Interview

Story interviews

Ask a user questions that elicit recent stories about specific events or objects relevant to the design brief

Introduction

We all assume we know how to interview people. After all, we have all seen journalists' interviews and may have been interviewed ourselves. But interview techniques differ greatly, based on the desired outcome: a journalist's interview should challenge the interviewee, a talk show host's interview should entertain, a police detective's interview should uncover new facts about an event, and a sociologist's interview should contribute to a deeper understanding of human behavior.

Design-oriented interviews have yet another goal—to improve your understanding of behavior so as to directly influence your design. We distinguish three types of questions, which produce three different types of results: *stories*, *tutorials* and *opinions*, in order of usefulness for design.

Story interviews are the most useful, since they capture detailed examples of real-world interaction. Stories are always told in the past tense: the user recounts what happened, step-by-step, with as much detail as possible. The most effective story questions take advantage of human memory, asking users to recount a highly memorable event (critical incidents[1].), describe how they created a specific object, or describe a routine but extremely recent activity. The resulting stories offer designers maximum insight into the user's experience, including examples not only of what goes right, but, more importantly, what goes wrong. What causes the breakdowns? What workarounds do users try? Are they successful? Finally, were they any user innovations? The latter, while rare, are worth seeking out. An individual users' novel perspective can lead to an inexpensive, field-tested new market for the system.

Tutorial interviews are often the result of a story interview gone wrong—rather than describing a specific activity, the user invents a description of how that activity should be performed. If you hear phrases such as: "*I usually do this*" or "*Every time this happens, I do that*", the user has shifted from telling a story to treating you as a student and teaching you an

[1] John Flanagan developed the *Critical Incident Technique* when interviewing pilots who survived a plane crash, using episodic memory to help them remember the details leading up to the accident. Social scientists use variations of this technique to capture detailed examples of human activity in order to develop or verify theories of human behavior. Story interviews also take advantage of human memory, stressing the recency and salience of the events being recalled, but the interview goal is different. Rather than deveoping theory, designers use *story interviews* to learn the specifics of a particular user experiences, both good and bad, to inspire new opportunities for design.

idealized version of what is supposed to happen. Tutorials are useful if you are new to the topic and trying to understand what 'normal' behavior looks like. But they offer few insights into design, and encourage designing for stereotypes instead of real behavior. If you do decide to conduct a tutorial interview, ask for at least one specific story first. The user will remember more details and provide a better-grounded overview of the idealized activity.

Opinion interviews, although very common, are by far the least useful for design. This is counterintuitive, since both users and designers are all familiar with marketing surveys and assume that opinions are what is expected. While it is true that users' opinions can help to identify features that work well or 'pain points' that must be addressed, they offer few clues as to causes, nor do they provide inspirations for design. Worse, unless these opinions are may be made up on the spot, not grounded in actual experience. If you need to collect users' opinions, do so only at the *end* of the interview, after first obtaining specific stories and tutorial examples. Remember that opinions that arise naturally as the user talks about what actually happened will be far better grounded in their real experience, unlike opinions offered 'off the top of the user's head'.

Learning how to conduct story interviews is the most important method in this book. Story interviews take a bit of practice, but once you learn how to direct your interview questions back to capturing more details of the users' story, you will find the resulting interviews offer a wealth of design ideas. Unfortunately, since you are surrounded by other types of interviews, in the beginning, you may have difficulty recognizing that you are doing it wrong. The biggest clue is if you conduct an interview and do not find any insights for your design.

To summarize, if you master the story interview technique, users will tell you stories about recent, memorable interactions with technology. These offer new insights for your design and are far more useful than tutorial-like explanations and general opinions. Your designs will be better if they are influenced by multiple real stories, rather than general stereotypes of what usually happens.

What to do

A single interview may contain different types of questions that elicit different results: stories, tutorials and opinions, *always in that order*! If you let the user start with a generalized description of an activity, or worse, ask for their general opinions about an existing system, you will find it very difficult to bring them back to telling specific stories. Worse, their tutorials and opinions will be poorly grounded and perhaps invented. If you keep urging the user to get to the specifics of a specific, ideally recent, account of what they did, you can always get them to summarize in tutorial form afterwards, and they will spontaneously give you opinions as they go. Even better, their opinions will be better grounded in their actual experience, and more likely to truly reflect what they believe.

Story questions elicit detailed stories, ideally including breakdowns, workarounds, successful activities and user innovations. The key is to obtain a step-by-step account of what happened, with as much detail about the interaction as possible.

Preparation (before)

Topic: Choose a topic, the more specific the better. Clearly, if the design brief is to improve an existing system, you should focus on users of the current system, as well as users of any competing systems. If you are trying to create an entirely new system, you still need to have an idea of what people do today, without your system. Try to identify people who perform related activities.

Recruiting: Finding people to interview can be tricky if you do not have easy access to your potential target population. This is especially true if your system is designed for people with specific skills, such as doctors, or with particular challenges, such as the handicapped, or people with strong interests, such as fitness buffs. The best strategy is to find someone who is a member of that target audience who can introduce you to others. You can take also advantage of relevant social networks or special interest mailing lists. Ideally, go where they gather, such as a conference, meeting, or where they live.

