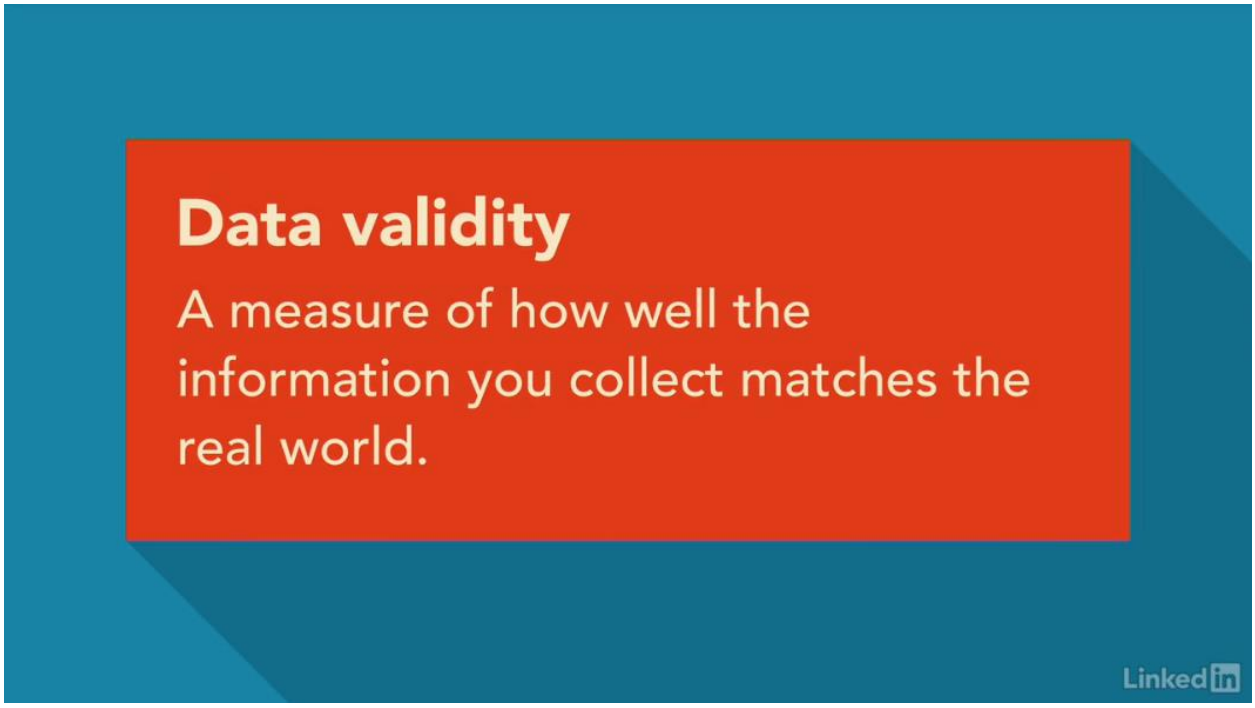
As I've already mentioned, I recommend taking handwritten notes rather than typing them, and I definitely don't recommend video recording. The most important thing to capture is user quotes that sum up the experience. It's also important to listen for users' goals, what they're trying to achieve, and to note down the actions they take towards achieving those goals. Rather than writing, user doesn't like iPhone keyboard, it's better to write the user's quote, like, "I'm used to using Swipe. "This is much slower." That type of data is richer and more believable. Do write down what people say and do. Take photos of the environment if you're allowed to as well. Engage with them. Smile and respond non-verbally, and ask open-ended questions, like, can you tell me more about that? Also ask for examples, times when the thing you care about has happened in the past. However, don't engage in conversation. Don't write down solutions or bug fixes. Remember, if you're doing this, you're distracted. Don't try to sell participants on your cool product ideas, and don't ask them to predict the future. Sometimes you'll see behaviors that confuse you. Wait until you think the user is finished or reaches a good stopping point, and then say something neutral like, can you tell me more about what you just did? Or, I noticed you did something, can

you explain that to me? It's important that you understand the behaviors you observe, but not at the expense of the user's comfort. You aren't there to teach them or to make things better. As soon as you start down this path, you stop being an apprentice and start being an expert in their eyes. By the end of the observation session, your writing hand should be cramping up. It's not uncommon to have multiple pages of notes. Remember that the best notes are facts, not interpretations. Write down what users say and what they do. You can interpret those quotes and actions later. When you're done with each visit, it's nice to ask the person you observed whether they have any questions for you. It's also usual to pay people for their time, either cash, if it's a member of the public, or giftcards if it's someone inside your organization. If you can't pay people, work out what else you can do for them. Maybe there are some marketing doohickeys lying around the office, like T-shirts or mugs, or anything else with a logo on it. Those kinds of things can be a nice way of saying thanks.

---

3. **Experience Mapping**

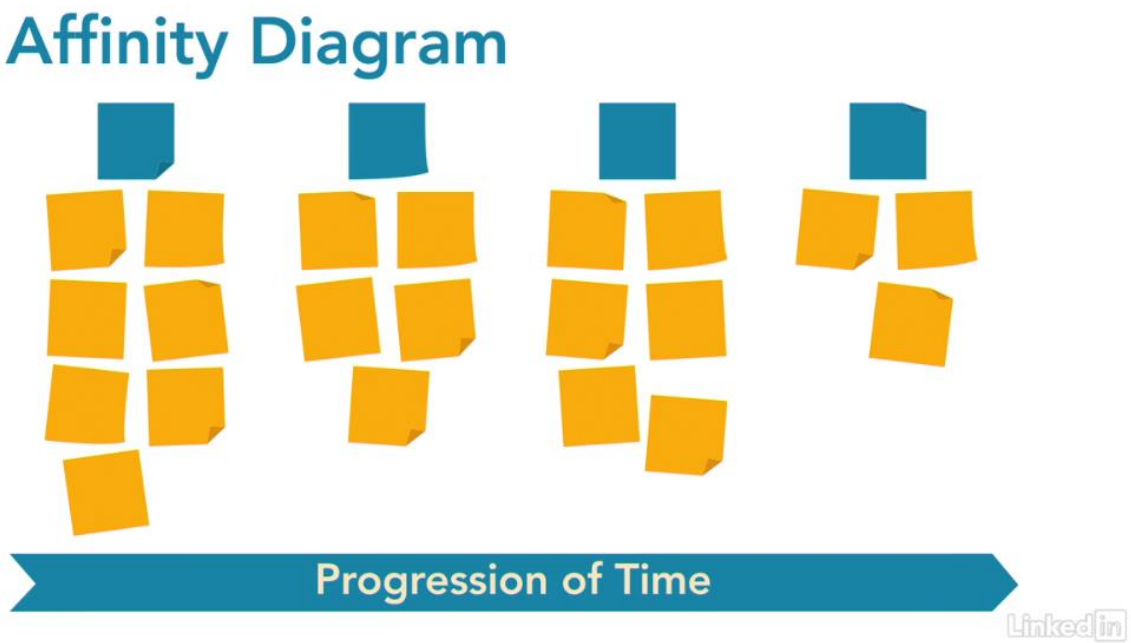
Experience maps depict user interactions

