What is usability testing?

Usability testing is a method for evaluating a product or service to ensure that it is usable for the target audience. During a usability test, participants complete tasks with a prototype or live software while a facilitator observes, asks questions, and takes notes. Usability tests are designed to simulate real-world interactions between users and a product or application to identify usability problems and areas of opportunity.

Usability testing is most useful when you need to:

- Test an existing product or a new design.
- Test interactions, wording, and design elements to make sure they are user-friendly.
- Identify confusion or issues.
- Understand the actual behavior and preferences of users.
- Discover opportunities to improve.
- Want to benchmark task performance for future comparison.

You don't need to have fully functional software to perform a usability test. In fact, you can test really rough concepts, even things you've sketched on paper! This saves tons of time, and makes it easier to change direction without feeling like you've wasted a lot of time. Low-fidelity software prototypes can be created using design tools like Sketch or Figma.

Usability testing should not generally be used as a discovery method or to collect quantitative data from a large sample of people. The table below shows how usability testing fits into the larger picture of user research.

Phase:	**Discovery**	**Exploratory**	**Testing**	**Listening and measuring**

When you need to:	Find out how people are doing things now Learn about the real users and environment	Understand opportunities and gaps Prioritize who you are supporting and what their goals are	See how people will react to something Find ways to make something more user-friendly	Understand why they feel the way they do Establish metrics to measure the impact of changes
Appropriate tools can include:	Field study Diary study Competitive analysis Stakeholder/User interviews	Journey maps/Service blueprint User stories Survey Card sort	**Usability / Prototype Test** Accessibility test Heuristic evaluation Tree jack Benchmark test	Semi-structured interview Sentiment scale (e.g. NPS, SUS, etc.) Telemetry/Log analysis
Outputs can include:	Insights summary Competitive comparison Empathy Personas	Dendrogram Prioritized backlog items Structured data visualizations	User quotes /clips Usability issues by severity	Satisfaction metrics Broad understanding of user behaviors

	Distinguish what people say from what they actually do		Where people click/look for things Measured impact of changes in product	

Why do usability testing?

Usability testing helps reduce a software project's risk by giving designers an opportunity to validate their work prior to investing in development. It enables one to explore and learn about users' existing mental model, understand user preferences, and identify blind spots that designers may have. Performing usability testing against software applications after their release can be a way to benchmark performance and track changes from release to release.

Watch this short video by Jakob Nielsen talking about why usability testing is such a powerful tool in the UX toolkit.

How to perform a usability test

Preparing for a usability test

Start to prepare for a test by forming a hypothesis about what details will matter the most to your design and decision-making. Then choose tasks that line up with those goals. In selecting features to test, consider:

- Frequency of use.
- What's new or or will be highly publicized.
- Known trouble spots.
- Risk related to incorrect usage.
- Importance to the customer.

Depending on the answers to the above questions, consider what you will test and who you will test it with. Can your goals be accomplished by testing a low-fidelity prototype? What tool will you use to build that prototype? Who are your target users and how can you find them? Below we will discuss these topics in more detail including how to recruit test participants, why your test script is important, how to create a test script, how to moderate a usability test, and what to do with your results.

Recruiting usability test participants

In order to conduct a meaningful usability test, you must know who your users are. Where possible, you want to test with actual users/customers or people who could potentially fit into the target user group. If you are uncertain about who to test with, further discovery research and/or discussion with the product team might be warranted. You may choose to create a screener questionnaire to help identify people who might fit into the target user group for your application.

Decide whether you will be testing in person or remotely. Remote testing can be easily conducted using any internet conferencing platform, which opens up your activity to a much larger pool of potential participants. If you are doing a remote test, make sure participants can access the conferencing platform and have access to any software or prototypes prior to the start of testing to ensure that valuable testing time will not be wasted on solving setup problems.

For simple qualitative usability tests, it has been shown that 4-6 participants are enough. A study by the Nielsen/Norman Group found that most usability problems in any given application are found after testing only 6 participants. But keep in mind that more participants may be needed if you are wanting to test multiple personas and roles, or if you plan to gather quantitative data.

Recruiting can be one of the most difficult and time-consuming parts of usability test planning. As is the case for conducting any external research, you should review Red Hat's policies on handling personally identifying information (PPI) and providing compensation and gifts to participants.

The usability test script

The test script is a structured document that outlines the test, the tasks that users will be asked to perform, follow-up questions, and success criteria. It includes:

1. General introduction
2. Background questions
3. Tasks for testing
4. Summary questions

Preparing a test script prior to testing helps you to think through the questions you will ask, what tasks or activities you intend to test, and maintain consistency across test participants so that results can more easily be compared and summarized when done.

1 - Introduction

The introduction is used to set context and make the participant feel comfortable. it should:

- Explain what the application is and why the participant is there.
- Brief the participant on what's coming and express the need for them to think out loud
- Remind them that the software is being tested, not them, and that there are no right or wrong answers.
- Obtain permissions to record, inform them of their privacy rights, and collect any necessary paperwork.

2 - Background questions

The activity should start with some questions to get the participant talking and collect some background information. Typical background questions should focus on things like:

- Habits and behaviors
- Background or education
- Preferences
- Experience using this or similar applications.

3 - Tasks for testing

Tasks are the heart of a usability test. You will want to construct reasonable scenarios that you would expect users to engage in and observe participant performance to identify potential usability problems. Task scripts should be written to be goal-oriented and give participants instructions about what they

should accomplish, not how to do it. For example, if you are writing a usability test script for a Japanese restaurant's website, the task description below might be appropriate.

> **Task 1:** You have heard good things about the vegetable sushi. Find one and show me how you would order it. Once you are ready to check out, let me know that you're done.

