Empathy

* Observe
* Engage
* Immerse

Tacit Knowledge: Knowledge that is derived from internalized information and experiences that are hard to be put into words

Process vs Practice

* Predefined workflow, established rules
* Actions taken when rules fall apart

Slips vs Mistakes

* Small errors that prevent completion
* Fixed with small improvements
* Errors users cannot correct
* Design easier mental model or provide more information

How to document findings?

* Notes: Can’t capture the entirety of observation, only critical points
* Video: Rich data, reviewing/transcribing process is lengthy, intrusive
* Audio: Less intrusive, reviewing/transcribing process is lengthy

Open ended questions

* No yes or no questions
* Better but not always good
* Questions can be too complicated to think of an answer on the spot
* Some silence during interviews can be good

Questions to avoid:

* Biased / vague questions
* Do you like the UI?
* Can’t ask because interviewee would be pressured to say yes
* Use interviews to learn about user’s perception of your design, not it’s usability

Set goals and practice

Online or onsite interviews?

* User convenience and comfort
* Interview convenience
* Context and examples
* Bias

Immersion: Observation strategies for improvement

* Contextual inquiry: Observing people in their natural context and asking questions later to fill the gaps
* User lead: Learn from “lead users” who find interesting workarounds to your problem and adapt them to your solution
* Diary studies: Record every day

Establishing Requirements

* We don’t choose requirements
* Requirements come from understanding user’s needs
* Require clarification refinement and re-scoping
* Can be justified and relatable

System requirements

* Functional
* What does the system do?
* Non-functional
* What should it be? Maintainable, fast usable
* Quantifiable
* Data
* What data does it need to store?

Contextual requirements

* Physical: Environment product is used in.
* Does it need to be waterproof?
* Social: Sharing of information across distances
* Organisational: Hierarchy, user support, communications structure and infrastructure

User requirements

* Characteristics
* Expertise
* Novice: Clear instructions, constricted path
* Expert: Flexibility
* Frequency
* Infrequent: Clear instructions
* Frequent: Short cuts

Personas

* Focus on most important ones
* Don’t give every single detail
* Balance with time constraints
* Capture important differentiating features
* If you cannot make a rigorous person, make provisional personas.

Persona Types in order of importance

* Primary
* Secondary
* Supplemental
* Customer
* Served

Constructing Personas

* Identify behavioural variables: Activities, goals, age range
* Plot interviewed subjects into behavioural variable range: Novice to expert, price to service conscious, necessity to entertainment
* Identify significant behavioural patterns: Find clusters of common patterns to form basis of a persona
* Synthesize characteristics and goals: Infer goals that lead to behaviour pattern
* Bullet points of characteristics
* Check for redundancy and completeness: Perform additional research to find missing behaviours
* Expand on bullet points
* Designate persona type

Empathy map: Tool to help synthesize observations to create persona

* Say
* Do
* Think
* Feel

Helps with

* Identifying needs
* Communicating with stakeholders
* Collect data from user

Context Scenarios:

* Informal narrative story about a situation with product
* Avoid specifying how products work

Key path scenarios:

* Concrete UI
* Interactions with UI described
* Most important paths only

Validation scenarios:

* Concrete UI
* Shorter than key path
* Used for what if questions

Use case:

* Formal sequence of actions performed on a specific UI

Mental Model

* Designer’s mental model becomes conceptual model for system
* There will be gaps between the conceptual model and the user’s mental model

Conceptual Model

* High level description of how a product is supposed to be used
* Objects exposed to users
* Actions users can perform on each object
* User visible attributes of objects
* Relationships between objects

Developing Conceptual Models

Lexicon

* Consistent naming conventions in UI and documentation- Helps users learn objects, attributes and actions
* Not implementation objects: User’s language independent from implementers
* Lexicon requires more policing and discussion

Metaphors

* Things that are metaphors to represent functions, magnifying glass to zoom
* Mapping what the functions are to users
* Easier to use as it resembles real life examples
* User’s familiar knowledge on everyday items adapted to unfamiliar technology
* Gives users a language to discuss interactions
* Can be misleading, does not scale well, degrade overtime

