



World Leaders in Research-Based User Experience

How to Maximize Insights in User Testing: Stepped User Tasks



Kara Pernice

February 9, 2020

Summary: You can learn the right kind of things and much more in user tests if you start with broad tasks instead of immediately leading to areas of interest. Prepare additional, focused tasks that can be used to direct users.

When [writing usability-testing tasks](#), you must walk the thin line between telling users too much and too little. Too often usability tasks direct users straight to the site area in which the team is interested, whether it's a redesigned website or a new piece of content. This approach will usually reap some information about the feature's usability, but it leaves on the table the potential to learn about the important topics of discoverability and findability. It's also the reason why some companies are doing lots of testing but still producing unhelpful designs. You can learn more if you start users off with broad instructions before directing them to what you are interested in. Prepare directed tasks that target your points of interest, but give them to participants only if the broad tasks don't give you the insights that you need.

In This Article:

[Stepped Tasks](#)

Stepped Tasks

Definition: A **stepped task** in a usability test is a task that is part of a set of related tasks (the “steps”) that present progressively more specific and elaborate instructions for the same activity. The first step in a set of stepped tasks is the vaguest, but observing users trying it may deliver all the information needed by the researchers, if the user performs the task fully.

Think of the second stepped task as a followup question that provides a micro hint intended to steer participants in the right direction if they didn't interact with the UI areas of interest during the first attempt. The third step (if needed) would give an additional micro hint — and so forth, for as many steps as needed.

An easy way to keep track of subtasks is to add a letter after the task number, or number them as decimal points of the main task; thus, for task 4, you would have steps 4, 4A, 4B (or 4, 4.1, 4.2, 4.3), and so on. When I am facilitating a usability test, even small things like this help me streamline the process and keep my team in the loop about what we are trying to accomplish in the research. (Note: I never show the users the task numbers because I may give tasks out of order or skip some, which is disconcerting to some people. Also, seeing task numbers can cause users to wonder how many tasks they will be asked to do, which can be distracting.)

Set of Stepped Tasks: Example

Let's look at an example of a set of stepped tasks in a usability test. Imagine that you want users to discover, find, and use the calendar feature on the corporate intranet.

Research goals: How easy it is to discover, find, and use the calendar?	
Task 4: Your colleague Richard Smith asked you to hold January 12 for a meeting from 2:00PM to 3:00. What might you do to make sure you don't forget?	What the task accomplishes: Suggests the available functionality without specifically stating that it's available in the UI, saying what it's called, where it is, or how to use it
User action: Goes to the intranet's calendar and makes an entry Facilitator action: Stops the multistep task here because all the relevant information has been collected	What we've learned: The user realized there is a calendar without being directly told there is one and used the features we were interested in.
User action: Picks up his phone and adds a to-do item Facilitator action: Gives the user the next step in the multistep task because the goal of the study has not been reached yet	What we've learned: The user decided to use a different strategy to accomplish the broad task, and we've got no information about the intranet calendar.
Task 4-A Can you find another way to make sure you keep January 12 from 2:00PM to 3:00 for a meeting with Richard Smith?	What the task accomplishes: Suggests the available functionality without directly pointing the user to the calendar feature, naming it, or indicating how to use it

	<p>User action: Goes to the intranet's calendar and makes an entry</p> <p>Facilitator action: Stops the multistep task here because all the relevant information has been collected</p>	<p>What we've learned: The user discovered the calendar without being specifically told there is one, and made an entry. (But we also know that our calendar was not the first thing to come to mind, so we do have some work to do.)</p>
	<p>User action: Says, "I wonder if there is a calendar," looks for one, but doesn't find it, and gives up on looking for one the intranet; instead, he says he'd write a physical sticky note and put it on his desk</p> <p>Facilitator action: Gives the user the next step in the multistep task because the goal of the study has not been completely reached yet (specifically, it's unclear if the user can find and use the calendar)</p>	<p>What we've learned: He realized there could be a calendar without being specifically told there is one, but could not find the calendar. We don't know if he can use the calendar.</p>
	<p>Task 4-B: <i>There is a way to schedule a meeting on the intranet. Can you use the intranet to make sure you keep January 12 from 2:00PM to 3:00 for a meeting with Richard Smith?</i></p>	<p>What the task accomplishes: Tells the user there is a calendar-like feature without specifically saying what it's called, where it is, or how to use it</p>
	<p>User action: Goes to the intranet's calendar and makes an entry</p> <p>Facilitator action: Stops</p>	<p>What we've learned: Once told that a calendar-like feature was available, the</p>

