

Interviewing Users

by **JAKOB NIELSEN** on July 26, 2010

Topics: Research Methods

Summary: Despite many weaknesses, interviews are a valuable method for exploratory user research.

What users *say* and what they *do* are different — a point I've made countless times. I even wrote a column entitled "First Rule of Usability? Don't Listen to Users" that's as relevant today as it was in 2001. (The best usability methods are highly stable, which is why learning valid methodology has such a strong career-long ROI.)

Observant readers have complained that I violated my own prescription in my recent analysis of response-time delays. In that article, I reported on a series of *interviews* we conducted with users when researching the concept of Brand as Experience. So, what's with suddenly asking people what they think instead of observing their actual behavior?

The answer is that interviews are in fact an appropriate user research method — if you use them only in the few cases for which they generate valid data.

What Interviews Can't Do

Before getting to the good side of interviews, let's review their many bad points. (Many of which they share with focus groups which are vastly overused in web design projects.)

The critical failing of user interviews is that you're asking people to either **remember** past use or **speculate** on future use of a system. Both types of responses are extraordinarily weak and often misleading.

- Human **memory is fallible** (as discussed further in our Human Mind seminar). People can't remember the details of how they used a website, and they often tend to **make up stories to rationalize** whatever they do remember (or misremember) so that it sounds more logical than it really is.
- Users are **pragmatic and concrete**. They typically have no idea how they might use a new technology based on a description alone. *Users are not designers*, and being able to envision something that doesn't exist is a rare skill. (Conversely, *designers are not users*, so it doesn't matter whether they personally think something is easy.)

Envision a **timeline** of user comments: only one spot generates valid data — the present: what the user is doing *right now*. Having users misremember the past or mispredict the future should both be shunned.

One of the reasons that **specs don't work** is that users (and management) can't tell whether a specification documents something that will solve their problem once built. It sure sounds good in writing, but there are endless case studies of "user reps" signing off on stuff that ended up as big failures.

This is why Agile development and paper prototyping methods are valuable. When users have something concrete to interact with, it's usually obvious when you're solving their problems in a way that's easy and pleasant to work with — and equally obvious when you're not.

Most **specific design questions can't be answered by interviewing users**. Here are some of the **things you won't learn in an interview**:

- Should the *Buy* button be red or orange?
- Is it better to use a drop-down menu or a set of radio buttons for a certain set of choices?
- Where should the Foo product line reside in the IA?
- Is it better to have 3 levels of navigation, or should we stick to 2 levels even if it means longer menus?
- How should you write the *Help* information to best teach people how to correctly use the system?

Sure, you could ask users each of these questions, but the answers will be completely unrelated to effective design for a real website. Dilemmas relating to specific UI elements can be resolved only by watching users interact with a design that implements a specific solution, so that you can see how well it works in real use. (Or you can implement multiple options and run a comparative test.)

Similarly, you can't ask "Would you use (potential future) feature X?" because users can't predict something they haven't seen. You can't even ask "How useful is feature Y?" for features that already exist. Indeed, in many studies, facilitators asked users to comment on specific features that didn't exist but were seeded into the interviews as ringers; the users provided copious feedback.

(The take away? If you're compelled to ask users how much they like your features, then be sure to include a few nonexistent features to collect a baseline.)

In one famous study, Microsoft asked customers to suggest new features for Office 2007 before starting work on that product. Most of the requested "new" commands already existed in Office 2003, so the design team correctly concluded that their main problem was the discoverability of the existing functionality.

The way to assess features is to have people use them. Definitely pay attention to users' comments while they're engaging with the features. You can even ask for supplementary comments immediately after tasks, while the features are still fresh in their minds.

What Interviews Can Tell You

For our brand study, we wanted to learn how using a website over time builds users' impressions of that site and their expectations for its brand promise. In other words, we weren't interested in individual page designs — which we'd study through user testing — but rather we wanted to know what users thought of a site after using it. And that's best assessed by asking them.

Interviews are also useful when you want to **explore users' general attitudes** or how they think about a problem. After getting this info, it's your responsibility to design features that address the problem (and to test prototype designs of those features to ensure that you got them right).

The **critical incident** method is especially useful for such exploratory interviews: Ask users to recall specific instances in which they faced a particularly difficult case or when something worked particularly well. These extreme cases are often more vivid in users' minds, and will give you the details needed to come up with useful features.

(In contrast, if you ask people how they *usually* perform a task, they'll often describe an idealized workflow without the many shortcuts and deviations that characterize real projects, whether at home or in the office. One of the key determinants for good application workflow is to avoid idealized situations and design for the way people actually do stuff.)

Beware the Query Effect

Whenever you do ask users for their opinions, watch out for the query effect: People can **make up an opinion about anything**, and they'll do so if asked. You can thus get users to comment at great length about something that doesn't matter, and which they wouldn't have given a second thought to if left to their own devices.

It's dangerous to make big design changes because "users didn't like this" or "users asked for that." If you ask leading questions or press respondents for answers, they might make up opinions that don't reflect their real preferences in the slightest.

For example, if you quiz people about your visual design, you'll inevitably get comments about the colors, even if they're not particularly important to the users. On the other hand, if you hear people mention the colors (unprompted) while they're using the site, then there's probably something to consider. (Say, a comment like "Wow, this neon blue really hurts the eyes," or a more positive statement like "The ultramarine is nice and calming.")

Combine Methods: Data Triangulation

Interviews are great supplements to other usability methods. If you could do only one thing, I would always recommend user testing. But why limit yourself to one method? Each method can require only a few days' work, so you can combine multiple methods on all but the very smallest budgets.

Let's return to the example that prompted this column: our latest findings regarding website response times. If you want to know the best speed for a specific pageload, forms handling, or AJAX widget manipulation, you have to watch users perform representative tasks with these designs. If something is too slow, you'll note users become impatient, forgetful, and ultimately leave the site. But if you ask them weeks later, they won't know specifically which UI element was too slow; nor will they recall the number of seconds that constituted the limit of their attention span. Conversely, if you want to know the branding impact of sluggish or snappy sites (our recent research question), then interviews are fine. For this higher-level question, you'll want to learn **what made such a strong impression that it stuck in users' minds** long after they used the sites.

Each method has its strengths and weaknesses. Taking the best input from each method will give you a much richer understanding than you could gain from any one method alone. Also, by supplementing each method with a range of other approaches, you can **triangulate** the findings and guard against misleading outcomes.

Copyright © 1998-2017 Nielsen Norman Group, All Rights Reserved.