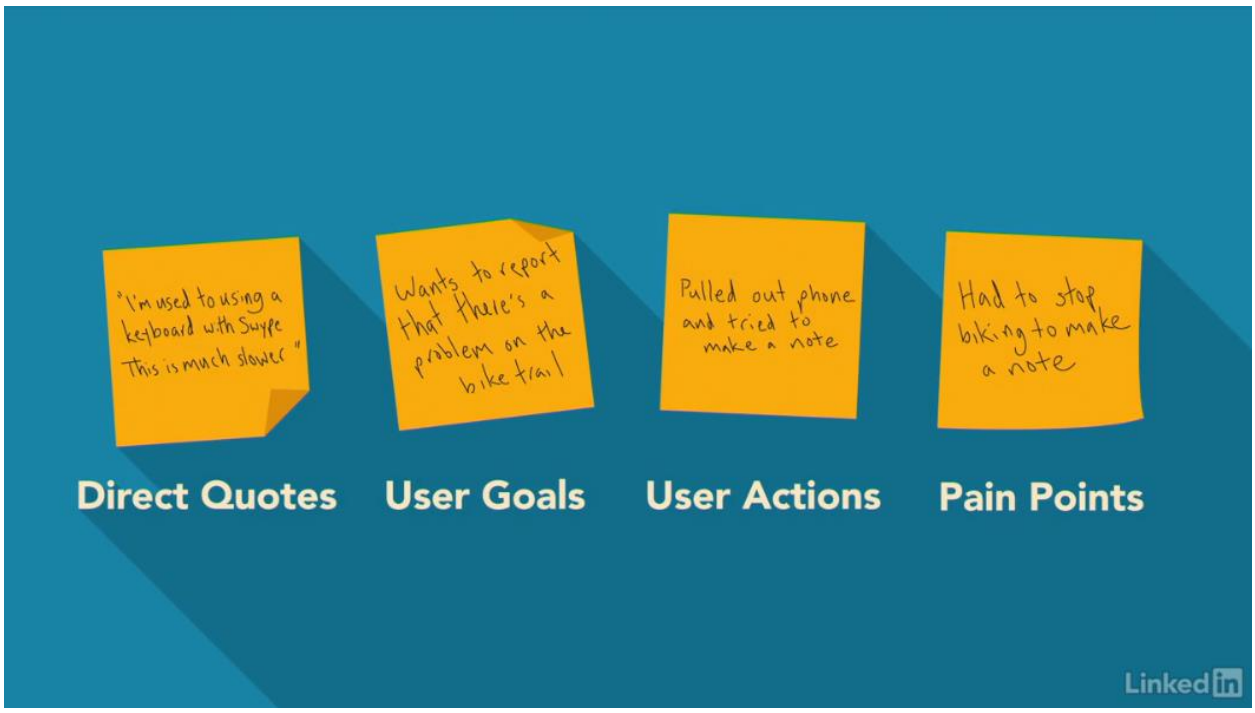


Analyzing qualitative information like written site visit notes isn't as simple as plugging numbers into a spreadsheet. However, it's a lot more fun, and it can lead to some really interesting insights about your users. These insights are the things you use to bulid products that delight your customers. The best way I've found to quickly turn a pile of site visit observations into a visual story about users' tasks and pain points is to use experience maps. Experience maps are affinity diagrams on steroids. After you've observed your users, you create an experience map to extract pain points, goals, and persona information. The experience map is a way of depicting the user's path through their task in a way that highlights the order of the actions and the paint points they experience along the way. The idea is to get all the observations from every visit into one location. Doing this allows everyone on the team to see how the things they observed fit into the big picture. Because different people attend to different site visits, this data merging activity puts everyone literally on the same page. Building the map should be a team activity. Everyone who was on the visits should be in the room at the same time. This is where you find

