Chapter 9

Usability Testing Methods

> *" Would you fly in an airplane that hasn't been flight tested? Of course not. So you shouldn't be using software that hasn't been usability tested. "*
>
> [Ben Shneiderman, The Front Desk, BBC Video, 1995.]

Empirical testing of interface design with *representative users*.

Seven Testing Methods

Usability testing methods include:

- *Thinking Aloud Tests*: Test users verbalise thoughts while performing test tasks.
- *Co-Discovery Tests*: Two test users explore an interface together. Insight is gained from their conversation while performing test tasks.
- *Formal Experiments*: Controlled experiment, face-to-face with test users, measurements and statistical analysis.
- *A/B Testing*: Controlled experiment on (part of) actual user population, typically (remote) web site users, with measurements and statistical analysis.
- *Post-Test Interviews*: Qualitative feedback after a test.
- *Post-Test Questionnaires*: Quantitative feedback after a test.
- *Usage Studies*: Usage data is collected from a small number of users working on their own tasks in their natural environment over a longer period.

References

++ Steve Krug; *Rocket Surgery Made Easy*; New Riders, 2009. ISBN 0321657292 (com, uk) [Krug 2009]

\+ Carol Barnum; *Usability Testing Essentials*; 2nd Edition, Morgan Kaufmann, 2020. ISBN 0128169427 (com, uk) [C. M. Barnum 2020]

\+ Jeffrey Rubin and Dana Chisnell; *Handbook of Usability Testing*; 2nd Edition, Wiley, 2008. ISBN 0470185481 (com, uk) [Rubin and Chisnell 2008]

+ Joseph Dumas and Janice Redish; *A Practical Guide to Usability Testing*, Revised Edition; Intellect, 1999. ISBN 1841500208 (com, uk) [Dumas and J. Redish 1999]

° Thomas Landauer; *Research Methods in Human-Computer Interaction*; In *Handbook of HCI*, Helander (Ed.), North-Holland, 1988. ISBN 0444705368 (com, uk) [Landauer 1988]

° Ericsson and Simon; *Protocol Analysis: Verbal Reports As Data*; MIT Press, May 1993. ISBN 0262550237 (com, uk) [Ericsson and Simon 1993]

° van Someren, Barnard, and Sandberg; *The Think Aloud Method*; Academic Press, 1994. ISBN 0127142703 (com, uk) [van Someren et al. 1994]

++ Andy Field et al; *Discovering Statistics Using R*; Sage Publications, Mar 2012. ISBN 1446200469 (com, uk) [Field et al. 2012]

++ Andy Field; *Discovering Statistics Using SPSS*; Sage Publications, 3rd Edition, 2009. ISBN 1847879071 (com, uk) [Field 2009]

+ Andy Field and Graham Hole; *How to Design and Report Experiments*; Sage Publications, 2002. ISBN 0761973834 (com, uk) [Field and Hole 2003]

° Shaughnessy et al; *Research Methods In Psychology*; 6th Edition, McGraw Hill, April 2003. ISBN 0071198903 (com, uk) [Shaughnessy et al. 2003]

Online Resources

K. Anders Ericsson; *Protocol Analysis and Verbal Reports on Thinking*; A brief summary and description of protocol analysis. `http://psy.fsu.edu/faculty/ericssonk/ericsson.proto.thnk.html` [Ericsson 2002]

Experience Changes Perception

- Experience changes one's perception of the world.
- It is almost impossible to "forget" an experience and put oneself in the position of someone not having had the same experience [Landauer 1988, pages 907–911].
- For example:
  - Karl Dallenbach's famous photo [Dallenbach 1951; Landauer 1988, page 909].
  - Hearing a clear version of a garbled voice recording [NatGeo 2011, 00:31:32-00:32:32]. [Video: `https://dailymotion.com/video/xq1rfl?start=1883`]

Why do Usability Testing?

- More often than not, intuitions are wrong!
- People often believe they understand the behaviour of others based on their own experiences.
- Designers often believe they understand the behaviour of their users.
- Observation and measurement (→ usability tests) can dispel such false beliefs.

Keyboard vs Mouse

From studies at Apple [Tognazzini 1992]:

- Test users consistently report that keyboarding is faster than mousing.
- The stopwatch consistently proves that mousing is faster than keyboarding, an average of 50% faster.

"In one study of this phenomenon (Tognazzini, Tog on Interface, 1992.), users were asked to do the same task using the keyboard and the mouse. The keyboard was powerfully engaging, in the manner of many videogames, requiring the user to make many small decisions. The mouse version of the task was far less engaging, requiring no decisions and only low-level cognitive engagement.

Each and every user was able to perform the task using the mouse significantly faster, an average of 50% faster.

Interestingly, each and every user reported that they did the task much faster using the keyboard, exactly contrary to the objective evidence of the stopwatch.

The most obvious take-away message from this is that people's subjective beliefs as to what is or is not quick are highly-suspect. No matter how heart-felt the belief, until a stopwatch shows it is true, do not accept personal opinion about speed and efficiency as fact. Instead, user-test." [Tognazzini 1999]

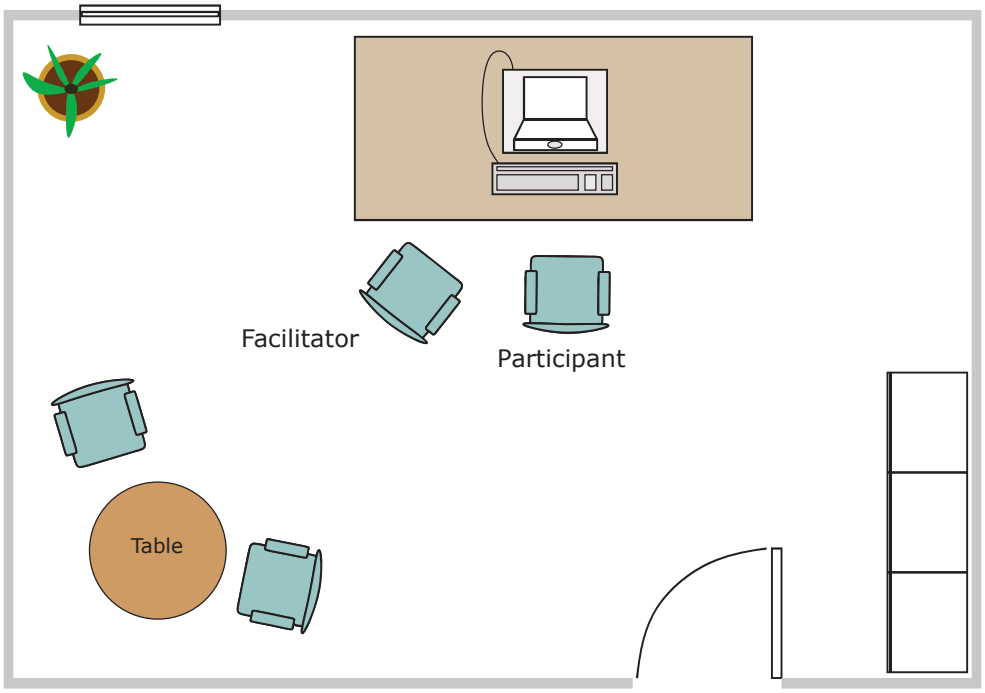

Figure 9.1: A simple usability test setup. This and the following test setup diagrams were inspired by those of Rubin [1994].

9.1 Preparing for Usability Testing

Test Environment

Ensure comfortable test environment:

- Organise *quiet* room.
- Put up sign "Usability Test in Progress – Do not Disturb".
- Disable telephones (fixed line and mobile).
- Ensure good lighting [K. Wilson 2017].
- Provide (non-alcoholic) refreshments.

Or use dedicated usability lab . . .

Test Equipment

- Digital video camera.
- Video tripod.
- Good microphone (table, lapel, or headset). A singer's microphone is no good, because it must be held right next to the mouth.
- Headphones (to monitor sound).

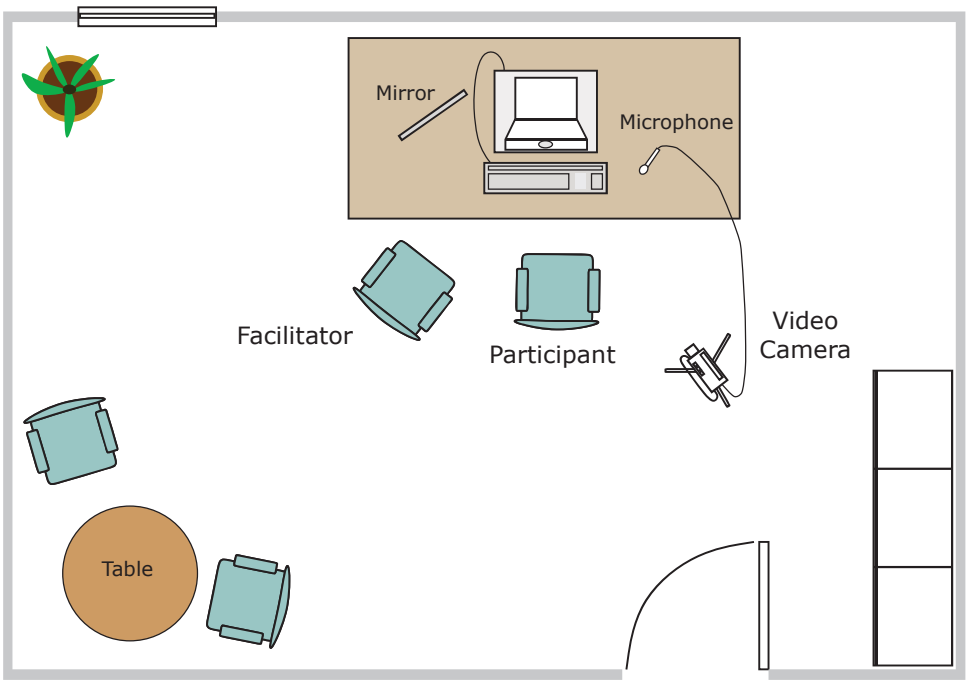

Figure 9.2: A typical single room, single camera usability test setup.

