Module 10: Conducting Usability test

1

Last week

- Usability testing dashboard
- Task -- > Scenario tasks
- Participant overview

2

This course (in chart form)

Dates	Topic	Assigned	Due
Module 8	Introduction: Definitions, Heuristics	**Group project**	
Module 9	Planning and preparing Usability Testing	**Brief**	
Module 10	Collecting data - Conducting Usability Testing	**Report**	**Group project**
Module 11	Usability Test Analysis and Reporting		**Brief**
Module 12	Accessibility: The DOM and assistive devices		
Module 13	Content & Inputs		
Module 14	Presentations		**Report**

3

The Basics

1. Objective/Scope/Questions: what are you going to accomplish? What are you NOT going to accomplish? What are you trying to find out in order to accomplish your objective?
2. Participants: who is going to take the test (and why have you selected them)?
3. Methods: how are you going to generate data to answer your questions?
4. Metrics: how are you going to measure your data?

4

I think, because I talk about these assignments so early in this course, people say, "run some tests, report back, gotcha", without understanding **what kind of tests they are running**.

5

For the people who feeling a bit overwhelmed, let me say the following:

Usability testing **is not quality assurance testing**. Your site/app/whatever should already work *as you intend it to*.

User-testing as QA is a waste of time and resources.

6

Technical variables are bad.

Test users on the **same** browser/device/operating system wherever possible. This will increase confidence that your data says something *about your users*, and not about their particular technical environment.

7

Time is a lousy metric. Oh, it took users an average of 2.4 seconds to find the right button? Why do we care?

8

Assumptions about what's good or bad are not good, they're bad. Every year, I see people put in their tests plans some variation of "[arbitrary value] will be considered a pass". Says who? Why? Who cares?

9

3. Method

10

Moderated vs Unmoderated Testing

- ***Moderated usability testing*** - requires active participation of a qualified moderator using a pre-written test script to instruct participants on how to proceed with the testing.
- ***Unmoderated usability testing*** - not supervised. The test is prepared in advance, and participants are responsible for completing it on their own.

11

Advantages of moderated usability testing

- **More engaged participants** – builds trust and motivates participant to complete task
- **Collect additional information –** moderator can ask custom follow-up questions to help explain participant's actions
- **Ability to read the situation** – if the participant is unsure what to do next, the moderator can step in and ask

12

Step-by-Step Instructions for Conducting the Usability Test

- Set up test environment
- Participants: explain the purpose of the test, brief them on the tasks
- Important to have a script to ensure that there is consistency across all test sessions

13

Test procedure : moderator's role and duties

- Responsible for facilitating the test sessions, interacting with participants and guiding them through the tasks
- observe and take notes on participant behavior, interactions and usability issues

14

Test procedure: Script for Participant introduction and information

- First interaction with participants sets the stage for the usability test
- Script: overview of the tasks, specific instructions and addressing any questions or concerns
- Script ensure consistency and standardization throughout the usability testing process. This allows for accurate data collection and analysis and ensures fairness across all participants.

15

The usability test procedures provide the framework for conducting successful test sessions and gathering valuable insights into the user experience.

It's important for the procedures to be consistent in order to **maintain the integrity of the test**.

16

- **Moderated in-person user testing is the gold-standard of usability research**.
- allows for the highest assurance of identifying usability issues within a digital interface

17

Quantitative or qualitative

- **Qualitative research** involves collecting data through direct observation of a small group of people in order to assess behavior and provide context.
  - "Why?"
  - "How?"
  - "How can we fix this?"
- Qualitative methods produce unstructured or semi-structured observational findings like comments, preferences, and motivations.
  - "How much?"
  - "How many?"
  - "How often?"

18

Testing types and tools include:

- **Heuristics** (don't forget to say which heuristics - as we learned, there are many different heuristic checklists),
- **Eyetracking** (not recommended - I haven't found a webcam-based eyetracking solution that wasn't super janky, and, as we'll see shortly, you need **a lot** of users for your data to hold up),
- **Discovery** (finding problems by observing users pursue loosely defined goals),
- **Benchmark** (testing out how well solutions perform),
- **Remote/on-site**
- **Think-Aloud** (Nørgaard & Hornbæk, 2006, p. 209)
- 5-second
- First-click testing

19

Thinking Aloud: The #1 Usability Tool

Ask *test participants to think out loud while using the application or website — that is, verbalize their thoughts as they move through the user interface.*

- Discover what users really think about your design.
- User misconceptions can be turned into actionable redesign recommendations

20

Think-Aloud Benefits

- Cheap – no special equipment
- Robust – can get reasonably good findings
- Flexible – can use at any stage in the development lifecycle
- Easy to learn

21

To run a basic thinking aloud usability study, you need to do only 3 things:

1. Recruit representative users.
2. Give them representative tasks to perform.
3. Shut up and let the users do the talking.