Interaction styles

* Instructing: One way verbal or textual instructions
* Conversing: Two way
* Manipulating: Interact directly with objects
* Exploring: Move around in a space

Direct manipulation interfaces (Dragging to enlarge or shrink objects)

* Immediate feedback
* Continuous representation of object

Norman’s Action Cycle

Diagram

Description automatically generated

Timeline

Description automatically generated

Lesser gulfs the better

Reducing Gulf of Execution by making it easier to:

* Determine function of device
* Tell what actions are possible
* Find mapping: Intention -> Physical movement
* Perform the action

Reducing Gulf of Evaluation by making it easier to :

* Tell what state the system is in
* Find mapping: system state ->interpretation
* Remember desired system state

Ideation

* Imaginative
* Diverse perspectives
* Explore unexpected areas

Brainstorming guidelines

* Prepare setting
* Gather participants
* Set topics
* Mental warm up
* Brainstorm
* Quantity
* Defer judgement
* Encourage wild ideas
* Build on early ideas
* Use sticky notes
* Vote on favourites

After Brainstorm

* Define problem clearly
* Who are users?
* What activities?
* What breakdowns to overcome?
* Define solution clearly
* What ideas to use?
* How will users interact with system?

Rapid Prototype

* Build prototypes rapidly and cheaply
* Learn quickly
* Explore more possibilities
* Fail quickly and fail often
* Quickly test user interaction
* Gain more insights

Prototype

* Not required to be complete
* Easy to change
* Can be retired

Set measurable goals

Two rooms of creativity

* Creativity of making options
* Creativity of making the criteria to choose which option

What should a prototype evaluate?

* Function
* Form
* Experience
* Costs increase over time, building and evaluating prototypes early will help avoid expensive modification costs later
* Quantity > Quality because of experience with making lots of prototypes rather than sitting and thinking about quality
* But possibility can miss out on most high-quality idea
* Design moves from divergence > convergence
* Design multiple prototypes, make decisions to make it more specific

Functional Fixation

Parallel Prototype

* Two simultaneous prototypes combine feedback of both into one final prototype

Serial Prototype

* One prototype with multiple stages of prototype
* Find one good idea and continuously improve on it

Steal good ideas

Prototyping Methods

* Low Fidelity
* Nav Diagrams
* Storyboards
* Paper prototypes
* Faster and easier to change
* Draws attention to conceptual model and not aesthetic design
* Medium Fidelity
* Powerpoint
* Basic interactivity with hyperlinking
* POP
* Balsamiq
* Focuses on structure and content, black and white like paper prototype
* Figma
* High Fidelity
* Justin Mind
* Axure
* Partial Implementation

Form Prototype: Prototype that focuses on aesthetics

Experience Prototype: Prototype that tries to mimic the experience of using solution

Functional Prototype: Prototype that works (Barebones only the function)

Input

Keyboard (Text Entry)

* Command Keys
* Generic function keys
* Cursor movement
* Numeric keypad

DVORAK Keyboard

* Alternate hand typing
* Most common letters easiest to type
* Least common bottom row
* Right-handed

Nokia phones (Multi Tap mobile text entry)

* Multiple presses per letter

Resolving Ambiguity

* One press per letter
* Dictionary lookup

Smartphone keyboard (Mobile text entry soft keyboards)

* On screen keyboards
* Handwriting recognition
* Touch/ Stylus

Speech Recognition

* Faster than typing
* Speaking in complete sentences is cognitively challenging

Pointing devices

* Object motion
* Mouse
* Trackball
* Joystick
* Pointing stick
* Touching surface
* Tablet/Stylus
* Touchpad
* Touchscreen

Pointer device properties

* Mapping
* Direct: Pointer space and screen space is unified
* Indirect: Changes in pointer causes changes in screen
* Control Display Ratio
* Per unit of movement in physical space to distance travelled by cursor

Graphical user interface, text, application

Description automatically generated

Fastest to slowest

* Mouse
* Joystick
* Text keys
* Step keys

Fitt’s Law: The quicker you can reach a target object, the more convenient and easy