Figure 9.3: Simple usability test setup in 2002 with a single video camera. A mirror is used to capture the user's facial expressions. Video is recorded onto a VHS cassette.

Figure 9.4: Simple usability test setup in 2019 with a single video camera. Extrenal camera, mirror, and table microphone, in addition to screen video capture and webcam capture on the laptop. [Image used with kind permission of Angelika Droisner and Ana Korotaj.]

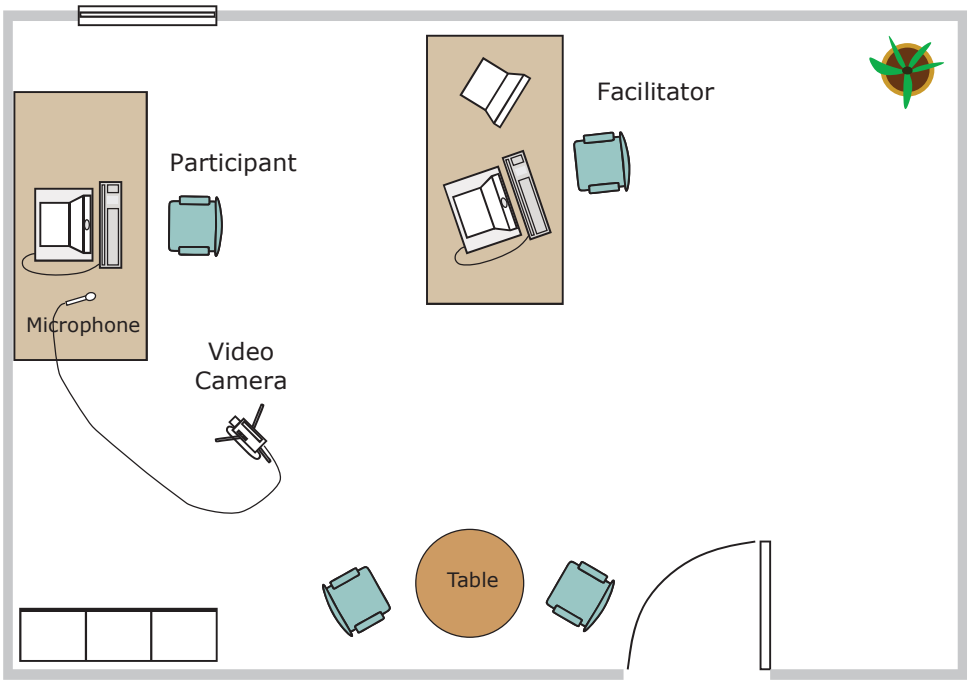

Figure 9.5: Single room test setup. Facilitator sits behind test participant monitoring video output and using logging software.

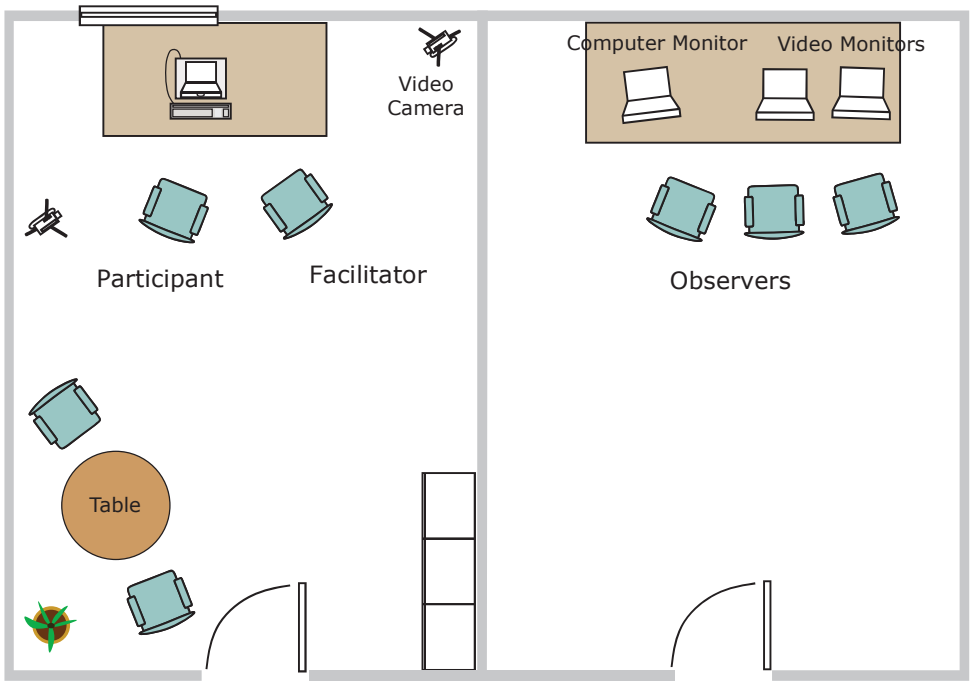

Figure 9.6: Observation room with electronic monitoring.

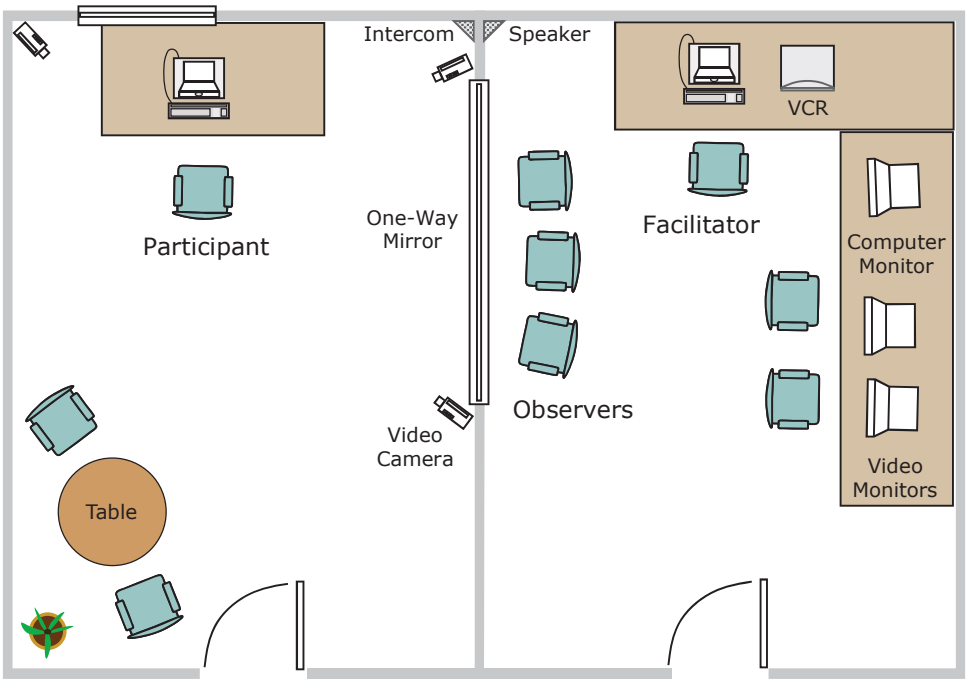

Figure 9.7: A classical usability lab, including an observation room with a one-way mirror.

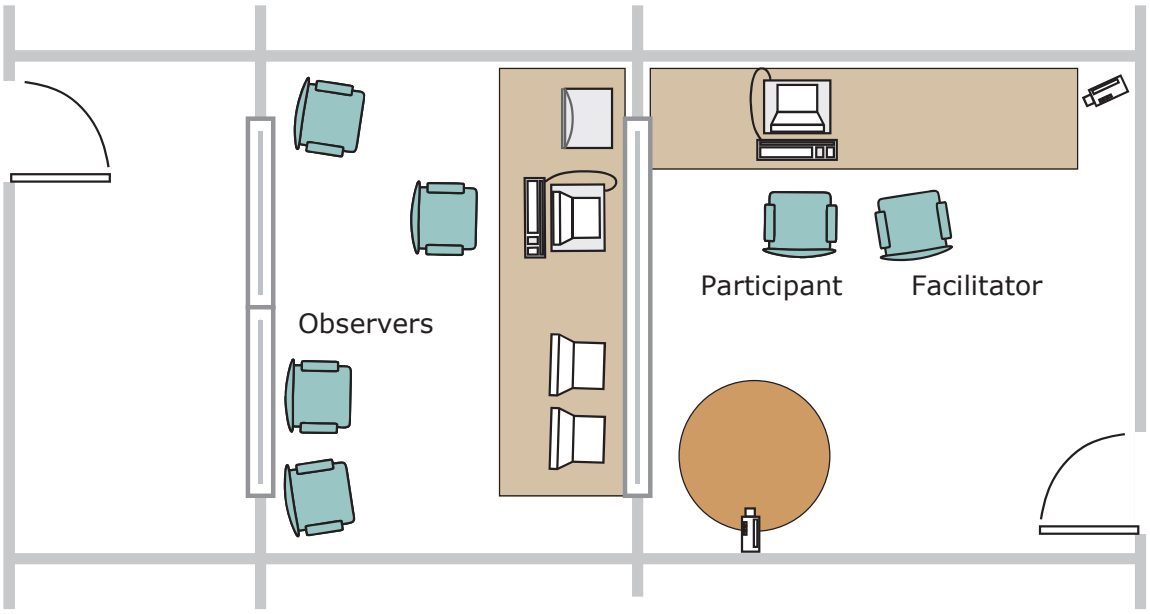

Figure 9.8: The standard usability lab at Microsoft headquarters in Seattle. There are 25 such labs on two floors, side to side. Users enter from the right, developers enter from the left. [Adapted from diagram at [Microsoft 2005].]