22

Navigation test

- Identifies steps that cause confusion
- Expect to see high completion rates of > %80 for each step
- When low completion rates, ask the following questions:
  - Did the users click somewhere unexpected?
  - Was there a particular element that misled the user?
  - Was the target element easy to overlook?
  - Are there any unnecessary elements on the page that is distracting users from target?

https://usabilityhub.com/guides/navigation-testing

23

5-second testing

- Measures how well a design quickly communicates a message
- Qualitative and quantitative feedback to help optimize design
- Method:
  - Show an image (application, website or wireframe) to user for just 5 seconds, after which the participant answers question based on their memory and impression
- Why 5 seconds?
  - Studies found that visitors spend only a few seconds assessing your website before deciding to stay or leave

https://usabilityhub.com/guides/five-second-testing

24

Which method?

- How quickly can users find information on my website?
  - First click test
- Can users complete a task in my software interface?
  - Navigation test
- Can users complete a purchase on my e-commerce store?
  - Navigation test

25

The importance of a script

26

You are less biased when you stick to the script

- A key reason we write our test plan is for transparency. We need to make our work **reviewable** so that our **unconscious biases can be recognized**. The more we can script, the better.
- Steve Krug is one of the big names in usability, and he has given us a wonderful sample script to work from. Let's take a look Steve Krug's sample script.

https://a11ycourse.ca/supplemental/test-script.html

27

What is a usability testing script?

A document that serves as a plan for your usability testing so that you, as a facilitator do not forget anything and keep your testing consistent for each participant.

28

Benefits of having a usability testing script:

- **Assures consistency**: you give all testers the same tasks and ask the same questions. It simplifies the analysis process later and helps to avoid confusion.
- **Encourages you to prepare tasks** and questions and think them through beforehand.
- **Stores all the things you need to say in one place**, making it easy to interact with the participants and avoid chaos.
- **Helps you make sure nothing is forgotten** and the test goes according to plan.

29

How to write a good usability testing script?

- Intro
- Define tasks
- Post-study

Image: https://unsplash.com/photos/5QiGvmyJTsc

30

Intro:

- Write a good introduction and **set a comfortable environment for the participant**. This allows participant to be honest and not afraid of expressing their confusion
- Tell them the process of the study
  - How long
  - Get their permission to record (or ask them to ask non-disclosure – if applicable)
  - Ask demographic & profile information
  - Ask them if they have any questions

31

Define the task

- Explain the task scenarios
  - 3-5 task scenarios
- Write down any questions you would like to ask either during or after the task

32

Post-study

- After tasks are completed, ask participants additional feedback
- Some questions to ask:
  - General impression of the test, website features and design, whether the product was useful

33

An activity

Get together in pairs, and read the script to one another.

Answer the following questions:

- What is the purpose of questions like "what's the split between email and browsing [in your weekly internet usage]"?
- Why do we need to assure the user "we're testing the *site*, not you"?
- What type of data is being collected here? Quantitative? Qualitative?
- Find four examples of the types of data that could be collected in this user test.
- What would you change in this script? Why?

34

4. Metrics

35

Metrics

What is the **data** you'll be collecting? How should it be **measured**? How can you be sure it fully **answers your questions**?

This is probably the **most important part of your plan**.

36

Examples of metrics

- **Successful Task Completion**
- **Critical Errors**
- **Non-Critical Errors**
- **Error-Free Rate** Error-free rate is the percentage of test participants who complete the task without any errors (critical or non-critical errors).
- **Time On Task**
- **Subjective Measures** These evaluations are self-reported participant ratings for satisfaction, ease of use, ease of finding information, etc where participants rate the measure on a 5 point Likert scale. A good, focussed example of this is the system usability scale
- **Likes, Dislikes and Recommendations** Participants provide what they liked most about the site, what they liked least about the site, and recommendations for improving the site.