Diagram, engineering drawing

Description automatically generated

Time taken to move to target Size S which is D distance away given a is start stop time and b is speed

Rectangular Menu or Pie Menu

Pie menu has larger buttons and have less distance from each button

Multifinger

* Using two fingers to zoom

Multipoint

* Two fingers to draw

Multiregion

* Using your entire palm to draw

Multiperson

* Touched by multiple people at once

Multitouch

* Using both hands

Occlusion problem: When things are too close together its hard to press the one u want

Long term memory

* Unlimited size because it’s associative
* No decay

Short term memory (Working memory)

* Size 5-9 chunks
* Decay 1 or 3 chunks (73-226) or (5-34) seconds

Visual Image store

* Size 7-17 letters
* Decay 70-1000 msec

Audio Image store

* Size 4.4 – 6.2 letters
* Decay 900- 3500 msec

Memory Strategy

* Pay attention
* Rehearse
* Chunk- Assosciate items together
* Use cues- Rhythms etc
* Organized
* PQRST Learning steps: Preview question re read study test
* External aids: Memo

Human attention stress and risk to performance

Diagram

Description automatically generated

Medium requirements of attention stress and risk causes highest performance

Trade off of low risk and high risk

Low risk

* Divergent thought
* Exploration/ Simulation
* Safety/ Playfulness
* Freedom to act

High risk

* Convergent thought
* Concentration
* Commitment
* Exhilaration
* Forced to act

Practice to store long term memory

Power law of practice (reason why people still use qwerty over DVORAK

A picture containing diagram

Description automatically generated

Design Implications

1. Representations to direct attention
   1. Cues
   2. Rhymes
   3. Acronyms
   4. Visual representation. 15 game, tictactoe sudoku
   5. Color representation
2. Avoid taxing working memory
   1. Abstractions reduce load (Simplified map of Singapore on MRT map)

Design Thinking

* Empathize
* Define
* Ideate
* Prototype
* Test

Test to make ideas better

* Learn about users
* User reaction to sharing stories
* Build empathy through observation
* Refine point of view
* Unexpected insights
* Redo solutions, reframe problem

Test with UX experts instead of real users

* Easier to recruit UX experts
* Feedback easier to process
* Less ethical practical issues
* Use experts that know problem domain and users

Problems with UX Experts

* Not always easy to find expert with right domain of knowledge
* Expensive
* Miss important problems
* Identify too many trivial problems
* Biases

Inspection Types

* Cognitive Walkthrough
* Give experts UI prototype and walkthrough
* Provide assumptions
* Experts walk through prototype with scenarios

4 Questions of cognitive walkthrough for UI EXPERTS

* Will user really want to do the action?
* Will correct action be visible?
* Will user recognize which action is correct?
* Will user interpret feedback correctly?
* Heuristic Evaluation
* Small set of evaluators examine interface
* Judge compliance with recognized usability principles
* Quick cheap easy

Heuristics

1. Visibility of system status: Letting users know what system knows
2. Match between system and real world: Keep language simple, don’t confuse users with confusing error messages
3. User control and freedom: Clearly marked exists, undo and redo
4. Consistency and standards: Same font, colour scheme, layout, terminology
5. Error prevention: Form validation before sign up
6. Recognition rather than recall: Minimise user memory load
7. Flexibility and efficiency of use: Shortcuts
8. Aesthetic and minimalistic design
9. Help users recognise diagnose and recover from errors: Error messages
10. Help and documentation
    1. Better to design a system that users don’t need help with though

Heuristic Evaluation Process Normal

1. Briefing of experts \*together\*
   1. Explore prototype once
   2. Do a task
   3. Present goals and heuristics
2. Expert’s Evaluation
   1. Independent evaluation
   2. Generate list of usability problems
3. Debriefing, expert’s discussion
   1. Discuss problems
   2. Generate list of problems

HE Process Paper Prototype

1. Briefing of experts individually if possible
   1. 1 and 2 from above combined
2. Debriefing

3-5 experts is the ideal number according to Nielsen

Information Design

People DON’T read online

Web pages must have scannable text

* Highlighted keywords
* Meaningful sub headings
* Lists
* One idea per paragraph
* Start with conclusion
* Half the word count or less