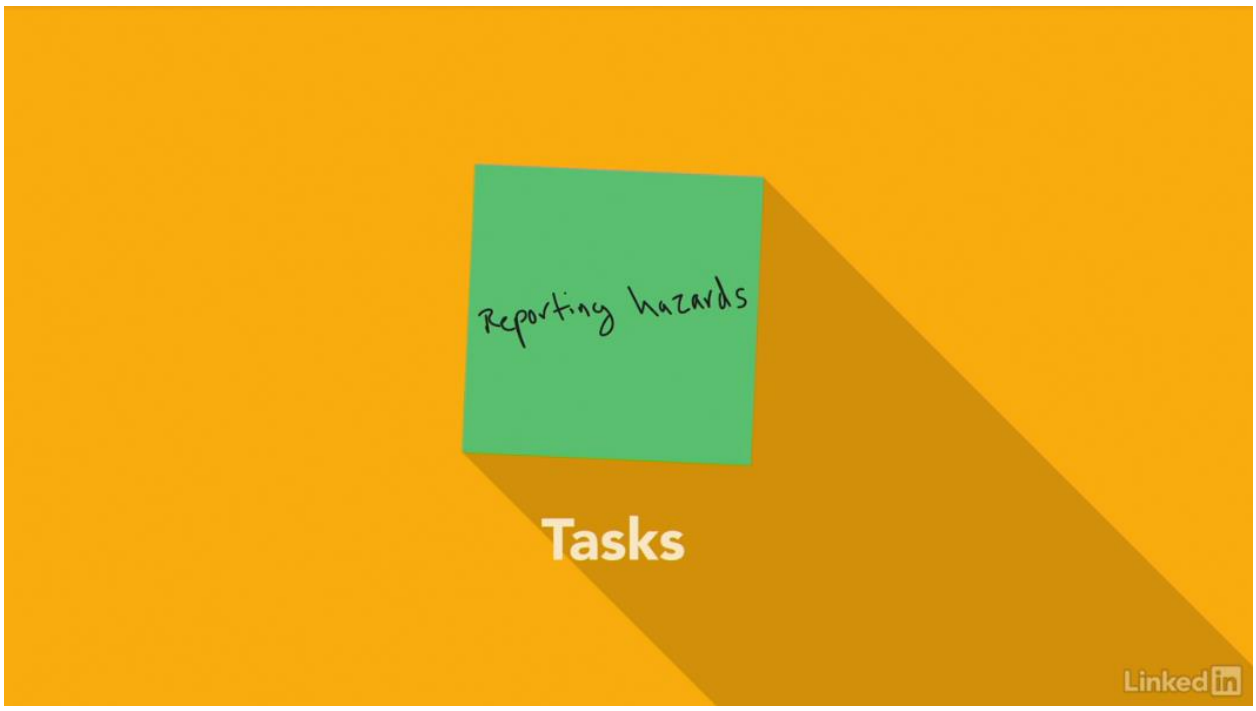
out how good your site visit observation notes were. What you will do is put all the user observations you made onto sticky notes and then arrange those on the wall in groupings that depict each of the stages in your user's tasks. Moving from left to right, you'll be able to see the progression through the task, and the issues that occur at different points in the process, such as when information moves from being paper-based to digital, or moves from a PC to a mobile device. The experience map process is more than just an opportunity to stick notes on the wall. It's also a time for each team member to share their experiences and build a common understanding of user issues. As team members talk through the items that they are placing on the experience map, every other team member is listening, and either relating similar experiences, or saying how their site visits differed. The shared discovery of user issues can really help the team to understand that they are not like their users, and that their users deserve a better product than the current process allows. I want to quickly mention observer bias and data validity here, because it's something that team members often ask about. People on the team who are used to gathering statistical information about users may be upset with how fluffy this exercise looks. However, underneath its fun surface, there is plenty of methodological rigor. Observer bias occurs when people writing down observations place their own interpretation on the activity they are watching. It also happens when people ignore some activities and focus on others instead. The experience map reduces bias by combining multiple observations from different observers, and by ensuring that the map creators focus on the problem, not the solution. Validity is a measure of how well the information you collect matches the real world. Here, you gathered your observations from real people who you know are representative users. You haven't changed that data at all. The stickys on the wall contain direct user quotes and other contextual information. The path you took from observation to reporting of the data is transparent. Next, we're going to look at the process of building an experience map in detail.

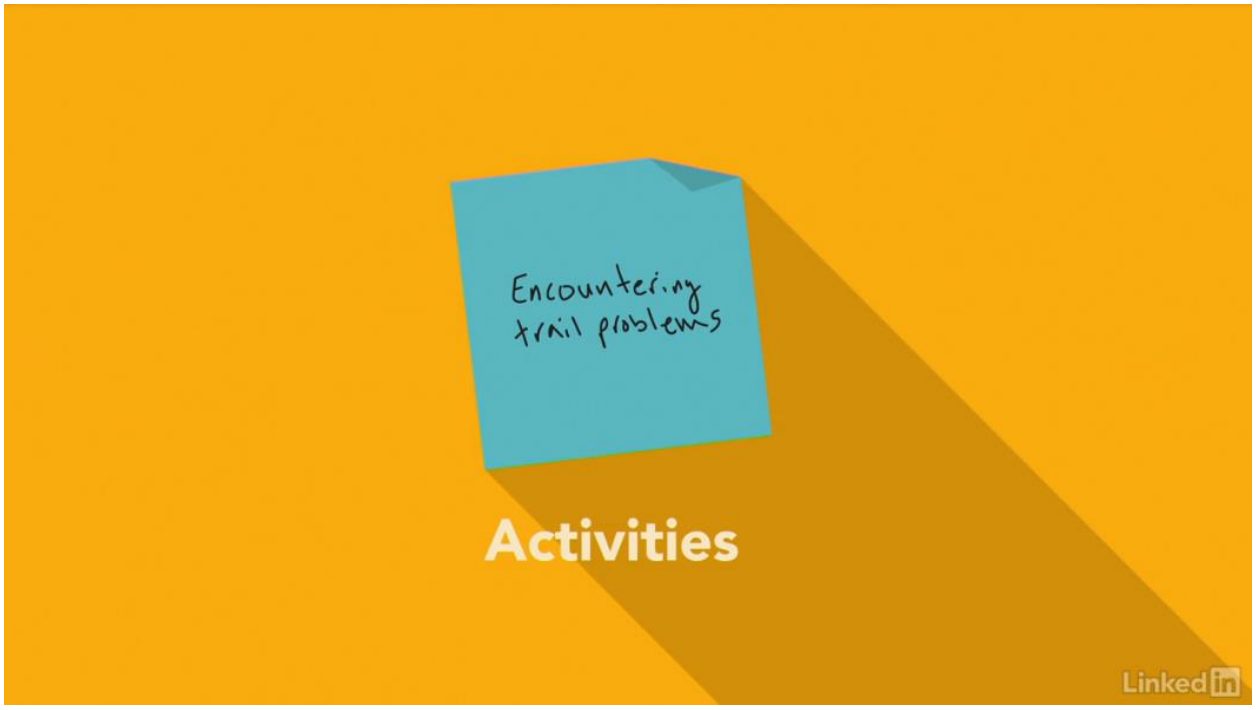
Build an experience map




"I'm used to using a keyboard with Swype. This is much slower."
Wants to report that there's a problem on the bike trail
Pulled out phone and tried to make a note
Had to stop biking to make a note
Direct Quotes
User Goals
User Actions
Pain Points
LinkedIn

