Conducting Usability Testing

An overview

1 Version

Version	Summary Change History	Updated by	Version Date
1.0	Initial Documentation	MZillhardt	04/01/2020
1.5	Expansion on concepts	MZillhardt	09/29/2020

2 Overview

Usability testing is all about getting real people to interact with a website, app, or other product and observing their behavior and reactions to it. Whether by watching session recordings or renting a lab with eye-tracking equipment, usability testing is a necessary step to make sure we build an effective, efficient, and enjoyable experience for our users.

Why?

Usability tests are run for many different reasons:

- To validate a prototype
- To find issues with complex flows
- To gather unbiased user opinions
- To get the insights that help create a better overall user experience

Usability testing can help us answer questions like:

- Why users come to a site
- What users are trying to achieve
- What other sites or products users might visit or use before or instead of ours

It can also help use identify any barriers that keep users from completing a task

- What is stopping users from completing a task
- What can they not find on a page
- Do they find wording or navigation confusing

Testing can also help us pinpoint what hooks that persuade users to take actions

- What is convincing them to take action
- What did they like most about the site

We can also measure how valuable the product is, and measure attitudes towards products

- Do users understand the point of the site
- Is it easy to navigate
- Would they use the site again

We use UX design because of the benefits that it generates - happy customers and increased sales. This happens when UX goals are met and the goals of the user align with the goals of the business.

3 Steps of usability testing

1 Create a test plan

First it is necessary to create a test plan. This lays out key parts of the testing and establishes not just what we are testing, but why we are testing it and what initial questions we are trying to answer. Testing will likely lead to more questions that we can explore further.

Parts of a test plan:

1. Product under test: What's being tested? What are the business and experience goals? Eg, reduce support calls or sell products online
2. Business case: why we need to test. Review expected benefits of testing and potential costs of not testing
3. Test objectives: goals of test. What specific questions will be answered? What hypotheses will be tested?
    a. Aspects of application that are of concern
    b. Tasks we think are difficult
    c. Groups of users we are worried about
    d. Feedback from users: phone calls, social media, email, etc
    e. Concerns of client/management/business
    f. Problems raised by design/development/marketing
4. Participants: outline who will be the demographic we will be testing, eg, providers, students, repeat customers, multi-plan users, etc
5. Test tasks: test specific goals, eg purchase a plan, get a quote, etc
    a. Self-generated tasks, asking what the user wants to do, are helpful as well
        - Interview participants to find out what they expect to be able to do, then ask them to be more specific as to what, eg purchase a plan -> purchase short term plan for family of four
    b. Part self-generated, telling user what you want to test but then fill in gaps
6. Responsibilities: who will recruit participants? Who will moderate the test? Who will provide technical support?
7. Procedure: outlines the agenda and show the play-by-play of the activities in the test itself
8. Location and dates: outline when and where the testing will occur

This is simply an outline for the testing plan. It is also necessary to have

- A post-test questionnaire
- Screen shots for annotation or a datalogging system

- Decide on what type of testing we will do

Types of testing

Moderated vs unmoderated

- Moderated testing requires that a moderator be present to conduct the testing. The moderator needs to have a clear understanding of the product as well as the people skills to be able to conduct the research. A moderated test allows researchers to ask follow-up questions, as well as dig deeper into user behavior and why they are engaging in the behavior they are. Users also have support if they do have questions trying to navigate a new product. Moderated tests are good for initial phases of new products with an incomplete UI.

Unmoderated testing

- Allows for more participants and can recruit participants easier. Unmoderated testing also allows for less chance of observer bias. Because they do not require a moderator, theses are quicker and cheaper to conduct, and can be better for tight deadlines. These are better for the end stage of production, where UIs have been already established, or already built products.

Remote vs in-person

- Compared to remote tests, in-person tests provide extra data points, since researchers can observe and analyze body language and facial expressions. However, in-person testing is usually expensive and time-consuming.
- Remote testing doesn't go as deep into a participant's reasoning, but it allows test large numbers of people in different geographical areas using fewer resources.

Explorative vs assessment vs comparative

- Explorative testing is open ended. It simply gauges a user's overall feeling of a product. This helps us find gaps and identify new features.
- Assessment testing targets a specific area and sees how well a user was able to use it, eg claims process or how to get a quote.
- Comparative testing involves seeing what solutions they prefer. This could be done via a/b testing methods or by comparing usability between competitive sites/products.