Credibility increased

* High quality graphic
* Good writing
* Links
* No “marketese”

Information Foraging: Metaphor of animals hunting for food to analyse users getting information

Weak Scent Common Symptoms

* Users can’t find wanted information
* Waste time
* Feeling negative
* Give up

Supporting information foraging

* Scannable text
* Informative visual design
* Affordance: Properties of objects that show users what actions they can take
* Principles of grouping
* Typeface
* Colour

Web design patterns

Small multiples

* Eyes make fine distinctions (0.1mm)
* Shrunken high density graphics

Economy of line: Keeping text concise

Visual Design

Monochromatic colour schemes

* Same primary colour, vary brightness

Analogous colour scheme

* Nearby on the colour wheel
* More vibrant colour
* One dominant, others accompaniment

Complementary colour scheme

* One dominant, others high contrast highlights

Diagram

Description automatically generated

Don’t use red of violet for text because infrared and ultraviolet are at the ends of visibility spectrum

Fonts

* Can evoke associations

Serif Font

The serif hypothesis: Easier to read, preferable for long stretches of text, provides anchor that guide reader’s eye.

Competing hypothesis- “Familiarity”: Legibility of text depends on what user is used to.

Typography line terms

A picture containing rectangle

Description automatically generated

Tracking or letterspacing

The o v e r a l l s p a c i n g of a group of letters

Line spacing

This

Is

Line

Spacing

Small capital: For Emphasis

Lower case numbers

Italics

* Roman
* Oblique- *Adding slant to roman*
* Italic

Jan Tschichold: Asymmetric Typography

* Constructive meaningful economical

Grid system

* Key pattern for implementing rationality, modernism, asymmetry
* NO elements are centred

Mobile Devices

Differences of mobile devices

* Multitasking ability
* Different interactions
* Small screens

Tips for mobile design:

* Only ask for permission when the function user is going to access requires it
* Provide text labels for icons
* Let user control zoom
* Give prompts after significant action

Designing for multiple screens with different pixel sizes

100x100 pixel image on 72ppi, 144ppi and 240ppi

The larger the ppi the smaller the image because more pixels are loaded per inch.

App designing tools

DP: Android’s Device Independent Pixels (DP)

PT: iOS’s Points

Evaluating Design

Why Evaluate Design with Users? (Why not experts)

* People don’t know what they want
* Costly, troublesome, many opinions
* Hard to predict what user will do

Goals determine study method

* Field studies
* Approach: Observe in context
* Pros: Realistic
* Cons: Less Precise
* Lab studies
* Observe in controlled environment
* Precise
* Less realistic than field studies
* Surveys
* Ask people questions
* Easy to run
* Less realistic than field or lab studies
* Inspections
* Experts analyse system
* Easy to run, no need to run recruit users
* Expert’s aren’t users

Qualitative Analysis

|  |  |
| --- | --- |
| Qualitative | Quantitative |
| Descriptions | Numbers |
| Hard to measure | Easy to measure |
| Themes and patterns | Statistical Analysis |
|  |  |

Observations

* Generally better
* Not aware of what we do
* Not always able to put thoughts into words

Questions

* Useful also
* Find out demographics
* Subjective assessments

Open ended questions are qualitative

* What did you like about\_\_?

Close ended questions are quantitative

* Binary yes no questions
* Strongly agree -> Strong disagree questions

Qualitative Data advantages

* Helps understand design problems
* Discover causes
* Find solutions
* Helps explain the problems to others

Transcribing qualitative

* Audio video transcriptions
* Useful for analysis
* Time consuming
* Most useful data in transcriptions
* Critical incidents
* Comments
* Questionnaire

Convert qualitative to quantitative through:

* Clustering: 7/10 people find that this task is difficult
* Coding: Putting observations into categories
* Nominal (No ranking): Red, Blue, Purple
* Ordinal (Ranking): Poor=1, Medium=2, Good=3
* Coding not always most reliable