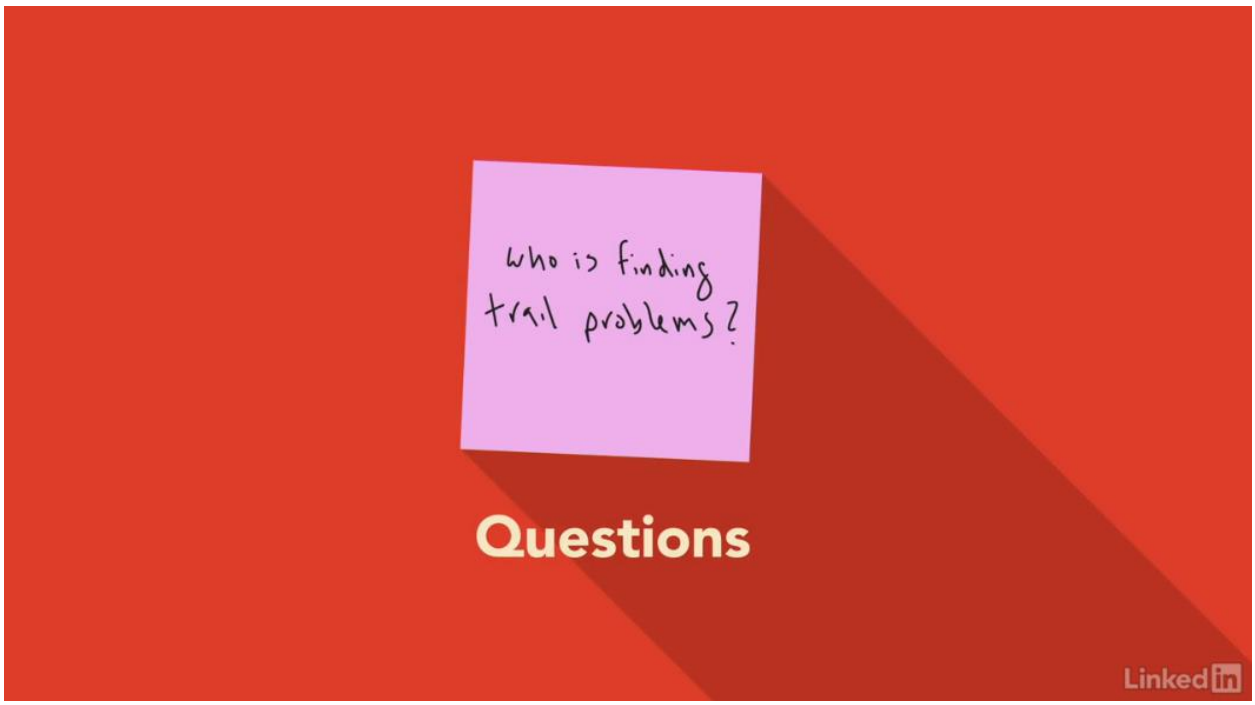

"I leave my phone in the car."
"I track all my rides with my phone."
LinkedIn

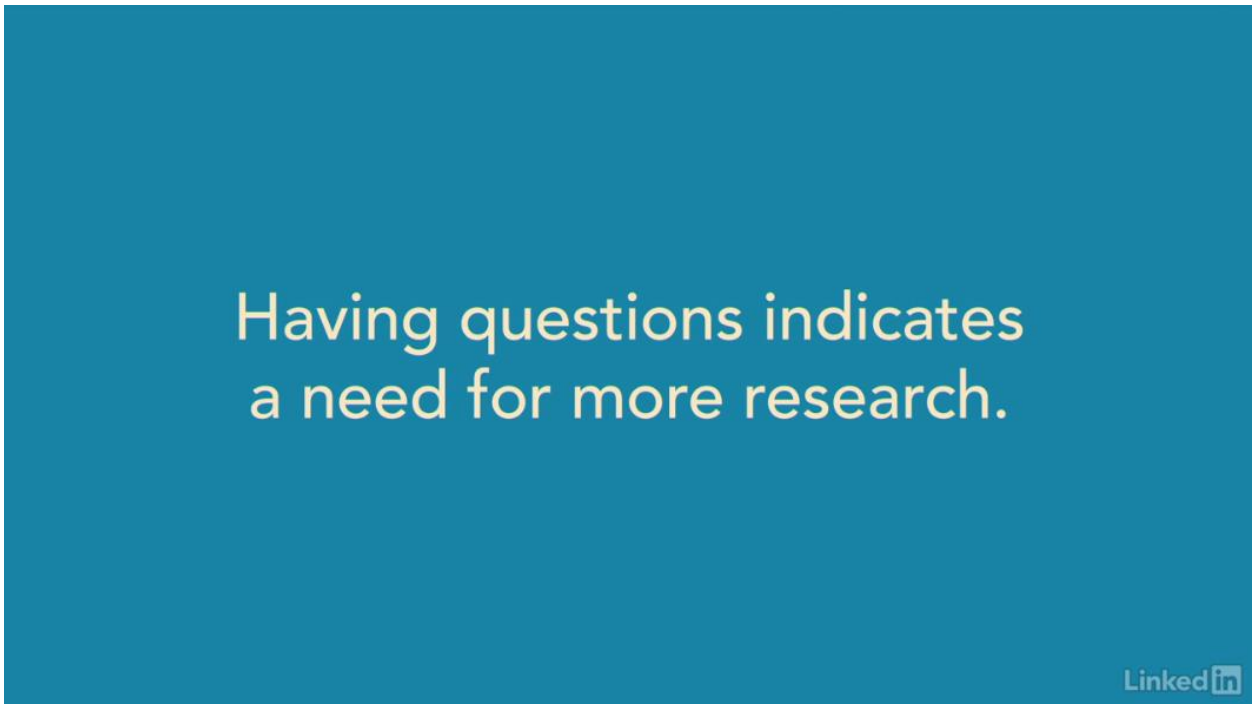

Reporting hazards
Tasks
Linked in

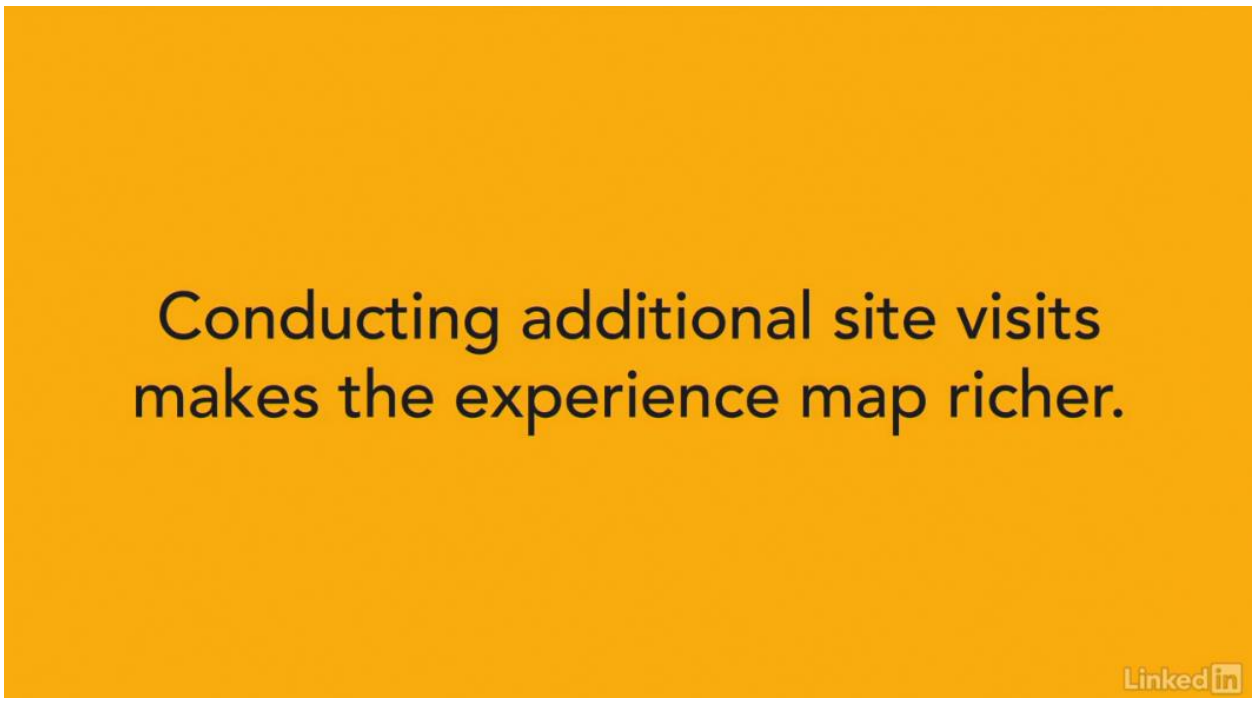

Encountering trail problems
Activities
Linked in


maps of problem areas on trail
Design Ideas
LinkedIn


Design ideas aren't user data;
they're your interpretation.
LinkedIn


who is finding trail problems?
Questions
LinkedIn

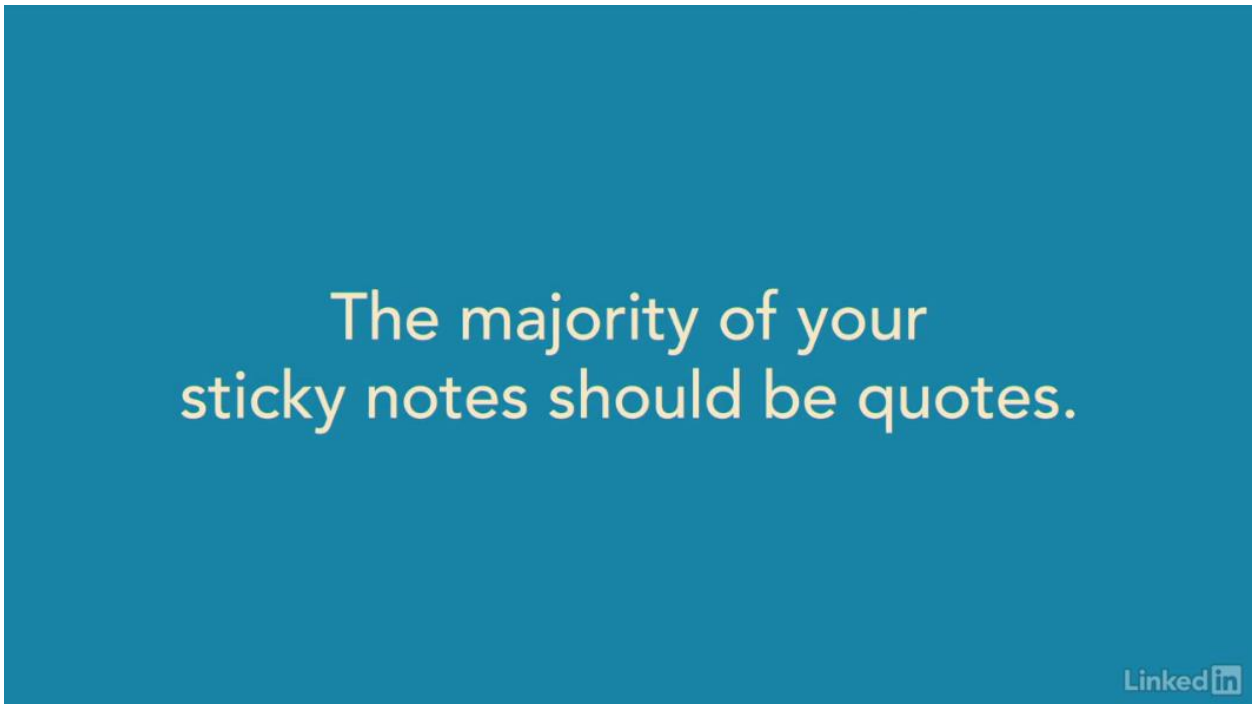

Having questions indicates a need for more research.
LinkedIn

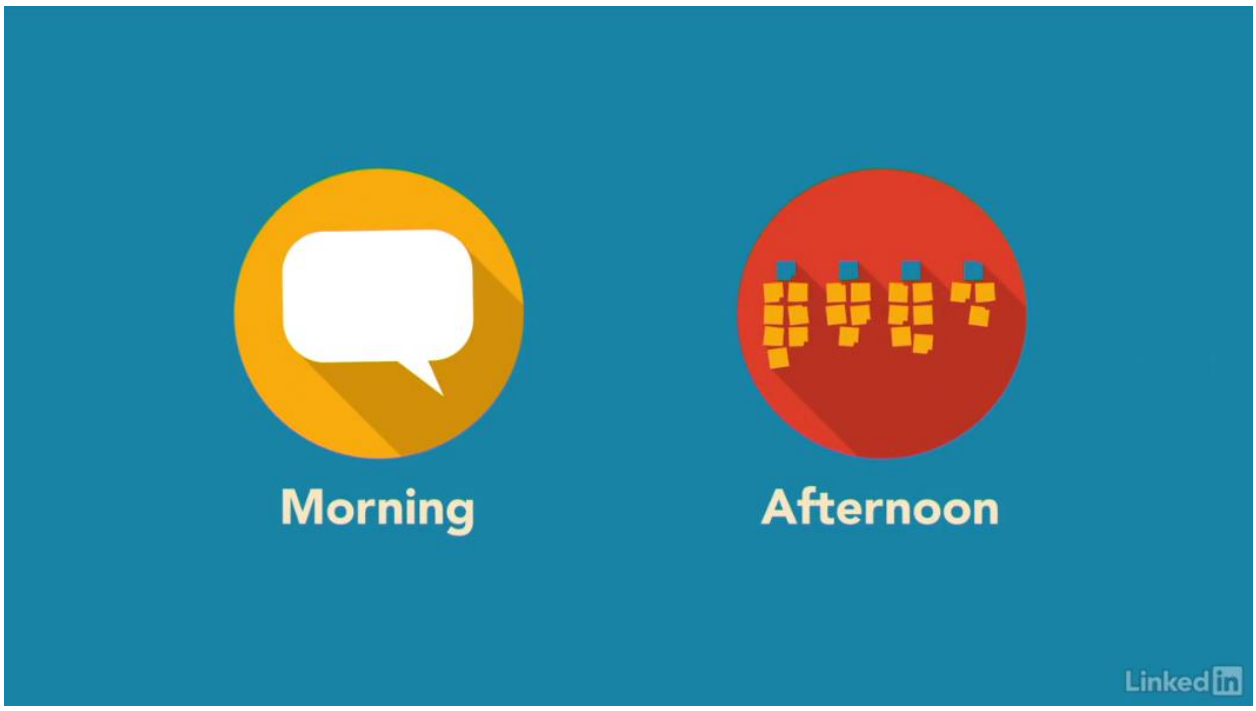


The experience map-making process, is very similar to affinity diagramming. It starts the same way, but then adds the dimension of time to make it clear how users' tasks progress. Everyone who attended a visit, brings their visit notes. They copy each observation from their notebooks onto a sticky note. Offices tend to have more yellow stickies than other colors. So use those for observations. Observations normally fall into one of the following categories. Direct Quotes, which is what the user said. Most yellow stickies on the wall should be direct quotes when you're finished. They are the most powerful kind of data you can share. Then, there are User Goals, what users said they were trying to achieve. Also, User Actions, observations of how they went through achieving what they wanted to. And Pain Points, things that stopped them from doing what they wanted. Start by putting a large sheet of paper up on the wall, so that you can move the map later. I'd suggest using at least five flip chart sheets side by side. Or just use butcher paper, which comes on a long roll. Everyone brings their sticky notes to the sheet of paper on the wall, and each person places their