USER TESTING CAN BE FUN

A small FAQ for small open source teams

Hex the squirrel

I already do a lot of testing, where does user testing fit in?

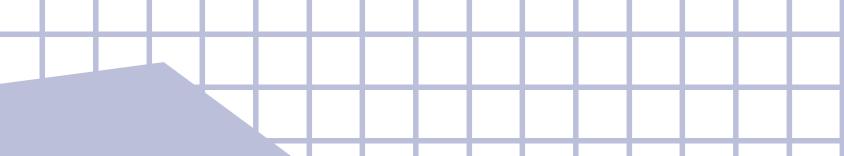User testing improves the design, leading to a better experience for your users.

For example:

You can find ways to improve the app that are high impact & low effort.

You can decide what to prioritize next.

You can discover features that users expect to be there, but aren't.

You can test a new design on a sketch pad, avoiding non user-friendly features before you code.

What is user testing?

User testing is a scenario-based method to get feedback about a specific area of your software.

Steps:

1. Find a handful of users (or less)
2. Watch them use your software one by one
3. Note your observations
4. Identify the commonalities
5. Make improvements

We have a big backlog of issues and emails already! Why should I find more problems?

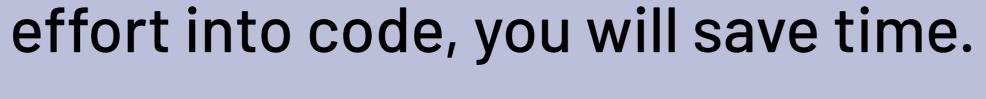Testing can help you prioritize your backlog and find improvements that will make the biggest impact.

You can find out what are the most interesting or common issues amongst a particular user group.

And you get to control who your target users are, rather than the loudest voices on the mailing list.
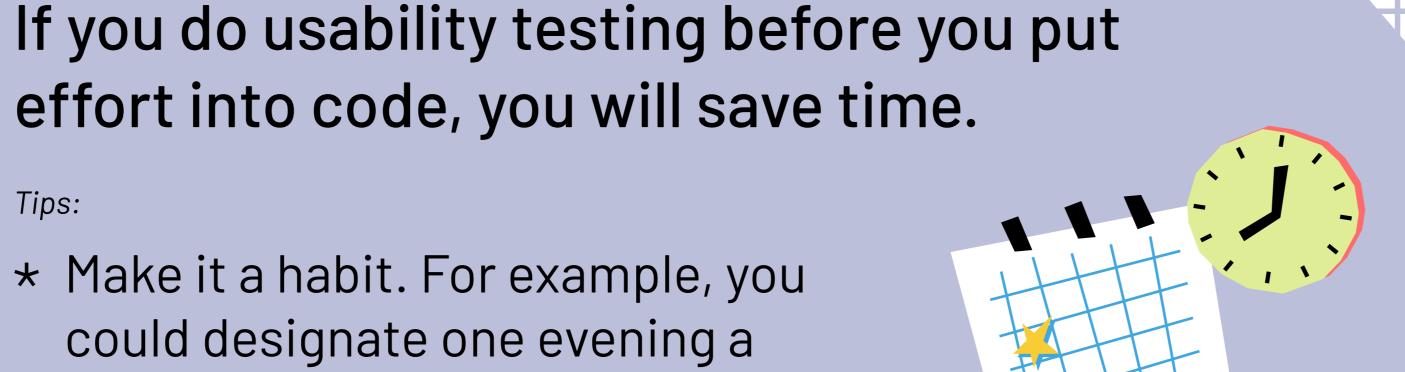
How much time will this take? I'm already very busy.

If you do usability testing before you put effort into code, you will save time.

Tips:

★ Make it a habit. For example, you could designate one evening a month for testing.

★ At a conference, hackerspace, cafe, crafting event, workshop, rave, etc? Even though 30-45 minutes is optimal, even 15 minutes of testing can make a huge impact.

What are we testing?

Pick a scenario that can be tested in 30-45 minutes.

Examples: sign up, onboarding, make an appointment

Do you have ideas that you're not yet 100% sure about? Use paper to test sketched ideas to figure out the effect.
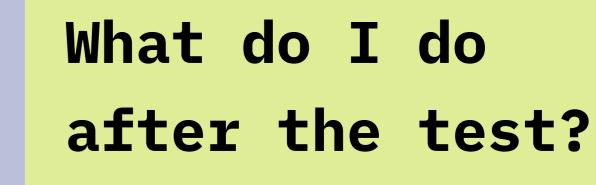
How can I make the testing representative?

It's best to test with participants who represent your users.

To recruit, get the word out! Advertise that you're looking for participants, and let them know the basic details.

The ideal number is five participants. With more than five, the results become redundant. Even so, one test is better than zero tests.

Should I tell them how to use my software?

Try not to explain your software, correct their mistakes, or defend your choices. Your job is to observe and your participant's job is to be honest.

Remember:

You can use some time at the end of the session to clear up any critical misunderstandings.

What do I do after the test?

Think through what you observed.

★ Did you notice commonalities?
★ What was confusing to your participants?
★ What worked well?

Share with your team & community!

Keep it short and answer these questions:

★ What did you do?
★ What did you learn?
★ What are you going to do now?