Lab Studies

* Control conditions for precision

Steps

1. Plan the study
   1. Goals: Earlier studies can have nonspecific goals. Later studies better with specific goals. Specific as in 95% of users should feel the system is useful.
   2. Methods
      1. Realistic tasks
      2. Don’t train participants
      3. No bias
      4. No breaking tasks into steps
      5. Sequence of events. Briefing, consent, study
   3. Participants
      1. How many
      2. Who
      3. Data collected anonymous
   4. Environment
      1. Realistic conditions
      2. Simulate real conditions
   5. Data
      1. Ask users to think aloud
      2. Time taken is longer however
      3. Doesn’t always give right answers
2. Recruit Participants
   1. How?
   2. Consider questionnaire and incentives
3. Prepare materials
4. Run study
   1. Dry run study to find problems with process
5. Analyse
   1. Collect data in one place
   2. Summarize analysis
   3. Weigh evidence
6. Recommend
   1. Rethink solution
   2. Rate severity and ease of fixing

UI Tools

* Tools provide standards

Standards

* Streamline processes

Types of tools

* Toolkits
* Frameworks
* Integrated Development Environments
* Text, timeline

  Description automatically generated

iOS interface guidelines

* Clarity: Clear text, negative space
* Deference: Fluid motion, pretty interface
* Depth: Conveying depth and layers well

Toolkits give users predictability, consistent UI. Gives developers path of least resistance, reusing code, easy modifiability, make UI dev accessible.

UI Tools yield better interfaces

* Easier to incorporate changes
* Tools let u put in less effort
* Consistency
* Rapid prototyping

UI Tools lower maintenance costs

* Less code to write
* Better modularization
* Hide complexities
* Higher reliability
* Easier to port app to new hardware/software

UI Tools can discourage change

* Harder to use non-standard widgets
* Hard to fix problems with tools

Web experiments

* Small changes have big differences

A/B Testing

* Evaluate difference and correlation through randomized controlled experiment

Control condition

* What you get when u change nothing

Treatment condition

* What you get when u change something

Valuable Insights

* Conversion rates
* Time taken

Sample population

* Half on control, half on treatment
* Assign to groups randomly
* Determine minimum sample size
* Bigger errors require less sample size to detect
* 0.2 difference requires 193 users at 0.5 sig level
* 0.8 difference requires 13 users at 0.5 sig level

Detect big problems before they wreak havoc

* Ramp up
* Start experiment with 0.5% participants
* Simple analysis to find large problems
* Slowly increase percentage up to 5%
* Abort
* If significant change for the worse is detected automatically abort

How to get valuable insights

* Focus on important part of problem, not entire prototype
* Not too many independent changes between A and B
* Make tasks realistic
* Fair test
* Compare two good alternatives

Designing Experiment

User study vs Experiment

User study

* Real users
* Measures data: How many errors

Experiment

* Also real users
* Tests hypothesis: People will make less errors on A than B

Goals

1. Internal Validity: Difference in measurement caused by change in condition
   1. What can hinder internal validity? (Confounding variables, things that change systematically with independent variables)
   2. Dealing with confounding variables (Constant variable across all cases, allowing a variable to vary randomly but uniformly for everyone. Higher generalizability of result, lower precision, requires more study participants, randomization within blocks: allowing a variable to randomly vary within constraints [Compromised approach])
2. External Validity: Can experiment result apply to other situations

Should participants use all conditions?

* Yes: Fewer people for more comparable results
* No: Takes too long, might taint the results

Between subject design: Half participants use one, half use the other

* Minimising learning and transfer across conditions
* Shorter sessions
* Easier to set up

Within subject design: All participants use both systems

* Requires fewer participants (Cheaper)
* Minimize random noise since all participants subjected to same conditions
* Problem: Ordering effects fatigue and learning.
* Solution: Half use one first then use the other one. Latin square for three of more conditions

Unmoderated User tests

* Doesn’t require researcher to be present in every session
* Software application provides instructions instead and facilitates
* Bad for early prototype testing: More errors which researchers have to help participants recover from
* Participants less engaged without researcher, behaving less realistically