sticky notes on the wall, calling out the item they're sticking up, so that others can put similar items near it. Place observations of tasks that occur early in the process on the left side of the paper, and ones that happen towards the end on the right-hand side. Feel free to move stickies around as you see better groupings. Once all the stickies are up, check their groupings. If there are a lot of sticky notes in one group, see how you could split that group up. If there are some groups with only a couple of sticky notes, could those be combined, or are they truly separate tasks? It's okay if people disagree. In fact, the disagreements highlight where the interesting stuff is. If two people saw different behaviors during different visits, you need to dig in to that area in more detail, so that you can resolve the disagreement. Once all the team members have grouped their sticky notes into tasks, label each group with a green sticky note. The green sticky note should describe the task that the group of observations refers to. It might not use the same language that the people you observed use. It just has to make sense to you. At this point, you should tidy up the experience map. Make sure that all the green tasks are arranged chronologically, with all the yellow observations vertically beneath the task they refer to. The last step is to group sets of tasks into activities. Give each activity a name, using blue stickies. What you should have at this point, is many columns of yellow stickies, arranged chronologically across the page from left to right. Each column having a green sticky title that describes what task the column refers to, and with a blue activity sticky note that describes a set of green tasks. As a development team, it's hard not to come up with design ideas, to resolve the issues you saw during the site visits. That's okay, but don't let these design ideas slow you down. There's plenty of time for design explorations later in the process. Instead, if you think of design ideas, add them on a different colored sticky note. Here, I've used orange, and then

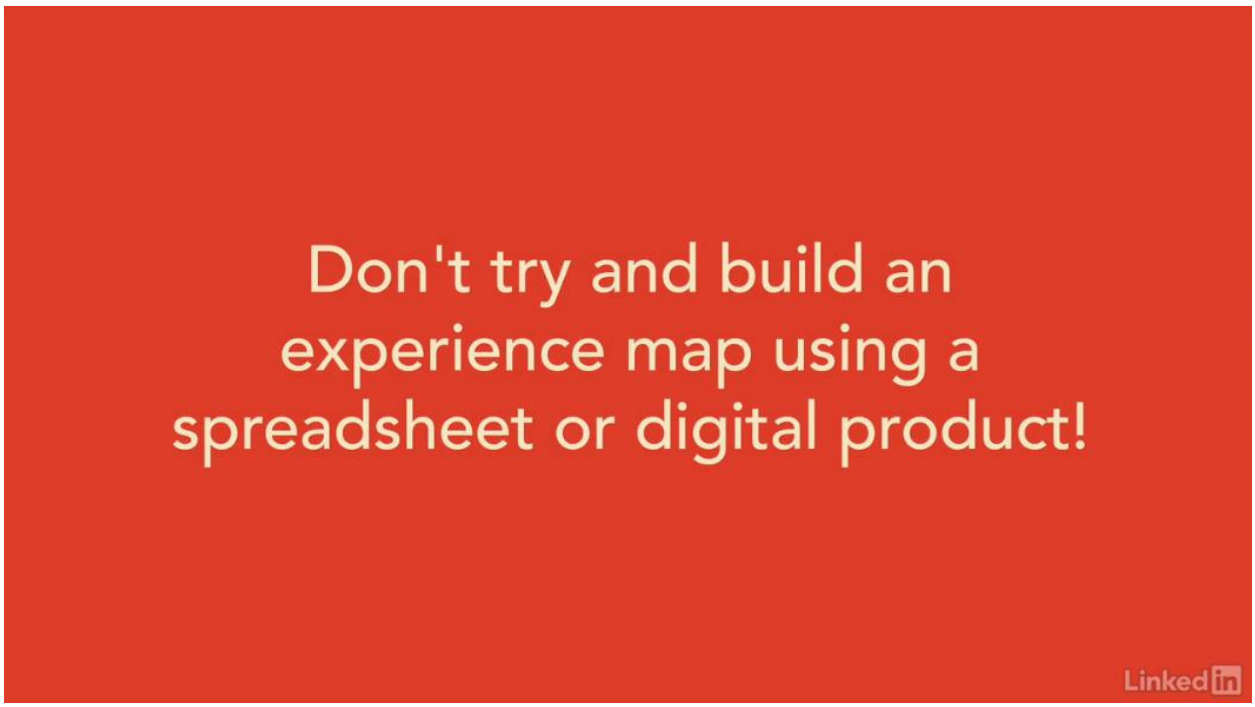
move on. Design ideas aren't user data, they're your interpretation. At this point in the process, your interpretation may be bad. So just get the idea out of your head and keep going. If you think of more questions that you wish you'd asked, or of data you want to check to see how prevalent a problem might be, add those questions on another different colored sticky note. I'm using pink here. Because it's about the only other common color you get in a multi-color set of stickies. Questions, the things you wish you had the opportunity to observe, are an indication of how much more research you need to do before you can be confident that you understand the user's process. Many of the questions stuck on the experience map will be answered in subsequent research. But if you end up with a lot of questions about user behavior, you probably need to get out and do more site visits before proceeding. Conducting more visits just makes the map richer. You might have heard the term experience map used to describe highly designed graphical products that are shown during executive briefings. Believe it or not, most of these experience maps, at least the good ones, started out as a diagram very much like the one I've just described. For our purposes though, there's no need to get all glossy. The low-fidelity set of sticky notes on the wall is more than sufficient for our needs, and it contains more data than the glossy versions.