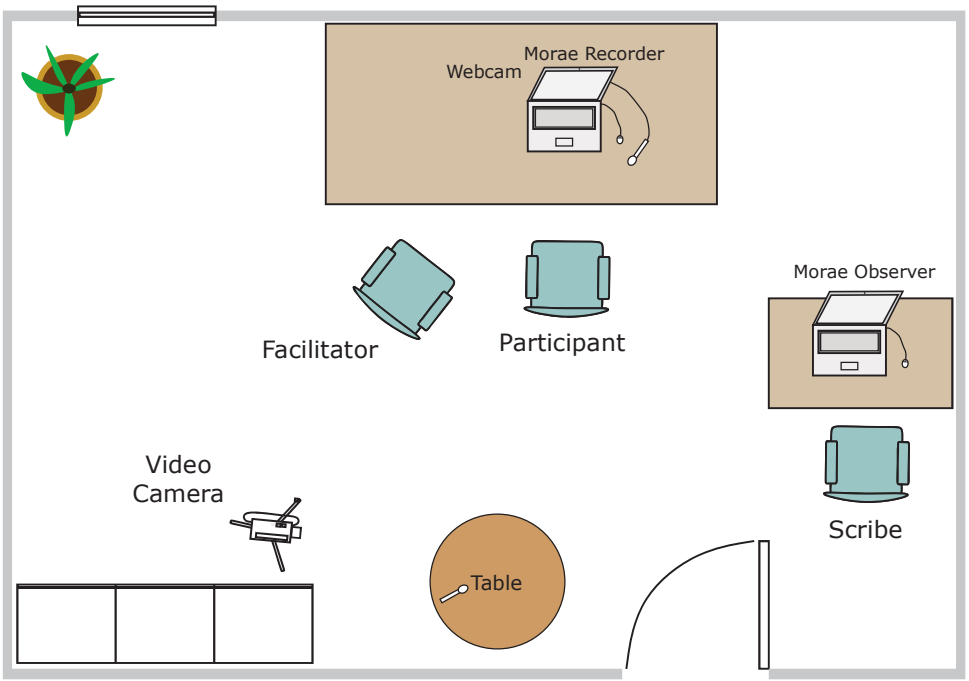

Figure 9.9: Morae software [TechSmith 2018] installed on the test laptop captures screen video, webcam video, and audio in snychronised streams.

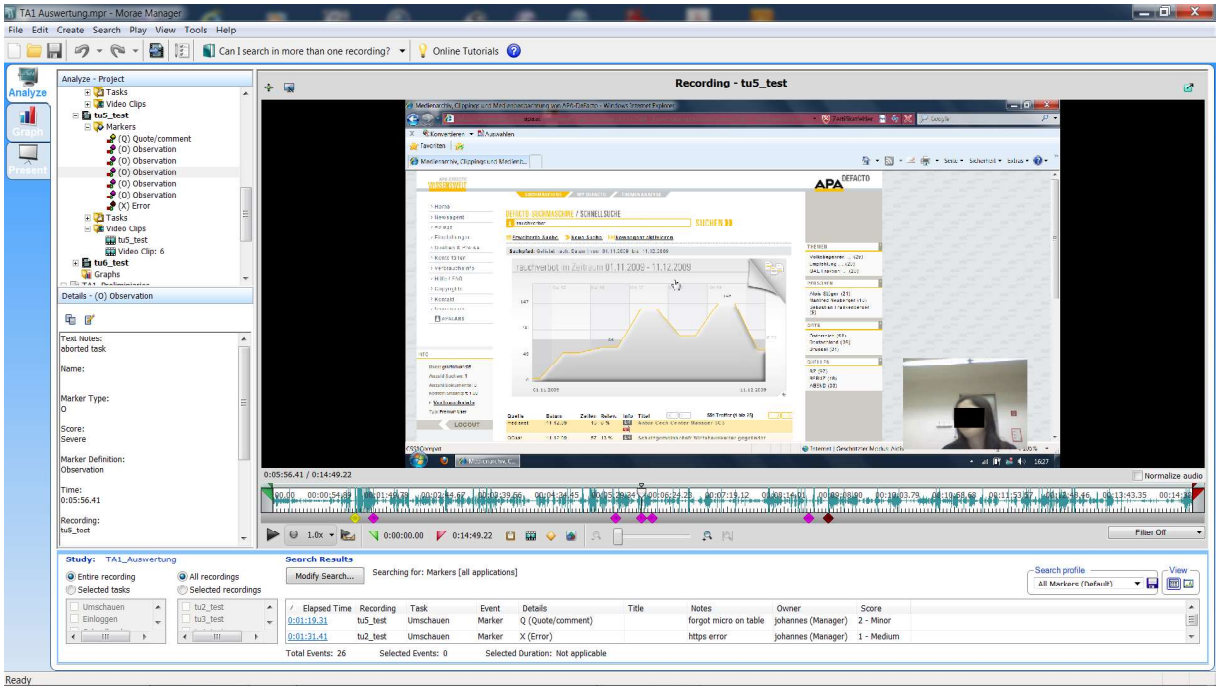

Figure 9.10: Morae captures screen video, webcam video, and audio in in snychronised streams. In Morae Manager, noteworthy events can be marked, and clips of findings easily assembled.

- Mirror (to capture user's facial expressions).
- Light (desk lamp or video lighting).
- Colour video monitor (to monitor camera image).
- Powerstrip, extension cables.
- Transportation cart, or rucksack.
- "Do not Disturb" sign.
- Refreshments.
- Logging software or forms.

Figures 9.11 and 9.12 show the portable usability kits used at Graz University of Technology.

Roles in the Test Team

- *Facilitator* (Administrator, Moderator, Manager, Monitor)
  In overall charge of test, responsible for *all* interaction with test user (introduction, test, debriefing).
- *Data Logger* (Scribe)
  Records activities and events of interest on paper, incl. time of occurrence. [Assign shorthand codes to expected activities *before* test.]
- *Video Operator*
  Responsible for recording *entire* test proceedings, incl. initial instructions and debriefing interview:

Usability Kit Inventory List

1. Tripod Hama Profil 74
2. Rucksack LowePro
3. Headphones
4. Headphones extension cable
5. Microphone Philips SBC ME570
6. Microphone extension cable
7. Headphone adapter
8. Usability kit inventory list
9. Video camera manual
10. Video camera power supply
11. Video camera mains cable
12. Microphone adapter cable
13. Video camera Sanyo Xacti HD1010
14. Video camera bag
15. Video camera remote control
16. Transcend SD HC Card 16gb

Figure 9.11: The portable usability kit used at Graz University of Technology. The inventory shows all of the components of the kit.

Figure 9.12: The portable usability kit used at Graz University of Technology. On the left the kit has been set up ready for use. The righthand photos illustrate packing the kit.

  - Check camera angles so user and interface both clearly visible.
  - Use *manual* focus (can't autofocus on computer screen).
  - Turn off any data fields (date, time, etc.) superimposed onto the video stream.
  - Ensure audio recording level is high enough.
  - Label, copy, and edit recordings.
- *Computer Operator*
  - Play the role of computer in a paper prototype.
  - Reset the interface to a clean state for each new test user (clear the cache, history, cookies, etc.).
  - Do *not* set the Home button to a web site under test, ask the user to type the URL.
  - Restart after system crash, unexpected hang-ups, etc.

Test Users

The users taking part in the test:

- Refer to them as *test participants* or *test users*.
- Never ever call them *test subjects* or *probands*. And certainly not so they can hear!

9.2 Six Stages of Conducting a Test

1. Develop the Test Plan
2. Select and Recruit Participants
3. Prepare Test Materials
4. Run a Pilot Test
5. Conduct the Real Test
6. Analysis and Final Report

   Note: *Always* do a pilot test!

9.2.1 The Test Plan

Main section headings:

- Purpose
- Problem Statement
- User Profile
- Method (Test Design)
- Task List

Task	*Description*	*Prerequisites*	*Completion Criteria*	*Max. Time*	*Possible Solution Path*
1	Please go to the website `tugraz.at` and spend a few minutes looking around the site.	Browser is open at `google.com`.	The user indicates that they have finished looking around.	3 minutes.	
2	When is the general admission period for next winter semester?	Browser is open somewhere on `tugraz.at`.	The user identifies the general admission period.	4 minutes.	Main Menu → Studying and Teaching: Academic Calendar → Academic Year.
3	Find and look through the curriculum of your degree programme.	Browser is open somewhere on `tugraz.at`.	The user has opened the PDF file of the curriculum.	4 minutes.	Main Menu → Studying and Teaching → Select a degree programme → Curriculum.
4	etc.				

Figure 9.13: An example internal task list for a usability test of of a university website.

- Test Environment
- Data to be Collected
- Content of Report

Task List

- Prioritise tasks by frequency and criticality.
- Choose those most frequent and critical to test.
- Make a task list for test team internal use only.
- For each task:
  - Define any prerequisites.
  - Define successful completion criteria.
  - Specify maximum time to complete each task, after which help may be given.
  - Define what constitutes an error.
- Do *not* instruct the test user to return to the initial screen (home page) at the beginning of each task. If they do so of their own accord, that fine.

  See Figure 9.13.

9.2.2 Selecting and Recruiting Participants

- Users are typically divided into different user groups, based on their characteristics and needs.
- Test each user group *separately*.
- Test at least 5 test users per user group.
- Choose *representative* test users who span the chosen user group.
- Recruit test users via employment agency, students, existing customers, internal personnel.
- Maintain a database of potential test users.
- Screening questionnaire (ensure users fit profile).

9.2.3 Test Materials

- Orientation Script
- Background Questionnaire
- Nondisclosure and Consent Form
- Training Script (if any)
- Task Scenarios
- Data Collection Forms
- Debriefing Interview Guide
- Feedback Questionnaire
- Checklist

Orientation Script

- Introduce yourself and any observers by first name (no titles or job descriptions!).
- Explain that the purpose of the test is to collect input to help produce a better interface.
- Emphasise that system is being tested *not* user.
- Acknowledge software is new and may have problems.
- Do not mention any association you have with product (do mention if you are *not* associated with product).
- Explain any recording (reassure confidentiality).
- Say user may stop at any time.
- Say user may ask questions at any time, but they may not be answered until after the test is completed.
- Invite questions.

