Module 2
User research

Introduction

By studying this module, you'll fully understand the importance of research as the key ingredient underpinning UX Design. Research is fundamental to understand the problems we are attempting to solve for users.

You'll learn about the range of different types of research such as qualitative v quantitative, attitudinal v behavioural. You'll also learn how to prepare and carry out a variety of research techniques.

Nothing changes mindsets, deepens understanding and can make or break products like usability testing. No matter where your UX career takes you, usability testing is a core skill that must be mastered. The good news is - like much else in UX - it's not difficult to do.

You'll learn everything you need to know about usability testing. Why you do it, how you do it and when you do it. During this module, you'll learn how to set clear usability test objectives, recruit users, prepare a test script and facilitate and record usability test sessions for desktop and mobile.

You'll learn how to set clear objectives for customer and stakeholder interviews, to prepare scripts and to conduct and record interviews. Talking and listening to users is an essential part of the UX designer's job so the project on depth interviews will help build your confidence in carrying out interviews with users/customers.

You'll also understand how to conduct card sorting exercises, online surveys, heuristic evaluations as well as understanding the concept of A/B testing and how it helps optimise and refine existing software.

Your recommended reading for this module includes Just Enough Research, which gives a great overview of the best research methods that you can easily do yourself.

Steve Krug's Rocket Surgery Made Easy is a small classic about conducting your own usability tests and his book Don't Make Me Think covers a lot of heuristic evaluation material. And we recommend The Mom Test, a great book on customer interviews.

There's a lot going on in this module so, as usual, use the note sheets below to capture key concepts during the video lessons.

In terms of projects, there are four projects to complete during this module:

- Usability tests
- Depth interview
- Online survey
- Competitive benchmark

Topics covered include:

- Research landscape
- Qualitative research
- Quantitative research
- Observational/attitudinal
- Usability testing
- Defining test objectives
- Test scripts
- Finding users
- Desktop / mobile setups
- Card sorting
- Depth interviews
- Online surveys
- Stakeholder interviews
- A/B testing
- Heuristics

Recommended reading

Just Enough Research
Erika Hall
@mulegirl

Rocket Surgery Made Easy
Steve Krug
@skrug

The Mom Test
Rob Fitzpatrick
@robfitz

Don’t Make Me Think
Steve Krug
@skrug

Additional resources

- The Distribution of Users' Computer Skills Worse Than You Think - “One of usability’s most hard-earned lessons is that you are not the user.”
- When to Use Which User-Experience Research Methods
- User research - what’s tomato ketchup got to do with it?
- Ten things I wish I knew as a UX Research team of one
- Never Ask What They Want
- How to find users for a usability test
- UXPin: The Guide to Usability Testing
- Survey Monkey
- Dan Siroker (Optimizely) at Web Summit 2014 - A/B Testing on the Obama campaign
- Seven lessons I learned from the failure of my first startup, Dinnr
- Excuses, Excuses! Why Companies Don’t Conduct User Research - UXmatters
- Jakob Nielsen’s 10 Usability Heuristics - NNGroup
- Elevate Your UX with Heuristic Analysis

Note taking

The research landscape

If you're not doing research,
you're not doing UX

Quantitive (Quant) research. —"hard" science

- structured
- measurable
- numerical
- broad insight
- objective
- larger sample sizes

Qualititive (Qual) research

- unstructured
- hard to measure
- smaller sample sizes
- good insights
- subjective
- "soft" science

Observational

- Google analytics
- AB Testing
- Usability testing

Attitudinal testing

- surveys
- opinions.

I am not the target audience!

↳ Too much info / bias.

↳ The product is not for the professional or insiders.

↳ assumptions are dangerous.

↳ Research lets you test assumptions

Qualitative research

Shallow research = shallow insight
Avoid asking peoples' opinions.
Don't ask loaded questions.

- Unstructured
- not mensurable
- deep insights
- subjective
- open to interpretation
- smaller sample sizes
- 'soft' science

Quantitative research

- Structured /numerical
- measurable
- broad insight
- objective
- statistical
- larger sample sizes
- 'hard' science

Observational research

Observing is more powerful than listening

People are unreliable

Asking directly doesn't always help.

- watching what users do
  - → google analytics
  - → AB Testing
  - → usability testing

Attitudinal research

- Listening to what people say
  - ↳ surveys.
  - ↳ customer feedback.

Bias

- self - referential design
- knowing the answer
- being defensive
- asking leading questions

What is usability testing

Watching the user as software is used

→ user and moderator
↳ record user

→ inexpensive and uncomplicated.

→ Camtasia.
↳ user testing software

Benefits of usability testing

- presents user's experience
  ↳ user's perspective.
- unite stakeholders
- challenge assumptions.
- variety of user data.
- avoid feature debates.
- cost effective

- understand user goals. → primary through tertiary.
- what do people actually do?
- context of use
- how are we helping the user
- identify problems or roadblocks.
- comparative testing.
- Are we valuable? → would they use it?