Create a successful experience map





As needed, you can add observer's initials to notes.



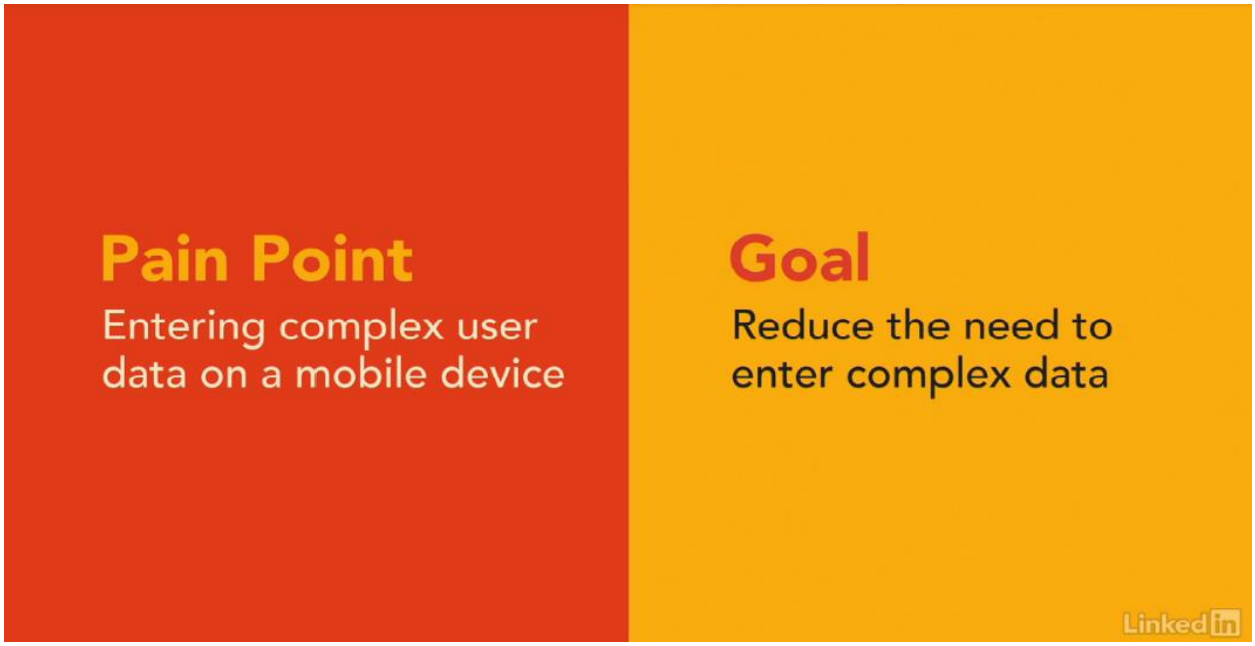
I've described the basic process for creating an experience map. But there are some additional things you should bear in mind as you go through the process to make sure it runs smoothly. The first and most important tip is to make sure that the majority of sticky notes that people add to the map are actual user quotes. Quotes are the most basic and most believable kind of data we can add to the experience map. They come straight from users with very little interpretation and so they're hard for people to argue with. Different team members will have seen different users. Bring them all together to do the map creation at the same time so they can share their experiences. Get team members to read out sticky notes as they place them so that other team members with similar findings can stick those up at the same time. Do this as soon after the sessions as possible because the details fade from memory over time. I like to run the site visits in the morning and then do the experience map creation that same afternoon. Teams will often get left with a set of sticky notes that describe user attributes rather than observations about specific tasks. That's great because we need that kind of data later in the process when we create personas. I normally leave a space to the left of the main experience map to stick these observations in one place so we can save them for the persona process. Although it's not truly necessary, if you have multiple observers it might help to put the observers initials in the corner of the notes for later reference. When you read through other people's sticky notes it's sometimes easier to get clarification about what someone meant if you can immediately ask them. My final tip is don't try and build an experience map using a spreadsheet or other digital product. I've seen it tried several times and the biggest issues are that it's too hard to get the big picture at a glance. Because moving things around is a pain people don't tend to rearrange their observation stickies after they first place them. All in all, it's better to work on paper and then capture a digital image of the result rather than trying to use a digital tool to make the map in the first place.