See Figure 9.14.

"Hi, my name is Keith. I'll be working with you in today's session. [Frank and Thomas here will be assisting me].

We're here to test a new product, the Harmony 3D Information Landscape, and we'd like your help.

I will ask you to perform some typical tasks with the system. Do your best, but don't be overly concerned with results – the system is being tested, and not your performance.

Since the system is a prototype, there are certainly numerous rough edges and bugs and things may not work exactly as you expect.

[I am an independent researcher hired to conduct this study, and have no affiliation with the system whatsoever]. My only role here today is to discover the flaws and advantages of this new system from your perspective. Don't act or say things based on what you think I might want to see or hear, I need to know what you really think.

Please do ask questions at any time, but I may only answer them at the end of the session.

While you are working, I will be taking some notes. We will also be videotaping the session for the benefit of people who couldn't be here today.

If you feel uncomfortable, you may stop the test at any time.

Do you have any questions?

If not, then let's begin by filling out a short background questionnaire and having you sign the nondisclosure agreement and consent to tape form."

Figure 9.14: An orientation script for testing the Harmony 3D Information Landscape.

Background Questionnaire (Pre-Test Questionnaire)

For better understanding of the user's background:

- Admin. data: date, test number, user number or id.
- General data: age (range), gender, occupation, educational level, etc.
- Computer experience: total time, frequency of use, types of software, have used a GUI before, . . .
- Application experience: total time, frequency of use, brand.

See Figure 9.15.

The facilitator should ask the test user the questions on the background questionnaire and fill it in (it is more efficient than handing the pen and questionnaire to the user).

Training Script

Exact written description of prior training:

- Demonstration of GUI.
- Demonstration of special interaction styles: mouse keys, drag-and-drop, etc.
- Walk-through of sample task.
- Demo of how to think aloud (for Thinking Aloud style tests).

Task Scenarios

The task descriptions given to the test users:

- Simple introductory first task (early success).

Background Questionnaire

Date: _____ Test No.: _____ User No.: _____

1. General Information

Gender: ☐ Male ☐ Female Age: _____ Occupation: _____

2. Sight Impairment

1. Do you use a sight aid when working on a computer?
   ☐ None ☐ Glasses ☐ Contact Lenses ☐ Other _____
2. Do you have any form of colour vision deficiency?
   ☐ No ☐ Yes, _____

3. Education

1. Highest educational grade you have attained:
   ☐ Secondary School ☐ University Degree ☐ Doctorate
2. If you are a student or graduate, please state your main area of study.
   _____

4. Use of Computers

1. How long have you been using personal computers (years)? _____
2. In a typical week, how many hours do you use a computer? _____
3. Which kind of personal computer do you use most?
   ☐ MS Windows ☐ Apple Mac ☐ Unix ☐ Other _____

5. Web Experience

1. How many hours per week do you use the World Wide Web? _____
2. Which kind of device do you most often use to access the World Wide Web?
   ☐ Desktop ☐ Laptop ☐ Tablet ☐ Smartphone
3. Which kind of internet connection do you normally use?
   ☐ xDSL ☐ Cable ☐ Fibre Optic ☐ 3G Mobile
   ☐ LTE ☐ Other _____
4. Which web browser do you normally use?
   ☐ Chrome ☐ Firefox ☐ Safari ☐ Microsoft IE/Edge
   ☐ Opera ☐ Other _____

Figure 9.15: A typical background questionnaire for a web site test.

Thank you for participating in our product research. Please be aware that confidential information will be disclosed to you and that it is imperative that you do not reveal information that you may learn during the course of your participation. In addition, your session will be videotaped, to allow staff members who are not present to observe your session and benefit from your feedback.

Please read the statements below and sign where indicated. Thank you.

I agree that I will disclose no information about the product research conducted by ABC Company Inc. or about the specifications, drawings, models, or operations of any machine, devices, or systems encountered.

I understand that video and audio recordings will be made of my session. I grant ABC Company Inc. permission to use these recordings for the purposes mentioned above, and waive my right to review or inspect the recordings prior to their dissemination and distribution.

Please print name: _____

Signature: _____

Date: _____

Figure 9.16: A combined nondisclosure and consent form.

- Realistic scenarios in typical order.
- If sequential ordering not crucial, randomise presentation order (→ counterbalances learning effect).
- Each task scenario on a separate sheet.
  [Do *not* hand the user all the tasks at once, but one at a time!]
- Do not guide participants through the task!
  [Describe the goal rather than individual steps.]

An Example Task Scenario

Task 2.	Find the number of the telephone hotline.

Data Collection Forms

- Define abbreviations for expected events → coding scheme. See Table 9.1.
- Use a probe mark like ⓘ to signal an event worth probing during debriefing interview.
- Paper or electronic data collection forms (or instrumented software). See Figure 9.17.

Debriefing Interview Guide

- How was it?
- Structured interview questions (things to discuss with user in any case).

Feedback Questionnaire (Post-Test Questionnaire)

Collect ratings, opinions, suggestions (hard to observe in other ways), for example:

Code	*Event*
S	Start of task.
E	End of task.
N	Negative observation (problem).
P	Positive observation.
Q	Quote or comment from user.
X	Error or unexpected action.
F	Facilitator prompts user.
H	Facilitator helps user.
T	Timeout, exceeded maximum time.
ⓘ	Probe during interview (probe mark).
C	Comment by facilitator.
*	Very important action.

Table 9.1: A simple coding scheme for logging events during a test.

Test:		User No.:
Date:	Time:	Page: of

Task	*Elapsed Time*	*Observations*

Figure 9.17: A generic data collection form.

Test: Edit HTML Document		User No.: 3
Date: 23 Apr 97	Time: 11:50	Page: 1 of 3

Task	*Elapsed Time*	*Observations*
1	04:25	X Opened wrong file. Found mistake. X Opened wrong file again. Self-corrected due to error message.
	06:00	P
	07:00	T
2	11:30	Q “I wish it were always that easy!”
	15:20	ⓘ Very long hesitation, then correct action.
	16:15	E

Figure 9.18: A completed data collection form with a probe mark during task 2.

1.	Getting to the right part of the site.	Very easy	3 2 1 0 1 2 3	Very hard
2.	Quality of information.	Very good	3 2 1 0 1 2 3	Very poor
3.	It is easy to read the text.	Very easy	3 2 1 0 1 2 3	Very hard
4.	The site's local search facility (if available and used).	Very good	3 2 1 0 1 2 3	Very poor
5.	Appearance of site, including colours and graphics.	Very good	3 2 1 0 1 2 3	Very poor
6.	Consistency of site.	Very consistent	3 2 1 0 1 2 3	Very inconsistent
7.	Speed of pages displaying.	Very fast	3 2 1 0 1 2 3	Very slow

Figure 9.19: An example feedback questionnaire using 7-point semantic differentials for a usability test of of a university website.

- Interface organisation matches real-world tasks?
- Too much or too little information on screens?
- Similar information consistently placed?
- Problems with navigation?
- Computer jargon?
- Appropriate use of colour?

See Figure 9.19.

Checklist

- Make chronological checklist of things to do, as shown in Figure 9.20.

9.2.4 Pilot Test

Always perform pilot tests of the *entire* test procedure.

You always find something you need to fix, such as:

- ambiguous instructions.
- unrealistic time estimates.
- ambiguous task completion criteria.
- misleading questionnaire questions.
- dead battery in microphone.

If you do not catch these things in a pilot test and one of these problems occurs with user number 1 of 10 scheduled at hourly intervals, it can ruin the whole test.

Maybe even run two pilot tests: once with colleagues, once with one or two real test users.

Test Checklist

1. Preparation:
   - ☐ Reset interface for new user.
   - ☐ Check that everything is ready in test room.
2. Opening:
   - ☐ Greet the participant.
   - ☐ Go through orientation script and set the stage.
   - ☐ Ask questions on background questionnaire: facilitator asks and fills out form.
   - ☐ Ask user to read and sign consent and non-disclosure forms.
3. Test Session:
   - ☐ Move over to testing area (computer).
   - ☐ Start computerised session recording (Morae).
   - ☐ Provide any prior training.
   - ☐ Provide training of thinking aloud.
   - ☐ User begins with tasks.
   - ☐ User finishes last task.
4. Closing:
   - ☐ Interview: how was it?
   - ☐ Structured interview questions.
   - ☐ Individual interview questions arising from test.
   - ☐ Feedback questionnaire. User fills out form.
   - ☐ Thank participant, provide any remuneration, show participant out.
5. Clean-Up:
   - ☐ Summarise thoughts about this test.
   - ☐ Organise data sheets and notes.
   - ☐ Check and archive session recordings.

Figure 9.20: A example test checklist.

9.2.5 Conducting the Real Test

- Test facilitator handles *all* interaction with participant (other team members and observers remain completely quiet).
- Use screen capture software to record the entire test session, including the user's voice and (possibly) face. Later on, extract video clips to illustrate findings.
- Do not prompt or bias user during test (beware of non-verbal signals).
- Only assist if user in severe difficulty (make note of when and what help given).
- Conduct debriefing interview and/or feedback questionnaire.
- Test users who are waiting should wait *outside* the testing room. They should *not* observe a current test in progress!

Debriefing Interview

1. Let user speak thoughts first: "So, how was it?".
2. Let them talk and talk, until they stop talking of their own accord.
3. Top-down: probe high-level issues from topic guide first, then more detailed questions about each task.
4. Probe specific issues arising from test notes ⓘ. See Figure 9.18.
5. Ask any questions passed to the facilitator from the observers (they should be written onto index cards by the observers).

See also Section 9.7.

9.2.6 Analysis and Final Report

- Compile and summarise data, for example:
  - Mean, median, range, and standard deviation of completion times.
  - Percentage of users performing successfully.
  - Bar chart of preference scores.
  - etc.
- Analyse data:
  - Identify errors and difficulties which arose.
  - Diagnose the source of each error.
  - Prioritise problems by their severity or criticality.