	<p>the multistep task here because all the relevant information has been collected</p>	<p>user found it and made an entry.</p>
	<p>User action: Says, “Okay, I guess there is a calendar,” looks all over for one, but doesn’t find it and gives up on looking for one on the intranet Facilitator action: Gives the user the next step in the multistep task because the goal of the study has not been completely reached yet (specifically, it’s unclear if the user can use the calendar)</p>	<p>What we’ve learned: Even when told that a calendar-like feature exists on the intranet, the user could not find it. We don’t know if he can use the calendar.</p>
	<p>Task 4-C: At this point the facilitator takes the person to the intranet calendar and gives him the same task: <i>Can you use this intranet tool to make sure you keep January 12 from 2:00PM to 3:00 for a meeting with Richard Smith?</i></p>	<p>What the task accomplishes: Brings the user to the calendar feature without explaining how to use it</p>
	<p>User action: Makes an entry in intranet calendar Facilitator action: Stops the multistep task here because all the relevant information has been collected</p>	<p>What we’ve learned: The user could not find the calendar; once taken to the calendar, he was able to use it to make an entry.</p>
	<p>User action: Cannot create an entry in intranet calendar or has issues doing so Facilitator action: Depending on the</p>	<p>What we’ve learned: The user did not realize there was an intranet calendar and could not find it after being told there is one. Once he was taken to the</p>

behavior observed, stops the multistep task here or continues giving additional subtasks to see the use of <i>individual</i> features within the calendar	calendar, he had issues trying to make an entry. (Thus, the design has problems relating to findability, navigability, and to the interaction design of the calendar feature itself.)
---	---

To seamlessly move participants to the next task in a set of stepped tasks, give them verbal reassurance. For example, you might say:

- *Thank you for doing that [as I write some notes to demonstrate that I am capturing what I saw]. I am getting a lot of very helpful notes. There is a way to schedule a meeting on the intranet. Knowing that...*
- *Thank you. That was very helpful. Do you have any comments about doing that? [Wait for comments.] I learned a lot from watching you do that. Now I am going ask you to try this. [Take them to the calendar area.]*

Then you could gesture to the print task:

Can you use the intranet to make sure you keep January 12 from 2:00PM to 3:00 for a meeting with Richard Smith?

Stepped Tasks to Test Features with Ubiquitous Labels

Stepped tasks are helpful in many cases, but the main situations when they are useful are when you want to avoid using a feature name in the task or you are testing discoverability or findability.

Discoverability of a UI feature or content refers to how easily users realize that the feature or content exists in the interface. Telling test participants upfront that a feature exists in the task instruction means that you miss the opportunity to learn whether they could discover it on their own. That's why, with stepped tasks, the first task is the broadest. But if participants don't discover the feature, you can slowly direct them to it in subsequent steps. At that point, you will have learned that the feature is not discoverable.

Findability of a UI feature or content refers to how easy it is for users to locate and access it, once they know it exists. Telling test participants upfront how to find a feature means that you miss the opportunity to learn whether they could find it on their own.