Be sure to practice your interviewing skills first, so you get the most out of each interview. Interviews can provide extremely useful information, if done well, but they take a great deal of time and you don't want to waste interview subjects.

Informed consent: Before you interview someone, need to let them know the purpose of the interview and how you will handle their data. This is not simply a matter of getting them to sign a legal consent form—you are ethically obligated to ensure that they understand the consequences of being interviewed and still agree.

Ethics: If your university or company has an existing *Institutional Review Board*, follow their procedures. But even if you work for a small startup or are taking a class, you should follow basic ethical guidelines, and always be very careful when handling interview data. You must consider both the risk of harm to the user and the likelihood of that risk. For example, if your interview might reveal embarassing information or raise legal issues for them[2], you must let them know in advance and *always* give them the option of dropping out at any time.

If the risk of harm is small, but the data is easily accessible, you should still restrict data access, by anonymizing the data and keeping it separate from personal identifiers before you share it with anyone. Best practices suggest that you assign a unique participant identifier to each person you interview, and keep a single paper, not electronic, record that links their names and personally identifiable data with the participant ID. Use only the participant ID to label the actual data. For example, I might assign Marie Dupont P07 as her participant ID. Her name should be paired with this ID once, on the project sheet, and all the remaining data should be labeled P07. The summary sheet should be kept in a locked cabinet, separate from the other data, with limited access. Ensure that there is no online method of linking her name with her data. See the Ethics chapter for more details.

Interview sheet: For each interview, prepare a sheet with the participant ID, the date and time, setting, and a short phrase summarizing the topic. Include your name, organization and project name. Next, write several story questions that will serve as your guide during the interview. If you are just learning this interview technique, include a few reminders about what to do if you hear red flag phrases, such as "Usually, I …" or "every time I…". You will not be able to anticipate all the possible branches of a question—you will have use follow-up questions to probe for more detail.

Procedure (during)

Begin by explaining who you are and remind them of the purpose of the interview. (Presumably you already explained the project and obtained their permission when you recruited them.)

Create a Design Resource (after)

You will find that, if you systematically schedule 30-60 minutes after each interview to reflect and tidy up your notes, you will save huge amounts of time later. Just as story questions work best when the event being described is recent, you will be able to remember the most when the interview is recent.

Later, gather your notes from your interviews and transcripts if you made them. Look for interesting quotes and comments that you marked as interesting during the interview. Look for anything surprising, paying special attention to breakdowns (where the user cannot perform the task as expected) and workarounds (where the user tried to solve the problem a different way). Some workarounds lead to more failure, others lead to success. Users may be happy with the workaround, or feel that it is a poor substitute. Finally, look for user innovations (where the user came up with a clever new way to accomplish something).

Chunk these into descriptive phrases. It helps to put them on small scraps of paper or postit notes, since the next step is to create categories for these different phrases. Phrases may fit into more than one category. Next, give the categories code names. Select the key categories relevant to the design, and create a list of specific examples from your interviews and other user data, with quotations, insights, and any related artifacts.

Annotated Glossary entry

Story interviews result in detailed, step-by-step stories of how one or more users interacted with technology at a particular point in time.

Tutorial interviews result in an individual user's summary of how an activity is normally performed, from their perspective.

Opinion interviews result in a user's general views about a particular technology or activities related to that technology.

Breakdowns occur when the user cannot accomplish a task or perform an activity as expected or desired.

Workarounds are temporary solutions to a breakdown, either from within the system or using other technology.

User innovations are creative 'out-of-the-box' solutions to a breakdown, usually taking advantage of hitherto unused properties of the existing system or by incorporating aspects of an external system.

Informed consent ensures that the user understands the consequences of engaging in a design activity before agreeing to participate.

A small experiment

If you do not believe me, try the following experiment.

Choose a mobile app or software application that you know well. Spend 10-15 minutes using the introspection technique to perform a task, recording what happened at every step. Write down what you wanted to do, what interaction you performed, what the system did in response, and your reaction. At the end, highlight any breakdowns or workarounds, as well as any surprises. You should now have a story of how you tried to accomplish a task, including the details of the interaction with the system. From those details, you and other people can infer how the system usually works and your opinions about it, but will also get specific examples of how the system breaks and what kinds of workarounds make sense in the current context. Of course, you cannot design a complete system base on a single story—but I would argue that, if you tried to, you would probably produce a more interesting and relevant design than if you relied solely on a tutorial, or worse, a set of opinions.

Now, try interviewing a friend about their use of the same app or application. Begin by asking her to tell you about the last time she used it, and to show you exactly what she did, step by step. Ideally, she should have access to the system, which will trigger her memory. If she veers into an explanation of how the software works, politely tell her she can explain that later, but right now, you want to know precisely what happened next in the story. As in the introspection, at each step, try to discover what she wanted to accomplish, what she did, how the system responded, and her subsequent reaction. Pay attention to breakdowns, workarounds, and any surprises, especially if she found an innovative way to perform a task. You should have the same level of detail in their answers as you did in your own.

Finally, try interviewing another friend about the same app or application. This time, just ask them to tell you what they think of the system. (Some people may tell you explicit stories, but most will offer general opinions or possibly a mini-tutorial. Spend the same amount of time as for the first interview and take notes. Did you get the same amount of information as you did for the introspection? Do you have actionable details? Did you learn anything new that might inspire new design ideas?