Final Report

- Title Page

- Description of Test Environment
  - Hardware, software version, test room, dates when tests were performed.
- Executive Summary
  - Concise summary of major findings, no more than a few pages.
- Description of Test
  - Updated test plan, method, training, and tasks.
- Test Person Data
  - Tabular summary of age, occupation, experience.
- Results
  - Tabular and graphical summaries of times taken, number of errors made, questionnaire responses, etc.
  - Discussion and analysis, amusing quotations.
- List of Positive Findings
- List of Recommendations List of problems discovered, in descending order of severity, and recommended improvements. For each recommendation:
  - diagnose why the problem occurred
  - illustrate it with a screen shot
  - rate its severity (0. . . 4 scale)
  - indicate exactly how many test users experienced the problem
  - include a reference to timestamp(s) on the video tape
  - possibly include an appropriate user quotation
  - describe your suggested improvements
- Appendices (raw data and tables).
  - Background questionnaires, consent forms, orientation script, data collection forms, video and audio tapes, transcripts, etc.

Example Recommendation

R12. **Sort Order Panel** (Severity 3.2)

- *Problem*: Users had problems understanding the sort order panel. In particular, the plus and minus icons used for increasing and decreasing order are non-intuitive.
- *Reference*: TP1, 00:08:15

  "What does this plus mean?"
- *Recommendation*: redesign the icons, for example as sloping ramps.

9.3 Thinking Aloud Tests

A thinking aloud test is a formative usability testing method:

- Test users are asked to verbalise their thoughts ("think aloud") while performing tasks.
- In other words, to provide a running commentary on what they are seeing, thinking, and doing.
- Provides insight into their thought processes and why things go wrong (process data).
- Relatively small number of test users (say 3 to 5).
- Many vivid and colourful quotes.

Detecting Vocabulary Problems with Thinking Aloud

Example from Lewis and Rieman [1993, Section 5.5]:

- Menu-based administrative system for law offices.
- System messages extensively refered to "parameter".
- Test users persistently misread "parameter" as "perimeter".
- Hard to detect such problems just by watching people's mistakes, much easier when they are thinking aloud.

The Thinking Aloud Method

Ask users to tell you:

- what they are trying to do
- things they read
- questions that arise in their mind
- things they find confusing
- decisions they make

Preparing the User

- Demonstrate thinking aloud for an unrelated task, e.g. looking up the films on tonight in the local cinemas (newspaper or online).
- Show user short video clip of a previous thinking aloud test.
- Have the user practice the technique using a different interface and unrelated task.
- Request questions be asked as they arise, but explain that you won't answer them until after the test.

Test Facilitator's Role

The facilitator must encourage the user to keep up a flowing commentary:

- Minimal encouragers:
  - non-committal “uh huh”, “mmm”, “yeah”.
  - head-nodding.
  - body language.
- Neutral, unbiased prompts:
  - “Tell me more.”
  - “Keep talking to me.”
  - “Please tell me what you are doing now?”
  - “I can’t hear what you are saying”
  - “What are you thinking right now?”

Video: Demo Usability Test

- Steve Krug’s book Rocket Surgery Made Easy [Krug 2009] has an accompanying demo video [Krug 2010a]. [Video: `https://youtu.be/1UCDUOB_aS8`]
- Watch the video and make a note of any usability problems you observe.
- Pause the video at 21:05.
- At this point, decide which were the top three usability problems you observed.
- Now watch the rest of the video.
- The video shows an example of a simple thinking aloud test.
- Facilitator (Steve Krug) sitting next to test user (his wife).
- Just screen and audio recording (no usercam).

  [I generally prefer to have a usercam where possible, so the user’s reactions and facial expressions are captured. Of course, such videos should never be published without the user’s explicit consent.]

Spontaneous Comments are Best

Do *not* direct the user with specific questions like:

- “Why did you do that?”
- “Why didn’t you click here?”
- “What are you trying to decide between?”

Spontaneous comments from the user are best.

Do Not Ask Why Questions

Specific “why” questions encourage plausible, but often unreliable, answers Higgins [2007], as shown in many classic studies:

- Maier [1931]. Problem: tie together two cords hanging from ceiling, too far apart to be grabbed. Solution: tie weight to one cord, set it swinging, grab other cord, wait for swinging cord to come within reach. When Maier "accidentally" brushed against one cord, 16 participants solved the problem. However, when asked how solution was arrived at, only one mentioned that seeing the cord swaying had prompted him. The others gave explanations such as: "It just dawned on me."
- Nisbett and T. D. Wilson [1977, pages 243–244] and T. D. Wilson [2002, pages 102–103]. Four *identical* pairs of nylon stockings were placed from left to right (labeled A to D) on a display table outside a busy supermarket. 52 participants were asked to say which of the stockings were the best quality. The preferences were 12% for A, 17% for B, 31% for C, and 40% for D, indicating a statistically significant position effect. When asked why they chose a particular item, participants made up plausible (but wrong) reasons, such as "superior knit, sheerness, or elasticity". In fact, there is a natural bias towards last of a number of closely matched alternatives.
- Johansson et al. [2005], Johansson and Hall [2008] and Simons [2010, 14:04-15:38] [Video: `https://youtu.be/eb4TM19DYDY?t=14m04s`]. 118 participants were shown 15 pairs of photos of women and asked which woman in each pair was most attractive. The chosen photo was pushed toward the participant for further comment and the non-chosen photo removed. The participant was then asked why they had chosen that photo. Three of the pairs were manipulated (through sleight of hand), such that the non-chosen photo was presented for comment. Most participants failed to notice any manipulation and made up plausible (but obviously wrong) reasons to justify their choice.

  Asking people to explain the reasons for their behaviour (introspection or self-reporting) is unreliable!

Listening Labs

A listening lab is a variant of a thinking aloud test, but without preset tasks:

- Users explore the interface and set their own tasks.
- Single user, thinking aloud.
- Environment simulates real-use setting.
- Pre-defined tasks often neglect what individual users want to accomplish and sometimes miss larger strategic findings.

See [Hurst and Terry 2000].

Pros and Cons of Thinking Aloud

++ finds many usability problems

++ finds *why* problems occur (process data)

\+ small number of test users (3 to 5)

\+ usable early in development process

\+ requires little facilitator expertise

\+ generates colourful quotes

\- - thinking aloud slows users down by about 17% [Ericsson and Simon 1993, page 105]

- depending on the instructions given to the user, having to think aloud can change the user's problem-solving behaviour (they might think more before acting).
- cannot provide performance data (bottom-line data)

9.4 Co-Discovery Tests

A co-discovery test is a variant of a thinking aloud test with two users instead of one.

- Two test users explore an interface *together*.
- There is natural interaction and communication between the test users.
- Marissa Mayer used a co-discovery test for the first ever user test of Google (search engine) [Mayer 2009, 01:00-04:14]. [Video: `https://aiga.org/video-makethink-2009-mayer`]

Pros and Cons of Co-Discovery

++ No unnatural thinking aloud.

- Need twice as many test users.
-- Validity issue: would the interface be used by two people working together in real life?

9.5 Formal Experiments

A formal experiment is a summative usability testing method:

- Controlled experiment with representative test users.
- Measurement and collection of quantitative data.
- Both objective measures (performance, success, . . .) and subjective measures (ratings, overall preference).
- *Never* ask users to think aloud during a formal experiment (thinking aloud slows users down and can change their behaviour).
- Summative evaluation: performed on (fully) implemented design(s).
- Followed by more or less rigorous statistical analysis.
- Two main uses:
  - assessing the absolute performance of an interface.
  - objectively comparing two (or more) alternative interface designs.

Formal experiments provide bottom-line data (measurements), but require larger numbers of test users for statistical accuracy (sometimes around 16 to 20, but often 50 or 100 or more).

Objective Measures

Collect objective, quantitative data by measuring or counting things such as:

- Time to complete specific task(s).
- Number of tasks completed within given time.
- Number of errors.
- Number of deviations (extra clicks) from optimal path.
- Accuracy (answer to question true or false).
- Ratio successful interactions : errors.
- Time spent recovering from errors.
- Number of commands/features used.
- Number of features user can remember after test.
- How often help system used.
- Time spent using help.
- Ratio positive : negative user comments.
- Number of times user sidetracked from real task.

Subjective Measures

Collect subjective, quantitative data, by asking users to give ratings and/or to express a preference (usually via a questionnaire), such as:

- The interface was easy to use (for example, on a scale of 0..6).
- The interface was cluttered (for example, on a scale of 0..6).
- Of the four interfaces, which was your favourite?

Validity

Validity: is measured data relevant to the usability of the real system in real world conditions?

Typical validity problems include:

- *Testing with the wrong kind of user*: For example, testing business students instead of managers for a management information system. However, testing business students will generally lead to better results than testing, say, mathematics students.
- *Testing the wrong tasks*: The results from testing a toy task in a prototype of a few dozen hypermedia documents may not be relevant to a planned system for managing tens of thousands of documents.
- *Not including time constraints and social influences*: Queues of people waiting in line, noise levels in the working environment, etc.

9.5.1 Testing Absolute Performance of One Interface

- One interface.

- Run an experiment to objectively determine whether the interface satisfies specific requirements.
- For example: measure how long it takes 20 expert users to perform task X.
- Result: an expert user can on average perform task X in 2 minutes 10 seconds ± 6 seconds.

9.5.2 Comparing Two Alternative Interfaces

- Two interfaces, A and B.
- Run an experiment to objectively determine which interface is better, according to some criterion (efficiency, error rate, etc.).
- Two different ways of designing an experiment: *independent measures* (also called *between-groups* or *unrelated*) and *repeated measures* (also called *within-groups* or *related*).

Independent Measures (or Between-Groups) Experimental Design

- Two equally-sized groups of test users.
- *Randomly* assign users to two groups.
- Group 1 uses only system A, group 2 only system B.
- Identical tasks for both groups.