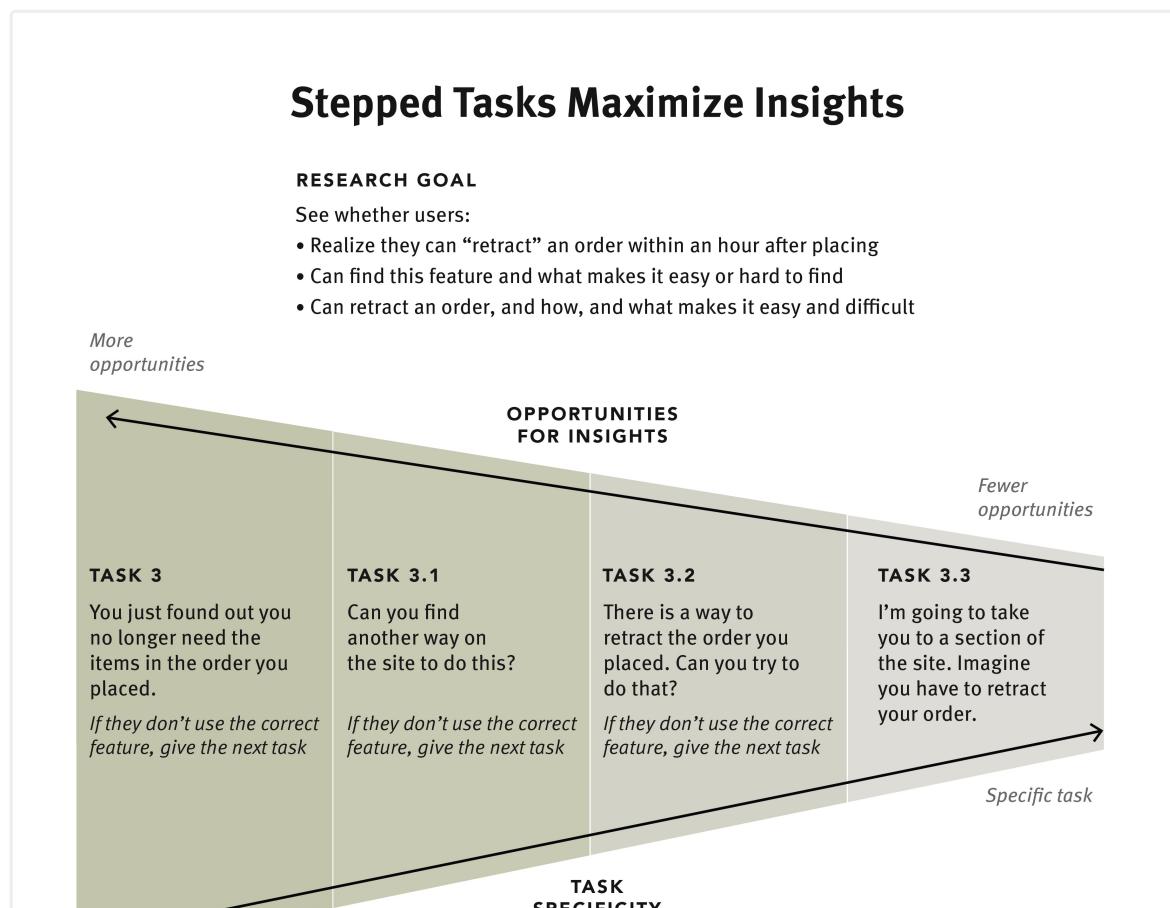
Avoiding Words in the UI

In general, stay away from using the same terms in the tasks as are used in the UI. This guideline

can be difficult to follow because ideally you have already done some user research and chose terminology that matches the user's mental model. Say for example, users think of the feature as a *graph*, so you named the button in your app *Create Graph*. It's hard to come up with a different word for the *graph* concept in order to test it. But it's worth doing it — because [using those same words in the task description](#) will [prime the user](#) to look for that name in the UI. Sometimes test participants will read a task and then type in the search box one or more words that appear in the instruction. Or, they will scan the UI for that exact word in the task description. Because of that, they may find the answer faster and the task may seem easier than it is, giving a false sense of confidence and satisfaction and misleading both the user and the researchers.

It can also be difficult to find a descriptive activity or phrase that replaces a commonplace word without generating an overly convoluted and abstracted task description. You may try it anyway, but also prepare a stepped task that either tries a different synonym or describes what the feature may do instead of mentioning its name. For example, to get participants to use a graph feature, you may give users numbers in a table and ask them to convey the data to a team, or show them a picture of a graph and ask them to reproduce it.

In our calendar example above, a calendar feature will help save time slots — hence the task instruction. The task doesn't mention the words *calendar* or *schedule* (a term used in the UI), so we can see if users would realize they have a calendar, think to use it, and find the command to create a meeting.





As stepped tasks become more specific, the opportunity to learn about discoverability and findability decreases.

Why Not Just Make Up a New Task During the Test Session?