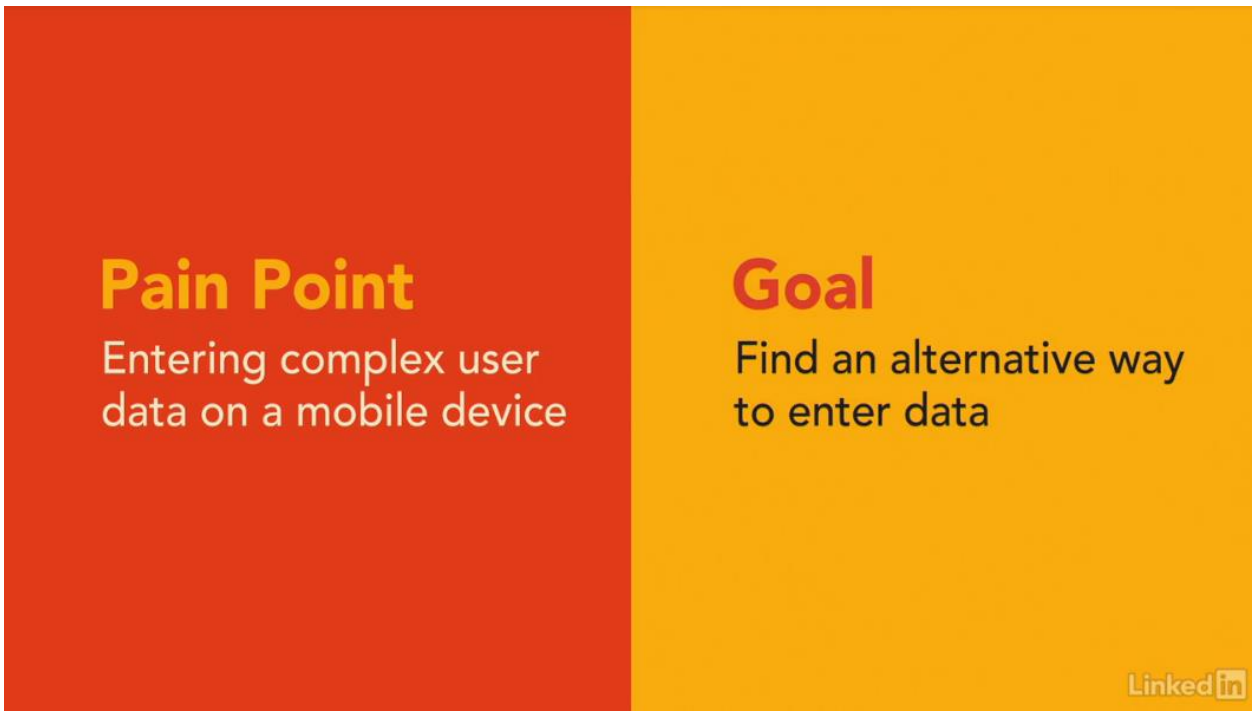
4. **Actionable Data**

Gather the pain points

With all the observation data sorted into columns in the experience map, it's pretty easy to identify the areas of the current interaction that are causing problems for users, the pain points. The best way to extract pain points from an experience map, is to get the people who created the map to dot vote. Dot voting involves giving every team member a small number of sticky dots, say five or six dots each. Team members can choose where to put the dots. They can put them all on one issue if they think it's really important, or split the dots over a couple of different issues, or place one dot each on several different issues. Everyone dot votes at the same time. Because they were the people who created it, their familiarity with the experience map, let's them quickly identify the top issue areas. What you'll find is that some observations get many sticky dots, others get a few, and some get none at all. Now, you copy out the issues from the experience map to a clean sheet of paper, starting with the one that had the most dots. You might choose to combine a couple of issues if they seem to be describing the same problem. Then, you also combine their dots, this way, the issue that the team gave the most dots to overall still stays at the top of the list, with the highest priority for fixing. When you've copied out all the observations with two or more dots, and consolidated issues where necessary, you'll have a list of about five to ten pain points. It's interesting to take a step back and just look at the pain points you've uncovered at this time. Some may have been ones you knew about before from customer feedback. But it's likely that some have their root cause in areas you were not aware of before. This shows the power of gathering data from user's locations. Issues arise that you wouldn't normally be aware of and that users may not have thought to report to you because they see the same thing every day. Pain points are a great way of exposing issues, but it helps to describe how we intend to resolve the pain points in general terms by turning them into goals.

Set goals

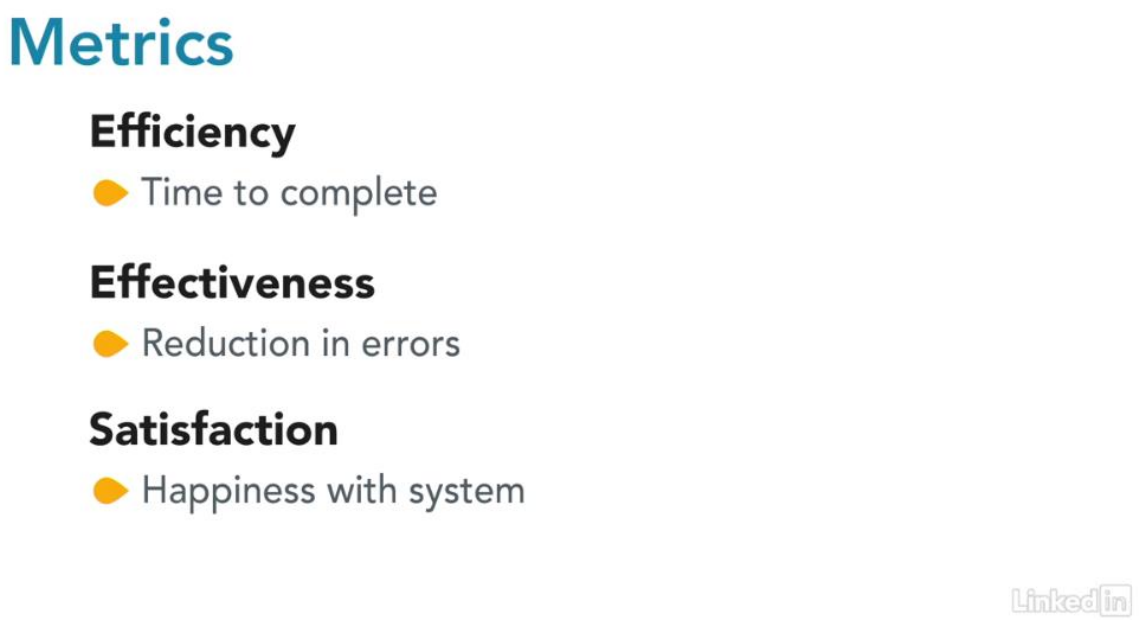




Once you have a list of Pain Points, it's relatively easy to rewrite each one as a goal simply by inverting it. List the benefit the users would get from your product if you fix the Pain Point. Then turn that description into a goal. For instance, you might've identified a Pain Point around entering complex user data on a mobile device. The benefit of fixing this problem is to reduce the frustration and number of errors around mobile data entry. Your goal might either be to reduce the need to enter complex data on a mobile device, or to find an alternative and more efficient and satisfying way to enter data. For instance, by using voice recognition. You already prioritized and merged the Pain Points from a user perspective. Now, you need to do the same thing again with your goals. But this time, also introducing a business perspective. This allows you to balance user need with business need in the solutions you design later on. Now it's time to go through the list of goals and decide which have the biggest impact on your product or business. Often, you'll find that fixing the biggest user Pain Point also has the biggest business impact. But that isn't always the case. Once you have your prioritized list of business goals, you're nearly

done. However, goals aren't much use without a way of understanding when you've met them. For that, you need metrics, which we'll discuss next.

Develop metrics



Metrics

Efficiency

- New input method is at least 20% faster.

Effectiveness

- New input method introduces 50% fewer data entry errors.

Satisfaction

- New version of app goes from 3/5 to 4/5 review stars.