Pros and Cons of Independent Measures Experimental Design

+ no problems with learning effect.

- cannot ask users which they preferred.
- generally needs twice as many users.
- large individual variation in user skills (std. dev. $\approx 50\%$).

Repeated Measures (or Within-Groups) Experimental Design

- One group of test users.
- *Randomly* assign users to two equally-sized pools.
- Users work with both systems.
- Pool 1 uses system A first, pool 2 system B first.
- Two sets of different but equivalent (equally difficult) tasks.

Pros and Cons of Repeated Measures Experimental Design

+ automatically controls for individual variability.

+ can ask users which they preferred.

+ generally needs fewer test users in total.

- transfer of skill between systems (learning effect).

Example Experimental Designs

Independent Measures	
System A	*System B*
John	Dave
James	Mariel
Mary	Ann
Stuart	Phil
Keith	Tony
Gary	Gordon
Jeff	Ted
...	...
Bill	Edward
Charles	Thomas
Celine	Doug

Repeated Measures	
Participant	*Sequence*
Elisabeth	A, B
Sven	A, B
Amanda	A, B
Claudia	A, B
Terry	A, B
Nigel	A, B
Barry	A, B
...	...
Ben	B, A
Michael	B, A
Richard	B, A

Statistical Analysis

Answers questions such as:

- Is there a statistically significant difference between system A and B? [hypothesis testing]
- How large is the difference? [point estimation, averages]
- How accurate are the results? [standard deviation, confidence intervals]

with statements such as:

- We are 95% certain that the time to perform task X is 4.5 ± 0.2 minutes.
- System A is faster than system B at the level $p < 0.2$.
  [20% chance that B is faster, but still choose A since odds are 4:1.]

Sample Size (How Many Test Users?)

- Depends on desired confidence level and confidence interval.
- Confidence level of 95% often used for research, 80% ok for practical development.
- Nielsen: survey of 36 published usability studies [Nielsen 1993b, pages 166–169]. Rule of thumb: 16-20 test users.
- If the differences are small, you might need 50 or 100 or more users to detect a statistically significant difference.

Case Study: Touchscreen Toggle Design

Study by Catherine Plaisant, University of Maryland:

- Home automation system with touchscreen display.
- Toggles (on/off switches) for lighting, climate control, security, etc.
- Six toggle designs: 1-Button, Rocker, 2-Button, Words, Slider, and Lever.
- Formal experiment with 15 novice users (undergrad students).
- Measures:
  - Objective: error rate.
  - Subjective: user satisfaction.

  [Performance (time) was apparently not measured.]
- Tech report [Plaisant and Wallace 1990] and short paper [Plaisant and Wallace 1992a]
- Video made in 1991, also shown at CHI'92 [Plaisant and Wallace 1992b]. [Video: `https://youtu.be/wFWbdxicvK0`]
- The video was successfully cited as prior art when Apple lost their law suits against HTC [Mueller 2012] and Samsung [Beem 2016] for infringement of the slide-to-unlock patent. [Video: `https://channel4.com/news/apple-touch-screen-patent-war-comes-to-the-uk`] [00:39-02:11]

See Plaisant [2016] for more on the work at HCIL and Pickering [2017] for a discussion of implementing toggle buttons on web sites.

Pros and Cons of Formal Experiment

++ collects quantitative measurement data (bottom-line data)

++ collects both objective and subjective measures

++ allows comparison of alternative designs

\- usable only later in development process

\- requires facilitator expertise

\- cannot provide why-information (process data)

\- - needs significant number of test users (20 or more)

9.6 A/B Testing

An A/B test is a summative usability testing method:

- Controlled experiment on a live web site with its real live users.
- A proportion of visitors are randomly assigned to a variant (B) of the web site (they have a slightly different experience), the others see the standard web site (A, the control).

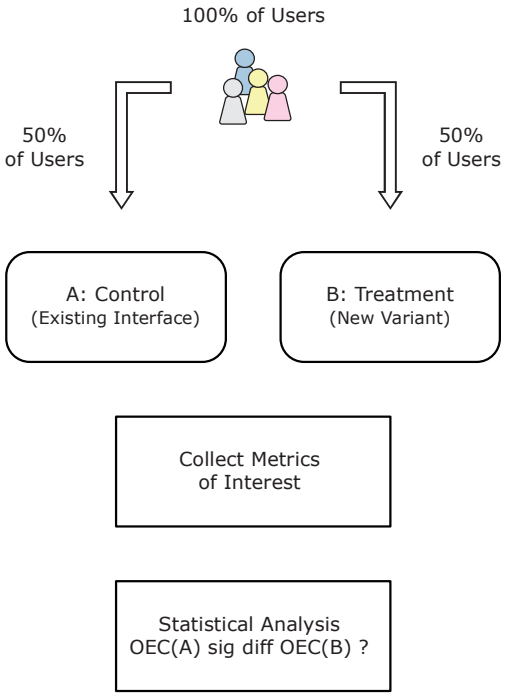

Figure 9.21: An A/B test is a controlled experiment on a web site involving its real live users. A proportion of users see a modified version and are then compared to a control group.

- A cookie is usually assigned, so that individual users always see the same variant.
- A metric (the *overall evaluation criterion*, or OEC) such as click-through rate is measured for each variant.
- The difference in OEC is examined for statistical significance.
- Originally used in marketing [Hopkins 1923], where direct mail with variants of brochures elicited varying response rates (go with the best one).
- Online A/B testing was pioneered at Amazon.com.
- Also called *split testing*, *bucket testing*, and *multivariant testing*.

See Figure 9.21.

References

++ Ronny Kohavi, Diane Tang, and Ya Xu; *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*; Cambridge University Press, 02 Apr 2020 [Kohavi et al. 2020]

++ Ronny Kohavi et al; *Controlled Experiments on the Web: Survey and Practical Guide*; Data Mining and Knowledge Discovery, Vol. 18, No. 1, Kluwer, Feb 2009, pages 140–181. doi:10.1007/s10618-008-0114-1 [Kohavi et al. 2009]

++ Ronny Kohavi and Stefan Thomke; *The Surprising Power of Online Experiments*; Harvard Business Review, Sep 2017. https://hbr.org/2017/09/the-surprising-power-of-online-experiments https://optimizely.com/resources/hbr-online-experiments/

++ Ronny Kohavi et al; *Seven Rules of Thumb for Web Site Experimenters*; doi:10.1145/2623330.2623341 http://exp-platform.com/Documents/2014%20experimentersRulesOfThumb.pdf

+ Thomas Crook et al; *Seven Pitfalls to Avoid when Running Controlled Experiments on the Web*; Proc. KDD 2009, Paris, France, Jun 2009, pages 1105–1114. doi:10.1145/1557019.1557139 [Crook et al. 2009]

+ Ronny Kohavi and Roger Longbotham; *Online Experiments: Lessons Learned*; IEEE Computer, Vol. 40, No. 9, Sept 2007, pages 103–105. doi:10.1109/MC.2007.328 [Kohavi and Longbotham 2007]

+ Ronny Kohavi et al; *Practical Guide to Controlled Experiments on the web: Listen to Your Customers not to the HiPPO*; Proc. KDD 2007, San Jose, California, Aug 2007, pages 959–967. doi:10.1145/1281192.1281295 [Kohavi et al. 2007]

◦ Bryan Eisenberg and John Quarto-vonTivadar; *Always Be Testing: The Complete Guide to Google Website Optimizer*; Sybex, Aug 2008. ISBN 0470290633 (com, uk) [Eisenberg and Quarto-vonTivadar 2008]

◦ Claude Hopkins; *Scientific Advertising*; Lord & Thomas, 1923. Copyright expired. Freely available in PDF. [Hopkins 1923]

Online Resources

++ Ronny Kohavi; *Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO*; Industry Day talk at CIKM 2008 [Kohavi 2008].

++ Ronny Kohavi; *Trustworthy Online Controlled Experiments at Large Scale*; Talk at Decisions 2018 [Kohavi 2018].

+ Shanelle Mullin; *The Complete Guide to A/B Testing*; https://shopify.com/blog/the-complete-guide-to-ab-testing

+ Wikipedia; *A/B testing*; http://en.wikipedia.org/wiki/A/B_testing

+ Microsoft; *ExP Platform*; http://exp-platform.com/talks/

◦ G2 Crowd; *Best A/B Testing Software*; https://g2crowd.com/categories/a-b-testing

◦ Visual Website Optimizer; https://vwo.com/resources/case-studies/

++ abtests.com at Internet Archive; https://web.archive.org/web/20120225010041/http://www.abtests.com:80/browse

+ wishpond; *50 A/B Split Test Conversion Optimization Case Studies*; https://blog.wishpond.com/post/98235786280/50-a-b-split-test-conversion-optimization-case-studies

◦ Bryan Eisenberg; *A/B Testing for the Mathematically Disinclined*; ClickZ, 07 May 2004 https://clickz.com/ab-testing-for-the-mathematically-disinclined/71201/

Videos: Ron Kohavi on A/B Testing

- Ronny Kohavi's talk at CIKM 2008 [Kohavi 2008, 00:00-14:40] is a great introduction. [Video: https://videolectures.net/cikm08_kohavi_pgtce/]

- Ronny Kohavi's talk at Decisions 2018 [Kohavi 2018] is a great update [33 mins.] [Video: https://youtu.be/kTAFOCynWIg]

Running an A/B Test

- The interface variants should be equivalently well implemented. Before a design makes it as far as an A/B test, it usually undergoes several iterations of formative usability testing.
- Multiple variants can be tested simultaneously (sometimes called A/B/N or A/B/Z tests).
- The test users are normal visitors to the site and usually unknowingly take part in an A/B test.
- It is good practice to start with a small fraction of users (say 1%) assigned to the test.
- After a few hours, compute a small number of "guardrail" metrics, to ensure that the new feature is not harming overall system integrity [Kohavi 2018, 11:40-12:30].
- Then gradually ramp up over several days to 10%, 20%, or 100% (=50% in each condition).
- After a day, compute a swathe of (a 1,000 or more) metrics, with email alerts for significant differences, to guard against any surprise effects in other parts of the system [Kohavi 2018, 12:30-13:16].
- An experiment typically runs for several weeks, so as to balance out any cyclical or calendar effects (more purchases at weekends, more visits by children during school holidays, etc.).
- The user base must be large enough (many thousands of users per week), so that there is a reasonable chance of reaching statistical significance.
- Problems and biases in the design or statistical analysis can be detected by running an A/A test (the same interface against itself). The result should be no statistically significant differences in all metrics.