Note that this says nothing about *how* to find an order a vegetable sushi entree. It expresses only what users will want to accomplish and simulates what someone would do if they were attempting to complete a take-out order on their own.

It's also useful to place tasks in a realistic sequence, if possible. For example, if you want to test how easily the user can configure an application to start using it, ask them to do that first before giving additional activities. We recommend limiting the number of tasks to 8 or less so that the participant does not feel overwhelmed.

You might want to start by asking for first impressions before they attempt to complete the first task. You can do this by starting the participant on a landing page and asking them to look around and describe what they think they could do before giving them specific tasks to carry out. Ask questions like:

- Can you explain what this software is, what can it be used for?
- What are your eyes drawn to first?
- What do you expect that you can do and/or what do you expect that you would do first?

Following the completion of a task, you should plan follow up questions that probe into specific areas of concern. If participants struggle, always try to understand what they expected and how that differs from the current design. This will give insight into the participant's mental model and where key disconnects between the user's model and the system model may lie.

4 - Summary questions

At the end of the activity, you should plan for some summary questions that ask the participant to reflect on their experience and call attention to both positive and negative attributes of the software. For example, you might:

- Ask the participant for their overall impressions - what they liked and what they didn't like.
- Revisit any areas that need more exploration.
- Solicit any ideas the user might have to make the experience better.
- If applicable, show alternative designs for feedback.

If observers are present, this is the appropriate time to allow them to ask questions. And at the end of the activity, always thank participants for their time!

To see how this is put into practice, view an actual usability test script used to test a prototype of a new tool targeted at enterprise developers.

Moderating a usability test

The usability test moderator oversees the activity to make sure it runs smoothly and yields objective results. They:

- Make sure the participant is comfortable and knows what to expect.
- Ask probing questions.
- Solve any technology issues that happen during the test.
- Gather and record task results. Consider what to do if a user fails to complete a task.
- Make sure observers know how to behave. Call them "note-takers!"

The moderator should be sure to ask open ended and unbiased questions to get a participant to explain what they are thinking and express their opinions regarding the current design. Sample questions include:

- What draws your attention on this page?
- What are the most important elements on this page?
- I see that you seem confused. How did you expect this feature/function to work?
- Are there any interface elements that don't make sense?
- It looks like you're not finding what you expect. Can you tell me what you are hoping to find, and where you expect to find it?
- If you were looking for more information, where would you expect to find it?

Usability test moderation is built on a foundation of good interviewing skills. For more information about how to construct good interview questions, see the User Interview method for more guidance.

Keep in mind that what people do is often more revealing than what they say. Look and listen for things like:

- **Confusing terminology:** "So restart would not be the right word for me. I think refresh would be better…"
- **Redundant workflows:** "I would have wanted just to download the file. This seems a little extra and a lot of work"
- **Things people overlook:** e.g. if some didn't notice the download option.
- **Preferences/alternate workflows:** "I prefer typing to selecting from a long drop-down list so I can finish the task faster."
- **User expectations** compared to **actual experience**.

Depending on the type of test you are running, there are a number of ways that you can record participant performance. You can:

- Capture verbatim comments.
- Ask the participant to rate the ease and efficiency of interactions. Two common approaches for this are:
    - System Usability Scale
    - UMUX-lite
- Rate the ease of use based on *your* observation.
- Record quantitative metrics like:
    - Time to complete each task.
    - Number of errors or clicks per task.
    - Success rate.

You may also consider having a dedicated note-taker who is observing the test and can relieve you of the burden of moderating the test and taking notes at the same time. Here are some additional best practices you should keep in mind if you are moderating a test.

DO	DON'T
• Remind the participant to think out loud. • Have the participant read back the task. • Allow them to say when they think they're done. • Let them struggle a bit. Ask them "What are you trying to do" or "what were you hoping to see there?" • Ask them what THEY think this does • Ask open questions, like "How often, if ever, have you…" • Ask specific questions like "What catches your eye?" or "Is there anything else you wish was on this page?" • Embrace awkward silence, let them talk, ask "What else?" LISTEN. • Probe non-verbal queues.	• Tell them "you did it!" • Rush them along, help them, OR force them to keep struggling for too long. • **Do a demo** • Ask leading questions like "Do you like it?" Or "Do you think this is a problem?" • Talk over them or explain the design rationale. • Start with complex tasks that might discourage the participant. • Call it a "test". • Invite a bunch of folks who will defend their work or interrupt.

Analyzing and reporting usability test results

It's important to collect and organize individual test findings in a way that can help you identify top problems and trends. Key sources of data to feed your analysis include moderator notes, observer notes, and recordings. It may help to give your observers a template to help them know what to look for and to ensure that observations are recorded consistently. User Interviews provides some templates for UX research note taking. You may also want to develop your own.

In analyzing usability test findings, your goal is to identify trends and problems that are repeatable across tests. You should:

- Review and group similar observations or problems.
    - Look for similar issues that originate from the same cause.
    - Group issues using affinity mapping or a similar technique.
- Eliminate groups with only 1-2 data points.
- For each group, try to summarize the problem in one short statement.
- Consider the severity of each problem. Does it prevent users from being successful reaching their goals or is it just a minor annoyance?

Finally, you'll want to summarize your key findings in a final report and support them with evidence where possible. Your final report can be more or less formal depending on your audience and should include:

- An **executive summary** that enumerates key findings and recommendations. It should:
    - List problems and recommendations in priority order
    - Report both strengths and weaknesses
    - Use video clips to support your findings and recommendations
- A **participant summary** to provide readers with an overview about who you tested. Note that this summary should NOT include any personal identifying information (PII).
- **Task descriptions and results.** These may include, but are not limited to:
    - Success rates
    - Times/clicks
    - Problems found
    - Direct participant quotes (or, video clips of people struggling!)
    - Recommendations