2 Recruit participants

Second it's necessary to recruit participants. Depending on the goals, who is recruited and how is important.

- Using data loggers on a site / product, allows us to see how users interact with the product behind the scenes. This would be an unmoderated testing, but we could still be able to target specific areas of the site for analysis.
- We can recruit users by having a pop-up poll to find users that want to participate in this study. That way we can target specific demographics and set specific tasks for them to accomplish. This would be a way to have targeted, moderated testing but do it remotely.
- Hire an agency to do it. This would be the most costly option, but would allow for targeting very specific subsets of a demographic.

3 Design the tasks

Third it's necessary to design what tasks the users need to go through to meet the business goals. It's necessary to plan the specific scenarios you'll take the participants through, and the tasks participants will be required to complete. Example scenario:

- *You're a student, and are planning a to do study abroad semester. You find out that you need to have some kind of health insurance for the duration of this. How do you go about it?* Participants will then go to the site and try and purchase a plan. We can then gauge if they are trying to find a quote, find out what plans work for them, or if they are able to even purchase a plan.

4 Running the test

Fourth it is time to do the actual usability test. Depending on the parameters we've established in the test plan, who we've recruited, and the tasks we've laid out, the usability test would be very different. Remote testing is different form in person, and moderated testing is different from unmoderated, however in any scenario it is necessary to ensure that all equipment is working, the users can clearly hear the moderator or if the testing will be unmoderated, have access to the protocols and the test tasks.

5 Analyze the test data

At this point after testing it is necessary to organize findings and looking for patterns and recurring issues in the data. The test plan can be of help here, because that's where we outlined exactly what we need to test and what questions we need to answer. We can review the problem areas, pain points, and business requirements here. Create categories of interest. Most likely, each category will correspond to one of the tasks that we asked users to complete during testing.

a. Review each test session. Review your testing sessions one by one. Watch the recordings, read the transcripts, and carefully go over your notes.

   Look for:

   - Issues the user encountered while performing tasks
   - Actions they took
   - Comments (both positive and negative) they made

b. For each issue a user discovered, or unexpected action they took, make a separate note. Record the task the user was attempting to complete and the exact problem they encountered, and add specific categories and tags (for example, location tags such as *check out* or *landing page,* or experience-related ones such as *broken element* or *hesitation*) so you can later sort and filter. Make sure statements are concise and exactly describe the issue. Data may look like this:

User ID	Category	Task	Problem	Tag1	Tag2
1	quote	Get a quote	User unable to determine what type of plan they need	filter	
1	eclaim	File a claim	User dropped off at file upload		

c. Draw conclusions based on both qualitative and quantitative results. Quantitative analysis will give you statistics that can be used to identify the presence and severity of issues. Qualitative analysis will give you an insight into why the issues exist, and how to fix them.

   Quantitative metrics could be:

   - Success rate: the percentage of users in the testing group who ultimately completed the assigned task

- Error rate: the percentage of users that made or encountered the same error
- Time to complete task: the average time it took to complete a given task
- Satisfaction rankings: an average of users' self-reported satisfaction measured on a numbered scale

Qualitative metrics helps to illustrate why certain problems are happening, and how they can be fixed. Issues involving the same tasks, eg how many users experiences the same problem with a certain step such as getting a quote, will help us conclude that there is an issue that needs to be resolved.

d. Rank issues based on their impact. Consider how global the problem is throughout the site, and how severe it is; acknowledge the implications of specific problems when extended sitewide (e.g., if one page is full of typos, you should probably get the rest of the site proofread as well).

   Categorize the problems into:

   - Critical: impossible for users to complete tasks
   - Serious: frustrating for many users
   - Minor: annoying, but not going to drive users away

e. Report on the results.

   - Showcase the highest priority issues.
   - Be specific. It's not enough to simply say “users had difficulty with entering payment information.” Identify the specific area of design, interaction, or flow that caused the problem.
   - Include evidence. Snippets of videos, screenshots, or transcripts from actual tests can help make your point.
   - Present solutions. Brainstorm solutions for the highest priority issues.
   - Include positive findings. In addition to the problems you've identified, include any meaningful positive feedback you received. This helps the team know what is working well so they can maintain those features in future website iterations.