

Task Scenarios for Usability Testing

Turn User Goals into Task Scenarios for Usability Testing

Marieke McCloskey

January 12, 2014

Summary: Guidelines for usability test tasks: engage participants by writing task scenarios that are realistic, encourage an action, and don't give away how the interface should be used.

The most effective way of understanding what works and what doesn't in an interface is to **watch people use it**. This is the essence of [usability testing](#). When the [right participants](#) attempt realistic activities, you gain qualitative insights into **what is causing users to have trouble**. These insights help you determine how to improve the design.

Also, you can measure the [percentage of tasks that users complete correctly](#) as a way to communicate a site's overall usability.

In This Article:

- [What Users Need To Be Able To Do](#)
- [Engage Users with Task Scenarios](#)
- [Conclusion](#)

What Users Need To Be Able To Do

In order to observe participants you need to give them something to do. These assignments are frequently referred to as **tasks**. (During testing I like to call them "**activities**" to avoid making the participants feel like they're being tested).

Rather than simply ordering test users to "do X" with no explanation, it's better to situate the request within a short **scenario** that sets the stage for the action and provides a bit of explanation and context for *why* the user is "doing X."

Before you can write the task scenarios used in testing, you have to come up with a list of general user goals that visitors to your site (or application) may have. Ask yourself: **What are the most important things that every user must be able to accomplish on the site?**

For example, nngroup.com users *must* be able to accomplish 3 main goals:

- Find articles on a specific topic
- Sign up for UX Week seminars

- Learn about our consulting services

Engage Users with Task Scenarios

Once you've figured out what the users' goals are, you need to formulate task scenarios that are appropriate for usability testing. A task scenario is the action that you ask the participant to take on the tested interface. For example, a task scenario could be:

You're planning a vacation to New York City, March 3 – March 14. You need to buy both airfare and hotel. Go to the American Airlines site and jetBlue Airlines site and see who has the best deals.

Task scenarios need to **provide context** so [users engage with the interface](#) and pretend to perform business or personal tasks as if they were at home or in the office.

Poorly written tasks often focus too much on forcing users to interact with a specific feature, rather than seeing if and how the user chooses to use the interface. A scenario puts the task into context and, thus, ideally motivates the participant.

The following 3 task-writing tips will improve the outcome of your usability studies.

1. Make the Task Realistic

User goal: Browse product offerings and purchase an item.

Poor task: *Purchase a pair of orange Nike running shoes.*

Better task: *Buy a pair of shoes for less than \$40.*

Asking a participant to do something that he wouldn't normally do will make him try to complete the task without really engaging with the interface. Poorly written tasks make it more difficult for participants to suspend disbelief about actually owning the task. In the example, the participant should have the freedom to compare products based on his own criteria.

Coming up with realistic tasks will depend on the participants that you recruit and on the features that you test. For example, if you test a hotel website, you need to make sure that the participants would be the ones in their family responsible for travel research and reservations.

Alternatively, you can decide to let the participants define their own tasks. For example, you could recruit users who are in the process of buying a car and let them continue their research during the session, instead of giving them a task scenario. ([Field studies](#) are ideal for observing users in their own environment as they perform their own tasks, but field studies are more expensive and time consuming.)

2. Make the Task Actionable

User goal: Find movie and show times.

Poor task: *You want to see a movie Sunday afternoon. Go to www.fandango.com and tell me where you'd click next.*

Better task: *Use www.fandango.com to find a movie you'd be interested in seeing on Sunday afternoon.*

It's best to **ask the users to do the action**, rather than asking them *how* they would do it. If you ask "How would you find a way to do X?" or "Tell me how you would do Y" the participant is likely to answer in words, not actions. And unfortunately, [people's self-reported data is not as accurate](#) as when they actually use a system. Additionally, having them talk through what they would do doesn't allow you to observe the ease or frustration that comes with using the interface.

You can tell that the task isn't actionable enough if the participant turns to the facilitator, takes her hand off the mouse, and says something like "I would first click here, and then there would be a link to where I want to go, and I'd click on that."

3. Avoid Giving Clues and Describing the Steps

User goal: Look up grades.

Poor task: *You want to see the results of your midterm exams. Go to the website, sign in, and tell me where you would click to get your transcript.*

Better task: *Look up the results of your midterm exams.*

Step descriptions often contain hidden clues as to how to use the interface. For example, if you tell someone to click on *Benefits* in the main menu, you won't learn if that menu label is meaningful to her. These tasks bias users' behavior and give you less useful results.

Task scenarios that include terms used in the interface also bias the users. If you're interested in learning if people can sign up for the newsletter and your site has a large button labeled *Sign up for newsletter*, you should not phrase the task as "*Sign up for this company's weekly newsletter.*" It's better to use a task such as: "*Find a way to get information on upcoming events sent to your email on a regular basis.*"

Avoiding words used in the interface is not always easy or natural and can even be confusing to users, especially if you try to derive roundabout ways to describe something that already has a standard, well-known name. In that case, you may want to use the established term. Avoiding clues does not mean being vague. For example, compare the following 2 tasks:

Poor task: *Make an appointment with your dentist.*

Better task: *Make an appointment for next Tuesday at 10am with your dentist, Dr. Petersen.*

You might think that this second task violates the guideline for tasks to be realistic if the user's dentist isn't really Dr. Petersen. However, this is one of those cases in which users are very good at suspending disbelief and proceeding to make the appointment just as they would with a differently-named dentist. You might need to have the user pretend to be seeing Dr. Petersen if you're [testing a paper prototype](#) or other early prototype design that includes only a few dentists.

Conclusion

If the task scenario is too vague, the participant will likely ask you for more information or will want to confirm that she is on the right path. Provide the participant with all the information that she needs to complete a task, without telling her where to click. During a usability test, mimic the real world as much as possible. Recruit representative users and ensure that each task scenario:

1. is realistic and typical for how people actually use the system when they are on their own time, doing their own activities
2. encourages users to interact with the interface
3. doesn't give away the answer.