Lessons Learned in Running A/B Tests

Based on Ronny Kohavi's insights [Kohavi 2018]:

1. *Choosing a good metric as the OEC is really hard*: Ideally, look for short-term metric which predicts long-term benefit [Kohavi 2018, 13:21-22:38].
2. *Most ideas fail*: At Microsoft, around ⅓ ideas are positive, ⅓ are flat (no sig. diff.), and ⅓ are negative [Kohavi 2018, 22:38-24:59]. If a new feature shows no statistically significant improvement (flat), probably better not to ship it (unnecessary extra code).
3. *Small changes can have a big impact, but these are rare*: When a small change makes a big difference, they become the subject of a keynote talk! [Kohavi 2018, 24:59-26:40].
4. *Most progress is made incrementally by small changes*: Extremely rare for an experiment to improve OEC by > 2%. Ship once a month rather than every three years. [Kohavi 2018, 26:40-28:02].
5. *Validate the experimentation system*: Carefully check the stats, filter out bot traffic, run A/A tests [Kohavi 2018, 28:02-31:58].

   Remember Twyman's Law: "Any figure that looks interesting or different is usually wrong."

Examples of A/B Testing

- Microsoft runs around 2,000 A/B tests a month [Kohavi 2018, 06:00-06:09]. Every new feature goes through an A/B test before it ships.
- Google puts every design change through an A/B test [Holson 2009; Walker 2009]. "Let data decide.", as Marissa Mayer out it [Mayer 2009, 06:33-07:39]. [Video: https://aiga.org/video-makethink-2009-mayer]

  Google famously tested 41 shades of blue to find the best colour for links. Adding a little red to the blue link colour increased the click rate [Mayer 2009, 07:39-11:08].

  Google's former Lead Visual Designer, Douglas Bowman, resigned over the issue [Shankland 2009; Bowman 2009].
- Amazon has run many experiments to optimise its shopping cart [Eisenberg 2008].
- The BBC uses A/B testing for different design elements on its web sites [Hampson 2010].
- Many Facebook ad campaigns rely on A/B testing. Ad agency TBG typically tries out 27,000 versions of an ad [BBC 2011, 30:40-32:28]. [Video: https://dailymotion.com/video/xmso1d?start=1840]
- Luis von Ahn at Duolingo runs many A/B tests, including to discover the best ordering for teaching modules [von Ahn 2014, 22:45-24:08] and the best way to get users to come back [von Ahn 2014, 19:15-19:40]. [Video: https://youtu.be/FU47HMHPQRs?t=22m45s] [Video: https://youtu.be/FU47HMHPQRs?t=19m15s]
- Mark Pincus at Zynga is very enthusiastic about A/B testing [Pincus 2009, 46:36-47:50]. [Video: https://vimeo.com/412860029#t=46m36s]
- Amazon (and others) have experimented with random price tests: setting different prices for the same product to determine the optimal price (so-called *differential pricing*) [Martinez 2000; Ramasastry 2005; Useem 2017]. Amazon CEO Jeff Bezos said it was "a mistake" to have done so and they would not do it again [PSBJ 2000].

9.7 Post-Test Interviews

A post-test interview is a formative usability testing method:

- Ask test users questions *after* they have used the system to perform representative tasks.
- Useful supplement after a thinking aloud test.
- Provides *subjective* qualitative feedback about users' view of system: preferences, impressions, attitudes.

Post-Test Interview Procedure

- Let user speak thoughts first: "So, how was it?".
- Top-down: probe high-level issues from topic guide first, then more detailed questions about each task.
- Probe specific issues arising from test notes.
- Accept questions from any observers (should be written on a slip of paper for the facilitator to ask).

- Interviews should be transcribed into words for later reference and for full-text search.

Pros and Cons of Post-Test Interview

++ flexible: facilitator can probe interesting issues

\+ collect subjective user feedback about system

\+ simple

\+ cheap

\- hard to analyse and compare

9.8 Post-Test Questionnaires

A post-test questionnaire is a summative usability testing method:

- Written, structured form is given to users to fill out, after they have used the system(s) being evaluated.
- Collects quantitative data (ratings and preferences) which can be analysed statistically.
- Can also collect some additional qualitative feedback.
- Note, however, that designing truly unbiased questionnaires and surveys is a discipline of its own [Foddy 1994; Dillman 1999].

Styles of Question

1. **General**: age (range), gender, occupation, educational level, etc.

2. **Open-Ended**: suggestions, comments.

   I found the following aspects particularly easy to use (please list 0–3 aspects):

   _____
   _____
   _____

   Open-ended questions collect qualitative responses for formative feedback, but which cannot be analysed statistically.

3. **Likert Scale**: Quantitative ratings to judge user's agreement with specific statement (5, 6, or 7 point scale best).

   Overall, I found the widget easy to use.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
☐	☐	☐	☐	☐

   5 and 7 point scales offer the user a midpoint (a fence to sit on). Use a 6 point scale to force users to jump one way or the other. More than 7 point scales provide too little distinction between neighbouring points.

4. **Semantic Differentials**: Quantitative ratings with sliding scale between opposing pairs of adjectives (5 or 7 point scale best).

   Circle the number most closely matching your feelings about the interface.

Simple	3	2	1	0	1	2	3	Complex
Professional	3	2	1	0	1	2	3	Unprofessional
Reliable	3	2	1	0	1	2	3	Unreliable
Attractive	3	2	1	0	1	2	3	Unattractive

   Externally, on the questionnaire, the user sees unbiased ratings from 3 down to 0 and back up to 3. Internally, for statistical analysis, these ratings are converted to scores between 0 points (the worst rating) and 6 points (the best rating).

5. **Overall Preference**: A vote for one item from a set of choices.

   Overall, which hierarchy browser did you prefer?

Tree View (Windows explorer)	☐
Hyperbolic browser	☐
Treemap	☐
Information pyramids	☐

6. **Multi-Choice**: boxes to tick. [Tick just one box, tick multiple boxes, yes and no boxes]

   Which methods do you use to get help (tick any that apply)?

Context-sensitive help	☐
On-line manual	☐
Printed manual	☐
Google search	☐
Ask a colleague	☐

7. **Ranked**: place items in order.

   Please rank the usefulness of these methods (1 most useful, 2 next, 0 if unused)?

Menu-selection	☐
Button	☐
Accelerator Key	☐
Command line	☐

Standardised Usability Questionnaires

There are numerous standardised post-test (after a user has used the system) usability questionnaires:

- SUS (System Usability Scale); free; 10 ratings on 5-point Likert scale [Brooke 1996; Brooke 2013; Sauro 2011]. An example SUS questionnaire can be seen in Figure 9.22.

  Odd-numbered ratings are scored from 0 (left=worst) to 4 (right=best) points. Even-numbered ratings are scored from 4 (left=best) to 0 (right=worst) points. The points total is then multiplied by 2.5 to give a score between 0 and 100.

  Then convert score into a percentile, see `https://measuringu.com/sus/`.

- UEQ (User Experience Questionnaire); free; 26 ratings as 7-point semantic differentials [UEQ 2020]. An example UEQ questionnaire can be seen in Figure 9.23.

PARTICIPANT NAME: DATE:

System Usability Scale

For each of the following statements, please mark one box that best describes your reactions to TreeTest today.

		Strongly disagree				Strongly agree
1.	I think that I would like to use TreeTest frequently.	1	2	3	4	5
2.	I found TreeTest unnecessarily complex.	1	2	3	4	5
3.	I thought TreeTest was easy to use.	1	2	3	4	5
4.	I think that I would need the support of a technical person to be able to use TreeTest.	1	2	3	4	5
5.	I found the various functions in TreeTest were well integrated.	1	2	3	4	5
6.	I thought there was too much inconsistency in TreeTest.	1	2	3	4	5
7.	I would imagine that most people would learn to use TreeTest very quickly.	1	2	3	4	5
8.	I found TreeTest very cumbersome (awkward) to use.	1	2	3	4	5
9.	I felt very confident using TreeTest.	1	2	3	4	5
10.	I needed to learn a lot of things before I could get going with TreeTest.	1	2	3	4	5

Comments (optional):

Figure 9.22: An SUS questionnaire for a system called TreeTest, generated by usabiliTEST's free SUS PDF Generator [usabiliTEST 2020].

- QUIS (Questionnaire for User Interaction Satisfaction); commercial; 41 (upto 122) ratings as semantic differentials on 9-point scale [HCIL 2020].
- SUMI (Software Usability Measurement Inventory); commercial; 50 ratings on 3-point Likert scale [SUMI 2020].
- WAMMI (Website Analysis and MeasureMent Inventory); commercial; 20 ratings on 5-point Likert scale [WAMMI 2020]. See `http://wammi.com/samples/` and `http://wammi.com/demo/`.
- SUPR-Q (Standardized User Experience Percentile Rank Questionnaire); commercial; 8 ratings, 7 on 5-point Likert scale, 1 on 11-point NPS scale [SUPR-Q 2020].

The commercial providers usually offer the possibility to compare your own ratings with benchmark ratings across similar products or industries.

Net Promoter Score (NPS)

- Single rating of customer loyalty, based on *likelihood to recommend* (LTR).
- invented by Frederick Reichheld [Reichheld 2003; Reichheld and Markey 2011].
- Used by vast numbers of companies and organisations [Colvin 2020; Sauro 2012].
- 11-point Likert scale, as shown in Figure 9.24.
- Score: 9 or 10 → Promoter, 7 or 8 → Passive, 0 to 6 → Detractor.
- NPS = % of Promoters - % of Detractors [ranging from -100 to 100]
- NPS of 10 means 10% more Promoters than Detractors.