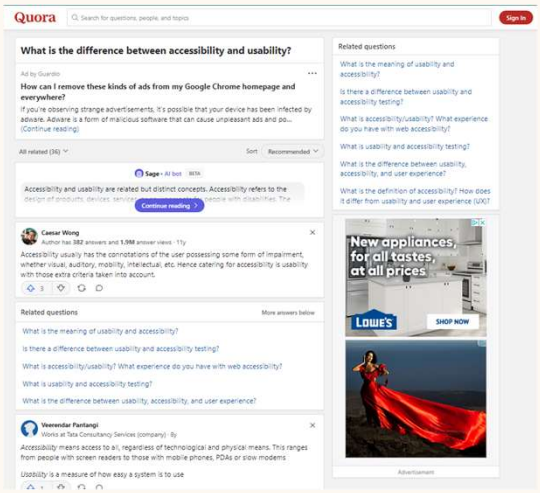
37

Additional sections to include:

1. **Executive summary**: the whole plan, summarized in a couple paragraphs. **This should be the last thing you write.**
2. **Schedule and location**
3. **Equipment**:
4. **Scenario tasks**

38

Usability testing task example #2: Quora



Quora is a Q&A social network

- *"You're just starting out investing in stocks. But you are confused about what stocks are suitable for beginners. Then you decide to ask questions on Quora. Show us how you can ask a question on Quora."*
- *"You are an article writer. To fill your spare time, you would like to share a link that you have written on Quora. How do you share a link in Quora?"*
- *You're reading a post on Quora, and then you come across an answer that is so smart and interesting that you decide to support the answer. Show us how you do it.*

Source:
https://bootcamp.uxdesign.cc/quora-usability-testing-ux-case-study-f3468fd46c1f

39

Quora Outcomes

- **The success rate of the whole test was roughly 71%**, with 45 attempts to perform the tasks and 10 failures.
- **Some of the usability problems they uncovered were:**
- Icons and buttons that are too small and hard to identify
- Confusing terms such as "downvote"
- The "find room" button did not correspond to users' expectations etc.

40

Learning to use the keyboard

- Keyboard controls are key for accessibility - that's why I'm having you test non-classmate users (if that's who you decide to test) with the keyboard.
- Feel free to teach them if they don't know how!
- Also - the only form control that's not captured here are radio buttons - they can be a little unintuitive if you've never used them with a keyboard before.

41

Command	Mac Shortcut	Windows Shortcut
Scroll up/down	⇧ / ⇩	
Previous page	Cmd + ⇦	Alt + ⇦
Next page	Cmd + ⇨	Alt + ⇨
Change focus to next focussable element	Tab	
Change focus to previous focussable element	Shift + Tab	
Click on focussed element (i.e. link, submit)	Return	Enter
Change the value of a form element (i.e. dropdown)	⇧ / ⇩	

42

An activity

1. Go to toronto.ca
2. Get to the section with the breadcrumbs "City of Toronto / Business & Economy / Industry Sector Support / Film / Plan Your Shoot / Letters of Notification"
3. Find the phone number for the Toronto Film, Television & Digital Media Office.

43

Component of the Week

44


```
const toggles =
document querySelectorAll '[role="switch"]'
) // For each element with role=switch
toggles forEach toggle => { // Add a click
event listener
toggle addEventListener "click" e => //
'checked' = whether or not 'aria-checked' is
equal to true let checked =
toggle getAttribute "aria-checked" ===
"true" // set aria-checked to the opposite of
the 'checked' variable
toggle setAttribute "aria-checked"
!checked ) }) }
```

The most complex concept when creating accessible content is **communicating a change of state**.

State changes are most often apparent because of **visual cues**.

In some cases, a toggle switch should just be a radio button (which, as a native form element, is inherently accessible). However, a 'toggle' can sometimes better communicate the idea of submitting a default binary value, rather than choosing one from a group of values.

45

Toggles are **not** inherently accessible, because **they are not native HTML elements** - so we have to **communicate their purpose**, and **not just visually**.

Luckily, **ARIA** recognizes that they are a common feature, so we have a role value available:

```
<button role="switch">
```

46

With this role, we can pass the aria-checked true and false values to indicate whether it's 'on' or 'off'.

We change these values with JavaScript.

We're also able to **tightly couple** our styles with the attributes that make this switch accessible - meaning we **make accessibility mandatory**.

47

One more important thing to note is the aria-labelled by attribute - meaning that anyone landing on our toggle (for example, by tabbing to it) will know what it's for, without having to backtrack.

48

Lab #2: Due Wednesday

Write a script for your usability testing by referencing the structure of Steve Krug's script.

https://a11ycourse.ca/supplemental/test-script.html

49