How to Usability Test Your Content

# 1. Decide which topics you want to test.

* You typically test task topics, though test participants might also choose to work with concept or reference topics if they need more information.
* Consider choosing topics that:
  + Most users will work with.
  + Are critical to a customer's success—for example, topics about product configuration or security.
  + Require users to interact with a challenging part of the UI.
* You can test more than one topic during a session, but be conscious of your participants' time. Consider how long it will take participants to work through each topic, and plan accordingly. The maximum length for a usability test should be 60 minutes.
* Try to choose topics with a range of difficultly, and test the easier topics at the beginning of the session. The goal is to manage participants' level of frustration. Extremely frustrated participants tend to give less constructive feedback.

# 2. Write tasks and create a test handout.

* Because of our DITA strategy, one topic equals one task.
* When writing the tasks, use the user's language, rather than the language of the product or the documentation. For example, if you were to test MadCap Flare's topic, *Adding New Snippets*, the task might be "Make a sentence available for reuse," rather than "Create a snippet."

Tasks should be specific enough that the participant knows what you're asking them to do, without you needing to prompt them. In the above example, you would supply the sentence that the participant will work with, and tell them where to find it.

* List the tasks on a handout, which participants can reference while testing.

For more information, see *Sample-handout.docx.*

# 3. Find participants and schedule time.

## Finding participants

* Usability testing is valuable, even if you test with only one person who isn't part of your target audience. That being said, ideally, try to find 3 or 4 participants that:
  + Are varied in age and gender.
  + Have the software experience that you want to test for.
  + Have the domain knowledge necessary to complete the tasks. For example, participants might need to understand healthcare terminology.

If you can't find users that have the appropriate domain knowledge, do your best to supply necessary information during testing.

* If you plan to record the test sessions, make sure you get each participant's consent in writing.

You can record using Go to Meeting or—if the test session is in-person—you can download a free screen recording software, such as iSpring's Free Cam.

## Scheduling time

* Once you find participants, send them a meeting invite for a mutually agreeable time.
* Plan where you will hold the test sessions:
  + For sessions that are in-person, try to find somewhere quiet, where you won't be interrupted. (I've learned from personal experience that the kitchen is not a good place to test.)
  + For sessions that are remote, you can use Go to Meeting. Make sure that participants have the meeting URL, and that you're set up to proctor the call.

# 4. Write the test script.

* During the test sessions, you read from a script to ensure that you're providing a consistent experience for all participants.

For more information, see *Sample-script.docx*.

* In the script, specify the materials that you need for each test, as well as any preparation activities.
* The way you write the script is a matter of personal preference.

I have an easier time reading scripts that are written in full sentences, but others might prefer a bulleted list. As long as you provide all participants with the same information, the format of the script is up to you.

# 5. Build the Flare project and save a copy of the output.

* Before you build, hide any content that you don't want test participants to see.

In my experience, it's best to overestimate the amount of time this will take, especially if you're testing an early draft of a help system or *pdf*.

* After you build, copy the files in the Output folder to your local machine. Use your local copy of the help files each test session.

Using the local copy ensures that all test participants work with the same version of the documentation, even if you have to build again before all the test sessions are completed.

# 6. Do the test once by yourself.

* Spend a few minutes running through each task as if you were a test participant. Verify that:
  + The test takes the amount of time that you anticipate.
  + Your content appears as expected.
  + Participants can complete the tasks without encountering bugs in the software.

If you encounter a bug, but can still complete the task, plan to help participants work though it during the test.

# 7. Hold the test sessions.

* Print the test handout and script, and gather other necessary materials.
* Take notes during each session.
* When speaking with test participants, do you best to be impartial.

Some great guidelines are available here: <http://www.sensible.com/downloads/things-a-therapist-would-say.pdf>.

# 8. Share your findings.

* If you think other members of the Documentation Group would benefit from your findings, write up a list and share them on Teamer, in the Usability Testing task.
* If the test revealed usability issues with the software, consider sharing your findings with the product stakeholders.

# Additional Resources

* Steve Krug's website:

<http://www.sensible.com/>

* *Don't Make Me Think* by Steve Krug

Chapter 9 discusses usability testing.

* *Running a Usability Test*:

<https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>

* *How to Conduct a Usability Test from Start to Finish*:

<https://uxmastery.com/beginners-guide-to-usability-testing/>

# Download iSpring

You can download iSpring's free screen recorder here:

<https://www.ispringsolutions.com/ispring-free-cam>