Defining your test objectives

→ Narrow focus

→ Cover tasks that help us achieve research objectives.

- Define clear goals
- Be specific
- Not a one-off event
- Don't confuse useability testing with functional testing.

Creating a test script

→ Guide for staying on the right track

↳ specific tasks and questions.

↳ keeps test on track.

Introduction: help users relax

↳ explain purpose of session

↳ technical setup

→ Keep in mind: (user):

- we're testing product, not user
- be candid and honest
- tell them to think out loud
- Please ask questions.

Interview:

- easy questions to start
- Describe interactions with similar software
- Understand goals.

Tasks:

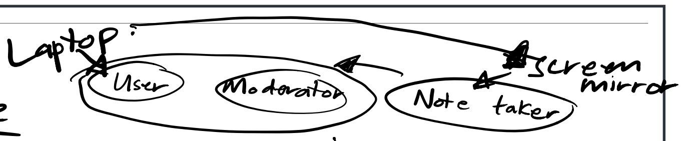
- Details about previous use if natural task exists.
- Pre-defined tasks.
- Lots of 'what' and 'why' questions.
- Clearly explain scenario

Finding users

- Colleagues → comes with bias
  → not great data.
- Friends and family → might be to close to company
- Recruit customers through marketing channels
- Professional recruiters. to find participants
  ↳ define criteria
  - when, where, how many
  - demographics? not always NB
  - product history might be ideal.
  - tech savvy users
  - duration + gratuity

We're not asking people what they want or giving them any kind of commitment

Moderating



Software

Desktop:
- camtasia.

Mobile:
- reflector

★Screenflow to record

Tips

- Ask the right type of questions.
- Rehearse
- Know the software you're testing.
- Don't tell users what to do
- Never take control during a task
- Read verbatim from the script.
- Practice!

- Don't ask yes/no questions — "what" and "why" rather
- No hypothetical or design questions.
- Specific Task questions

What to test

Existing product:

- current version
- prototype
- competitors.
- peers.

New product:

- competitors.
- peers
- prototype

When to test

Start early → test often.

Test in all phases before build.

One last test before launch

Understand problem during research

Validate design during prototype

Sanity check right before launch

Consent Forms are important

Online surveys

Easy to run
Quant and Qual data
Powerful — large number of responses
Unambiguous
Cheap.

Structured questions — multi choice (Quant)

Unstructured questions — own answer (Qual)
— interpretation needed.

Three golden questions
— Why did you visit website or app (goals)
— Were you able to complete task (and why not)
— What improvements would you make

Never more than 10 questions
Keep questions on one page
Don't ask redundant or useless questions

Customer interviews

Understand goals — multiple goals.
Understand context of use — where they're using it
— social aspects of use.

Don't ask for product feedback during interview — rather find out if it is useful

Don't ask for design feedback.

Don't ask hypothetical or questions for future

- Ask about their life
- Ask about specific things and experience
- Ask about what they use and how used.
- Validate assumptions with follow up (non-leading) questions.
- Interview → NOT interrogation
- Always record the interview

Stakeholder interviews

Understand business and tech goals.

Understand problems with product

Understand competitive landscape

↳ Get buy-in

Avoid talking features and solutions

Stakeholder brief:

- summary
- focus on agreements
- call out contradictions
- alignment with project goals.

Give focus – show stakeholders that we're listening

Card sorting

- A good way to organise content and/or product flow
- See user's mental model and terminology
- Patterns will emerge

Closed card sort – predetermined categories

Online tools or with post-its.

A/B testing

Two versions of same page

- run simultaneously and test usage

Multi - variant → combinations of different elements

Statistical data to back decisions

Competitive benchmarking

How do best in class solve problems

What can we emulate

What can we avoid or improve

Conventions to follow.

Similar industries.

Software should be interested in me

Heuristics → best practice guidelines.

Software should be personal.

Learn more through frequent use

Recognise user

Software should be forthcoming

Give me known information at start of user journey

No cryptic error messages.

Gray out unavailable options.

Software should be confident

Better options for doing and undoing actions.

Follow through on user actions

Don't always assume things are done by user in error.

Software should have common sense

Sensible flow of actions

Remove illogical options (or at least de-prioritise them)

Just because you can present options, doesn't mean you should.

Visibility of system status

Let people know what's going on.

Status message as action is done and as goal is reached.

Confirmation page.

Match between system and real world

Use real terminology

Language that users outside of your business can understand.

No ambiguity

Don't let machine take control of the communication

Freedom and control

Allow user feel in Control of actions and navigations.

Allow user to undo actions.

Recognition rather than recall

Give more info so that user Can more easily remember what they're working with

Don't expect user to remember very specific info

Don't force me your way

Don't frustrate user unnecessarily

Reformat data rather than forcing user to go back and fix.

Save me steps whenever possible

Save users time.

Needless timewasting frustrates users.

Better ways of entering data