1	annoying	☐	☐	☐	☐	☐	☐	☐	enjoyable
2	not understandable	☐	☐	☐	☐	☐	☐	☐	understandable
3	creative	☐	☐	☐	☐	☐	☐	☐	dull
4	easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn
5	valuable	☐	☐	☐	☐	☐	☐	☐	inferior
6	boring	☐	☐	☐	☐	☐	☐	☐	exciting
7	not interesting	☐	☐	☐	☐	☐	☐	☐	interesting
8	unpredictable	☐	☐	☐	☐	☐	☐	☐	predictable
9	fast	☐	☐	☐	☐	☐	☐	☐	slow
10	inventive	☐	☐	☐	☐	☐	☐	☐	conventional
11	obstructive	☐	☐	☐	☐	☐	☐	☐	supportive
12	good	☐	☐	☐	☐	☐	☐	☐	bad
13	complicated	☐	☐	☐	☐	☐	☐	☐	easy
14	unlikable	☐	☐	☐	☐	☐	☐	☐	pleasing
15	usual	☐	☐	☐	☐	☐	☐	☐	leading edge
16	unpleasant	☐	☐	☐	☐	☐	☐	☐	pleasant
17	secure	☐	☐	☐	☐	☐	☐	☐	not secure
18	motivating	☐	☐	☐	☐	☐	☐	☐	demotivating
19	meets expectations	☐	☐	☐	☐	☐	☐	☐	does not meet expectations
20	inefficient	☐	☐	☐	☐	☐	☐	☐	efficient
21	clear	☐	☐	☐	☐	☐	☐	☐	confusing
22	impractical	☐	☐	☐	☐	☐	☐	☐	practical
23	organised	☐	☐	☐	☐	☐	☐	☐	cluttered
24	attractive	☐	☐	☐	☐	☐	☐	☐	unattractive
25	friendly	☐	☐	☐	☐	☐	☐	☐	unfriendly
26	conservative	☐	☐	☐	☐	☐	☐	☐	innovative

Figure 9.23: An example UEQ questionnaire, recreated by Keith Andrews from the example at UEQ [2020].

How likely are you to recommend this product to a friend or colleague?

Not at all likely 0 1 2 3 4 5 6 7 8 9 10 Extremely likely

Figure 9.24: The Net Promoter Score (NPS) is based on a single rating, using an 11-point Likert scale.

Pros and Cons of Post-Test Questionnaire

++ can provide quantitative data (easy to analyse and compare)

\+ easy to repeat (recognise trends)

\+ simple

\+ cheap

\- less flexible than interview

9.9 Usage Studies

Usage studies are exploratory usability testing methods:

- Users' actual activity is estimated or recorded and then analysed.

- Three kinds:
  - Diary Studies
  - Software Logging
  - Observational Studies

Purpose of a Usage Study

- Insight into how long users spend on each activity.
- Insight into which software is used for what purpose.
- Insight into which features of software packages are most used or unused.

9.9.1 Diary Studies

- Users are asked to keep a diary (or logbook) of their usage of a system over several days or weeks (self-reporting).
- Repeat for several users.
- The diary entries are converted into estimates of the amount of time spent on various activities.
- Statistically analyse the resulting data.

Pros and Cons of Diary Studies

\+ anecdotal evidence (better than nothing?)

\- subjective estimates made by users

\-- self-reporting is highly unreliable

9.9.2 Software Logging

- An instrumented version of the software logs all user interactions.
- Users must give their informed consent.
- Can recruit a larger sample of test users (20–50+).
- Gather and aggregate the various log files.
- Statistically analyse the resulting data.

Pros and Cons of Software Logging

\+ objective log file data

\- all the software of interest must be instrumented

- hard (impossible) to infer the user's intentions and motivations

9.9.3 Observational Studies

- Record one or more typical days of use of a system (screen capture and user video).
- Users must give their informed consent.
- Repeat for several users.
- Manually analyse the recordings and encode the activities (begin and end) in a timeline.
- Statistically analyse the resulting data.

References

- Byrne et al; *A Day in the Life of Ten WWW Users*; unpublished paper, 2000. [Byrne et al. 2000]
- Byrne et al; *The Tangled Web We Wove: A Taskonomy of WWW Use*; [Byrne et al. 1999]

Finding Willing Users

Users are often reluctant to participate in a usage study, because they feel it is an invasion of their privacy.

Over several meetings and many hours, explain to potential test users:

- Exactly what will be recorded.
- That they can turn off the recording at any time.
- Exactly how the data will be analysed.
- That statistics will be aggregated on activities and software usage over several users.
- That statistics of individual user performance will not be aggregated.
- That individual users will not be identifiable in any reports or publications.
- That you will seek specific permission before showing or publishing and video, photographs, or screenshots.

Pros and Cons of Observational Studies

++ objective analysis of usage (not self-reporting)

- often difficult to find willing users

- - video analysis is extremely time-consuming

9.10 Remote Usability Testing

As well as testing with users face-to-face in the same physical location, it is possible to run usability tests remotely:

- *Moderated*: The facilitator moderates the test remotely.
- *Unmoderated*: The test user works alone without a facilitator.

References

+ Inge De Bleecker and Rebecca Okoroji; *Remote Usability Testing*; Packt Publishing, 23 Aug 2018 ISBN 1788999045 (com, uk) [Bleecker and Okoroji 2018]

◦ Nate Bolt and Tony Tulathimutte; *Remote Research: Real Users, Real Time, Real Research*; Rosenfeld, 01 Feb 2010 ISBN 1933820446 (com, uk) [Bolt and Tulathimutte 2010]

◦ William Albert, Tom Tullis, and Donna Tedesco; *Beyond the Usability Lab: Conducting Large-scale Online User Experience Studies*; Morgan Kaufmann, 01 Feb 2010 ISBN 0123748925 (com, uk) [Albert et al. 2010]

Online Resources

+ Kate Moran and Kara Pernice; *Remote Moderated Usability Tests: Why to Do Them*; NNGroup, 12 Apr 2020 `https://nngroup.com/articles/moderated-remote-usability-test-why/`

+ Kate Moran and Kara Pernice; *Remote Moderated Usability Tests: How to Do Them*; NNGroup, 26 Apr 2020 `https://nngroup.com/articles/moderated-remote-usability-test/`

+ Amy Schade; *Remote Usability Tests: Moderated and Unmoderated*; NNGroup, 12 Oct 2013 `https://nngroup.com/articles/remote-usability-tests/`

+ Rebecca Costa; *Remote User Testing: Your Guide*; Justinmind blog, 28 Nov 2019 `https://justinmind.com/blog/remote-user-testing/`

+ Salma Patel; *Remote User Research Tips*; 22 Mar 2020 `https://salmapatel.co.uk/user-research/remote-user-research-tips/`

+ Behzod Sirjani; *Conducting Remote Research*; 47-min video talk/interview, 02 Apr 2020 `https://userinterviews.com/blog/remote-user-research-tips-from-slacks-head-of-research-and-analytics-ops`

Specialist Remote Testing Tools and Services

- UserZoom; `userzoom.com`.
- Lookback LiveShare; `lookback.io`.
- Userbrain; `userbrain.net`.
- UserTesting; `usertesting.com`.
- TryMyUI; `trymyui.com`.
- Userlytics; `userlytics.com`.
- UX Testing; `uxtesting.io`.
- Playbook; `playbookux.com`.

Some tools support only moderated remote testing, some only unmoderated, others both.

Some services will organise remote test users for you (at extra cost).

Generic Videoconferencing Tools

- Cisco Webex; `webex.com`.
- Zoom; `zoom.us`.
- Skype; `skype.com`.
- GoToMeeting; `gotomeeting.com`.
- Google Meet; `meet.google.com`.
- jitsi; `jitsi.org`.

Pros and Cons of Remote User Testing

\+ can reach global user base

\+ no travel involved

\+ user's natural device and environment

\- often technical hurdles and difficulties

\- - not the same as being in the room

Remote Test Checklist

1. Preparatory Online Meeting:
   - ☐ Explain test procedure and recording.
   - ☐ Install any local software (Acrobat Reader, Webex native client, ...).
   - ☐ Practice filling and signing with Acrobat Reader.
   - ☐ Check test user's environemnt: internet connection (upload!) is fast enough, browser is installed, screen sharing works, usercam works (well enough), audio/mic works (well enough), recording works (well enough), etc.
   - ☐ Ask user to read, sign, and return consent form (sign using Acrobat and return by file share).
   - ☐ Schedule online test session with user.
2. Opening:
   - ☐ Start online session. Facilitator shares their screen.
   - ☐ Start recording.
   - ☐ Greet the participant.
   - ☐ Go through orientation script and set the stage.
   - ☐ Fill out background questionnaire (facilitator asks questions and fills out PDF form live in Acrobat).
3. Test Session:
   - ☐ Provide any prior training.
   - ☐ Provide training of thinking aloud.
   - ☐ Pass screensharing to test user.
   - ☐ Document test user environment (OS version, browser version).
   - ☐ User begins with tasks.
   - ☐ Provide tasks one by one (as individual PDFs by file sharing).
   - ☐ User begins with first task.
   - ☐ Encourage user to think out loud.
   - ☐ User finishes last task.
4. Closing:
   - ☐ Screensharing stays with test user.
   - ☐ Interview: how was it?
   - ☐ Structured interview questions.
   - ☐ Individual interview questions arising from test.
   - ☐ Feedback questionnaire (user fills out form live in Acrobat and returns by file sharing).
   - ☐ Thank participant, close online session.
5. Wrap-Up:
   - ☐ Summarise thoughts about this test.
   - ☐ Save and check session recordings.
   - ☐ Organise data sheets and notes.

Figure 9.25: A example checklist for a remote thinking aloud test.