

Introspection

Individual

Watch yourself perform a specified task with an existing system, keeping a step-by-step record of each interaction point

Introduction

Interaction designers take advantage of number of design techniques from various disciplines that help us to learn about the user's perspective, each with advantages and disadvantages. The fastest, cheapest and easiest (but not the best!) is called introspection. Introspection is defined as the direct observation of one's own thought processes and is the opposite of extrospection, the observation of things external to one's self.

Careful! If you a researcher, introspection is not considered a valid *scientific* approach. However, it can be a valid *design* technique, as long as it provides useful insights and encourages you, as the designer, to go beyond a purely functional or architectural view of the system and begin to think about the user's experience.

A word of warning, though, for computer scientists and engineers. Research has shown that they think about computer systems very differently than most other users. For a programmer, the computer is the focal point of his or her work, whereas for the 'ordinary user', the computer is only a means to and end, that is, to accomplish other 'real' work. Regardless of your background, it is important to remember that analyzing your own use of a system that you have designed or built provides only a limited and sometimes misleading view of the system. You must also observe other users, to understand their perspectives.

Try to find a balance between a very informal, high-level process and an overly rigid, obsessively detailed one. The former will miss useful details and insights; the latter will waste time and produce unnecessary detail. The goal here is to stay open to surprises and capture them, including any breakdowns, bugs, or unexpected features (positive as well as negative) that lead to new insights about the current design and suggest possibilities for a future design.

What to do

Preparation (before)

Set aside a fixed period of time, 15-30 minutes, and ensure that you will not be disturbed. Choose a current, real task that you need to do anyway and decide in advance how you plan to record your thoughts. You may use a screen recording or record audio as you talk through your actions. These will require later transcription but will provide the most detail about your actions and what you were trying to do at each point.

In addition, you should write notes (either electronically or on paper), alternating between doing the task and reflecting upon it.

Focus on your *interaction* with the existing system: What did you do and how did the system respond? What was your subsequent reaction? Was it what you expected? Was it what you wanted? Did the system interrupt you? Did you understand the interruption? Did you want the interruption? Keep going, step by step, using the system and recording your reaction to it, until the time is up. At the end of the session, think about what you learned. Are there particular details that are annoying (or particularly useful)? What surprised you? What is your overall reaction to the system? Have you any particular ideas about things you'd like to improve?

Procedure (during)

When you are ready, ensure that you will not be disturbed. Set a timer and start the task, in this case, editing a text document. Stop after every step and note the details of your interaction. (Note: If you perform a series of the same steps several times, you can note the repeated sequence, without restating the details, if the interaction is the same. Be sure to note how many times you repeated the sequence.)

You can write your notes in paragraph form or use a table format where you enter each step on a new line. If you use the text format, highlight the key interaction points when you are done. For both formats, use a symbol to indicate surprising events.

Example: Redesigning a Smartphone Map App

Today's google maps and bing maps are issued as 'free' applications that are funded through advertising. They also operate on the assumption that GPS is always available, and that 'getting there' is easy, since one can simply follow the GPS. This results in the current design which devotes a third of the screen real estate to the name and address of the current place being investigated and directions for how to get there. The rest of the screen is covered with pins, most of which are irrelevant to what you are looking for. You cannot turn off the ads for various places, so they become an annoying part of the map that you try to create if you're telling people how to get to your place or to a particular event. The goal is to redesign the map so users can discover what they are looking for and also repurpose the maps for other uses.

This year's project focuses on how people use maps on a smart phone. Choose a mobile app or software application that you know well. Spend 10 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.

Create a Design Resource (after)

When time is up, go back through your notes. If they are hand-written, type them and add any screen shots or images to illustrate unexpected events. If you have an audio or video recording, check the surprises indicated in your notes and find the corresponding audio or video to can get more detail about what happened.

The final design resource should consist of a step-by-step sequence of the actions you performed, how the system responded, as well as any surprises:

- goal: What did you want to do?
- user action: How did you interact with the system?
- system response: How did the system react?
- comment: Did it do what you expected?
- surprises: Highlight breakdowns, bugs, unusual events (positive or negative)

When you are done, you should have a deeper understanding of how the choices of interaction techniques made by the software designer affects the user's ability to accomplish a task.

Variation: Peer Introspection

Peer introspection: Pairs of users (or designers) to work together and explain to each other what they are experiencing. The advantages of this technique are that shy users are much more likely to talk to another person than just talking out loud, will be able to see and perhaps compare differences in how two users experience the same event, and you will find out more about how users think as they try to explain how the system works to their colleagues. The disadvantage is that one user may overwhelm the other or that they may come to a single conclusion rather than provide two different perspectives. In this case, the roles are separated, so that one person is the performer and the other is the scribe.

A good introspection results in new insights about the details of the interaction between the user and the system. Ideally, you should be able to highlight and explore the surprises you discover with other designers or with users (to see if they experience similar problems or find similar solutions).

Selected references / Bibliography

Flanagan, J.C. (1954) The critical incident technique. *Psychological Bulletin*, 51(4), 327-358. <http://dx.doi.org/10.1037/h0061470>

The "Think-Aloud" protocol, introduced by Ericsson & Simon, is a second-level introspection technique. The user is asked explain out loud what they are experiencing as they try the software. This provides the designer or researcher with a much richer understanding of the details of what users are thinking as they work

Lewis, C. H. (1982). Using the "Thinking Aloud" Method In Cognitive Interface Design (Technical report). IBM. RC-9265.

*Ericsson, K., & Simon, H. (May 1980). "Verbal reports as data". *Psychological Review*. 87 (3): 215–251. doi:10.1037/0033-295X.87.3.215.*

Ericsson, K., & Simon, H. (1987). "Verbal reports on thinking". In C. Faerch; G. Kasper. Introspection in Second Language Research. Clevedon, Avon: Multilingual Matters. pp. 24–54.

Ericsson, K., & Simon, H. (1993). Protocol Analysis: Verbal Reports as Data (2nd ed.). Boston: MIT Press. ISBN 0-262-05029-3.