



A crash course on remote, moderated usability testing

Sarah O'Donnell

UX Researcher at GitLab

The value of remote, moderated usability testing

Why should you conduct usability testing?

- Make changes based on data and behaviour, not assumptions.
- Spend time building features your users want.
- Launch features faster.
- Attract and retain users.

What are the advantages of conducting usability testing remotely?

- Greater diversity.
- More cost-effective.
- Users are in their natural environment.

Create your screener

A screener consists of:

- A brief overview of the study details (technical requirements, incentive, etc).
- Availability
- Permissions
- Contact information
- Filtering questions

Filtering questions

- Write a list of who you want to talk to.
- Work out what criteria will identify these people.
- Write a question to address each criteria.

For example:

Who: People who contribute to open source projects.

Criteria: People who have contributed to an open source project in the last month.

Poor question: Have you contributed to an open source project in the last month?

Good question: When did you last contribute to an open source project? (Within the last month/2-3 months ago/4-5 months ago/Over 6 months ago/I've never contributed to an open source project.)

Find the right people

- Use Google Forms to create your screener (<https://www.google.co.uk/forms/about/>)
- Share via social media, email and within your product.
- Create a research panel (<https://about.gitlab.com/researchpanel/>).
- Test with 5 users (<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>).
- Confirm users' participation (remind users of study details).

Writing a script

A script consists of:

- Introduction
- Warm-up questions
- Tasks
- Researcher prompts
- Wrap-up questions

Introduction & warm-up questions

Introduction

Aim: Build rapport with users.

What to include: Introduce yourself, explain what you're doing today and what's expected of the user.

Warm Up Questions

Aim: Get users talking. Build a better picture of who they are as a person.

Examples: Where in the world are you calling me from? What's the weather like there at the moment? What kind of work do you do? Have you always had this job? What are the things you most like about your job?

Tasks

Aim: Test the usability of a feature.

1. Write a list of what features/user goals you want to test.
2. Create a realistic scenario as to why a user might need to use that feature.
3. Don't lead users. Avoid using words found in the interface.

Uber Example:

User Feature/Goal: Contact driver about a lost item (these are the actual words used within Uber's interface).

Scenario: You left your umbrella in a taxi you took last Wednesday at 1:00pm. How can you find out if it's still in the vehicle?

Researcher prompts & wrap-up questions

Researcher prompts

Aim: Encourage quiet users to think out loud.

Examples: So what happened there? Was that what you expected to happen? Why or why not? Did you find what you were looking for? Is there anything else you could do at this point? What additional information could have helped?

Wrap-up questions

Aim: Gauge users' overview thoughts and opinions about what they've just seen.

Examples: If you could change one thing about this feature what would it be and why? How would you describe this feature? Who do you think the feature is designed for? Is there anything else you want to share with me today?

Testing day

Before you speak with users:

- Minimise distractions
- Hide anything you don't want to share on your computer.
- Do a test run. I use Zoom to run usability testing sessions.
- Grab a notebook and pen
- Get a glass of water

When you speak with users:

- Double check permissions
- Remain objective
- Try not to lead users
- Stay mostly quiet
- Watch the time
- Keep smiling! :-)

Analyse & share your results

- Edit videos. Ensure users cannot be identified.
- Upload the videos to a location where they can be watched (Google Drive) and viewers can leave comments (GitLab issues).
- Ask everyone to:
 - Write down important insights (good or bad) learned from each user.
 - Make a note of any ideas for solving problems.
 - Set a deadline to do this by.
- Consolidate everyone's comments.
- Look for recurring problems.

Thanks for listening!

- Tools I've mentioned today
 - Google Forms (<https://www.google.co.uk/forms/about/>)
 - Zoom (<https://zoom.us/>)
 - GitLab Issues (<https://about.gitlab.com/>)

- **Questions?**



@hellosarahod



hello@sarahod.co.uk