You may be asking: Why not write broad task descriptions, and, then just wait and see what happens during the test? If users cannot find or discover the feature, the facilitator could react in the moment by adding a task as needed. That is an option, and sometimes, in fact, you will be forced to take this route because you didn't predict that there could be an issue until you saw it. But I prefer to avoid this approach when possible, for several reasons:

- **It's hard to make up a good task under pressure.** Crafting test tasks can take multiple tries and reviews from others. Doing it in seconds with the user watching and waiting, when you are also focused on all the other responsibilities you have as the usability-test facilitator, is difficult and can result in a task that is too leading, ineffective, or otherwise poor. And while the stepped task examples in this article seem simple, part of what makes them so is that the facilitator predicted them and is prepared for them, versus being surprised in the test and needing to react then.
- **A written task reinforces that a usability test is an exercise in observation, not a conversation or an interview.** It's difficult to watch people in a usability test struggle. But we usually need to do it in order to learn about the issues in the UI and later fix them so our many users don't also encounter them. Still, some test facilitators tend to help test participants too soon because they feel uncomfortable watching them struggle.

Reinforce Observation, Not Conversation

The more the [facilitator talks to the test participant](#), the more the participant gets used to it and expects the usability test to be a conversation or interview. A good usability test has a rhythm; it goes like this:

1. The facilitator hands the user a task.
2. The user reads it aloud.
3. The user works on it (usually while [thinking aloud](#))
4. The [facilitator stays mostly quiet](#).
5. The task ends.
6. The facilitator asks followup questions.
7. The facilitator hands the user the next task.

This rhythm is broken when the facilitator needs to make up a new task.

Handing users a written task reinforces to them that the usability test is mostly an exercise in observation by the facilitator, not an interview or conversation.

To a lesser degree, reading a prompt that you have prepared as a stepped talk will keep you from saying too much, or getting into the habit of assisting because you are uncomfortable.

Conclusion

Some companies spend a lot of money on research, including user testing, but still produce poor designs. There are many reasons for this state of affairs. One is that the tasks given in user testing are too focused — an issue that I have observed countless times in my many years of coaching and guiding other researchers. By “too focused tasks” I don’t mean *leading tasks*, — that is, priming users, giving instructions, or divulging something about the UI that jades the task. Rather, I mean sound tasks that don’t take into account factors such as the tested UI feature’s discoverability and findability.

Testing discoverability and findability isn’t easy, but we should still strive to do it. Broad tasks give the users the opportunity to discover and access UI features by themselves, make the testing less directed or leading, and give you more insights about how the user naturally interacts with the UI. But, if users don’t get to the feature of interest on their own, a stepped task can bring the user there so you can also learn about the feature’s usability. Since each test participant is expensive, we want to wring as much insight as we can out of every session.

Related Courses

Discovery: Building the Right Thing

Conduct successful discovery phases to ensure you build the best solution

RESEARCH

User Research Methods: From Strategy to Requirements to Design

How to pick the best UX research method for each stage in the development process

RESEARCH

ResearchOps: Scaling User Research

Orchestrate and optimize research to amplify its impact

RESEARCH

MANAGEMENT

MANAGEMENT

Learn More:



**Always Pilot Test
User Research
Studies**
Kim Salazar · 3 min



**Level Up Your Focus
Groups**
Therese Fessenden · 5
min



**Inductively
Analyzing
Qualitative Data**
Tanner Kohler · 3 min

Related Articles:

**Open-Ended vs. Closed
Questions in User Research**
Maria Rosala · 5 min

**UX Research Methods:
Glossary**
Raluca Budiu · 12 min

**Cognitive Mapping in User
Research**
Sarah Gibbons · 16 min

**Formative vs. Summative
Evaluations**
Alita Joyce · 5 min

**Iterative Design of a Survey
Question: A Case Study**
Feifei Liu · 8 min

**When to Use Which User-
Experience Research
Methods**
Christian Rohrer · 9 min

Subscribe to Our Newsletter

Get weekly UX articles, videos, and upcoming
training events straight to your inbox.

Subscribe

Follow Us

[LinkedIn](#)

[Youtube](#)

[Twitter](#)

[Instagram](#)

[Podcast](#)

Certification

What is UX Certification?

Specialties

Exams

UX Certified People

UX Training

Virtual Training

In-Person Training

Browse Courses

Course Calendar

Consulting

Expert Review

User Testing

Customized Research

Team Training

Applied Workshops

Keynote Speaking

Free Guidance

Articles & Videos

The NN/g UX Podcast

About

[Why NN/g](#)

[About Us](#)

[People](#)

[Contact](#)

[Return Policy](#)

Copyright © 1998-2023 Nielsen Norman Group, All Rights Reserved.

[Cookie Preferences](#)

/

[Cookie Declaration](#)

/

[Privacy Policy](#)