
Usability testing

IRM believes that usability testing is a core discipline in our field. We try to practice it and to continue refining our skills in doing usability tests. The discussions that led to this statement have been useful in clarifying our thinking and describe the state of the art as we see it today. Perhaps this statement can be helpful to you. Send us your comments if this is not the case or if you see ways of improving the statement.

Usability testing is a core skill because it is the principal means of finding out whether a system (see our definition below) meets its intended purpose. All other skills that we deploy or cultivate aim to make usability (and, ultimately, use) successful.

Definitions:

- **System:** a collection of interacting and interdependent parts that are organized to meet some purpose. We are especially interested in the case when a system is designed for the use of other people. As used here, the word system could be a service, a collection of computer programs, a diagram, a piece of text, a methodology, a set of instructions, or a dataset.
- **Usability:** the extent to which the intended user can meet his or her goals using the system being tested.
- **Usability subject:** a potential user of a system who participates in a usability test.
- **You:** the person who is trying to understand the usability issues and usability extent of a system. You may or may not be the actual author of "the system" being tested.

Prepare.

Think about why you will be doing a usability test.

How do people interact with the system you are testing? What is difficult or easy for people to do? What makes sense about it? What is exciting about it? What changes would users like to see? What do they really hate about it? The way subjects actually use your system may reveal bugs that are invisible to you. It also may suggest enhancements that do not make sense or seem necessary when you test the system.

Consider the system as a whole.

Think of a usability test as a way of examining your thinking about the system. What are the boundaries around it? That is, a usability test is a means of thinking about the purpose and function of the "whole product" or "whole system." What additional tools, information, skills, and support will people really need to use your system? In a client/server context, you need to test "the whole" even though some elements may be fixed and beyond your control. (Of course, if you can't influence the behavior or structure of a system at all, there isn't much point in a usability test.)

Make sure the system is ready to test.

Convince yourself that the system is functional and usable from your own perspective. Usability testing is expensive and your subject's time is precious. Have you used all the mechanical tests that are available to you? For example, have you run a spell-checker on all of the text that the user will

see? Have co-workers or others on the design team checked out the system?

List several tasks that a user should be able to accomplish with the system.

The tasks should be simple so that first-time users have a chance of success. The tasks should be specific, and should be stated in a very clear way. (For example: "Determine the total of annualized salaries for classified staff in your department.") Describe those tasks in writing so you can give users something to work from. Write each task on a separate page to avoid foreshadowing effects. Do not include hints in the task descriptions. If the test lasts more than a half hour your subjects will be exhausted and so will you.

Make a list of potential usability test subjects.

Pick people completely unfamiliar with the system. (This implies that once someone has participated in a usability test, they are no longer useful in that capacity. They've been indoctrinated or exposed to the system.) The subjects could be people whom you know very well or don't know at all. There are different issues involved with each scenario. With friends, a rapport has already been established, they trust you, and they know you're not passing judgment on them. With strangers, this rapport needs to be established.

Plan for data collection.

How will you write down what you hear and what you observe? Would thumb-nail sketches of the screens the user will see be a good armature for your notes?

Schedule the test.

It is best if you can go to the users' own environment where they are most comfortable and it will take a minimum of their time. This allows them to focus on the tasks and the system, rather than on getting comfortable with a new mouse, a different version of the software, or a different context of whatever sort. These usability test notes assume that there will be one subject and one observer; sometimes a team of observers makes sense.

Prepare yourself to be objective.

It takes a lot of emotional and intellectual energy to listen and receive feedback that may be negative.

Don't take this "how" file with you.

If this testing ritual isn't intuitive and natural, it probably should be re-written. You've been nominated!

Present.

Establish rapport.

Take some time to establish a rapport so that the subject will feel comfortable during the test.

Establish the context.

You might explain how usability testing is part of the design and development process. Emphasize that **the system** is being tested, not them and not you. You are trying to understand how the system would work in their environment. (It's not very helpful if they are trying to imagine how other people would use the system; ask them to test the system for themselves.) It may be helpful to give some examples of how a usability test can improve a system (e.g., show how it gives confusing or contradictory signals, how too many or too few things are supposed to happen at once, or how specific assumptions about the users' knowledge and skills may or may not be true, etc.)

Ask the subject to verbalize their thoughts as they perform the tasks on your list.

Encourage them to say anything that comes into their mind during the process. Your role will be to write their comments down now for later discussion and analysis. Mention that there will be time later for discussion.

Don't help!

Mention that since the system that you're testing doesn't include you as an on-site help resource, the goal is for the subject to try to solve the problems at each step on their own. Can people accomplish the tasks which the system seeks to address **by themselves**?

Remember that you don't have to have all the answers.

A usability test should be a learning experience. If a question comes up that you cannot answer, let them know that you will get back to them later.

Don't take it personally.

Make an effort to not be defensive or take criticism personally. You'll probably hear things that indicate that the system is not perfect. Don't try to make excuses for the system. Just write down the comments.

Thank the subject.

Thank the usability subject for taking the time to help you to improve the system. Emphasize how the knowledge you gain from this process will be used to make a more usable system. You might describe how much change is really possible at this point: are you testing an early prototype or a beta release? In other words, can features be added or are you just fixing bugs?

Listen and observe!

Have fun with it!

Enjoy being the observer.

What does "hmm" mean?

Take it all in, even the gaps. Notice all the sounds and behaviors and comments that might be relevant. Remind the subject to verbalize and be as open with their thinking as possible.

Write everything down.

You can translate and expand your notes later. If you are tempted to ask the subject extra questions, make a note so that you can bring them up later.

Don't interfere or ask leading questions.

Do help them to feel comfortable speaking their thoughts out loud.

Engage in dialogue.

Ask any questions that have come up during the usability test.

If you formulate questions as you listen and observe, try to keep them to the end so that people are not distracted from the task at hand. If it is imperative that you know the answers immediately to continue with the test, go ahead and ask.

Verify your original task list and other assumptions, if possible.

Did they make sense to your subject? Why?

Plan for action.

Translate comments into system modifications.

Consider what part of the system needs to change. Remember that from a user's perspective the context for the system is part of the system. Who should be told about what you've learned from the usability test?

Consider asking for suggestions.

Discussions of how to respond to the problems a user found with your system may or may not be appropriate.

Acknowledge.

Thank them for their participation and contribution.

Remember that you and those who use the system are the beneficiaries.

Keep them informed.

It's nice if you can send a quick note when you're done with any modification that resulted from their comments saying, "Thanks for the your comments, please look at the system now and see if we've improved it as you expected..."

Cool off.

Remember that it takes time to digest what you hear from a potential user of the system. The system will be so much better because of the comments you've gathered, even if they are discouraging at the moment. Review your notes after you've cooled off.

Keep at it.

It takes a lot of skill to do a good usability test. You'll get better if you keep practicing.

The *categories* in this "how" file are based on Rick Mauerer, **Feedback Toolkit; 16 Tools for Better Communication in the Workplace** (Portland, OR: Productivity Press, 1994). The ideas have been borrowed by IRM staff from classes and books too numerous to list and have been elaborated through use by IRM staff.

[Up](#)[Comments](#)[IRM Home](#)

<http://www.cu.edu/irm/stds/usability/>

Last Updated: 09/26/2002 15:19:30